

# **RANCANG SISTEM RUMAH WALET CERDAS BERBASIS INTERNET OF THINGS**



## **TUGAS AKHIR**

*Disusun dalam rangka memenuhi salah satu persyaratan*

*Untuk menyelesaikan program Strata-1 Departemen Teknik Informatika*

*Fakultas Teknik Universitas Hasanuddin*

*Makassar*

**Disusun Oleh :**

**MUHAMMAD FACHRIAL YUNI YUNIZAR YUNUS**

**D421 14 022**

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2019**

## ABSTRAK

Budidaya walet merupakan usaha yang sekarang banyak ditekuni masyarakat Indonesia. Terlebih lagi mudahnya pembangunan rumah walet yang digunakan sebagai tempat budidaya sarang walet itu sendiri. Selain hal tersebut, terdapat juga faktor-faktor penting yang pembudidaya harus perhatikan dalam melakukan budidaya walet. Faktor penting tersebut adalah kondisi ideal rumah walet yang harus menyerupai habitat aslinya. Karena faktor tersebutlah membuat pembudidaya sering masuk pada rumah walet untuk memantau kondisi di dalam rumah walet. Penelitian ini bertujuan menggunakan *internet of things* dalam memantau kondisi rumah walet tanpa perlu masuk di dalam rumah walet tersebut.

Penelitian ini memiliki fokus kerja memantau suhu, kelembapan dan populasi walet secara jarak jauh pada halaman *web*. Dengan sensor yang digunakan adalah DHT11 dan *Transmitter Infra merah*. Kedua hasil dari sensor akan di proses pada *End Devices* dimana setelah itu seluruh data akan dikirim ke *Gateway* setiap ±60s menggunakan teknologi LoRa. Data yang telah diterima *Gateway* akan dikirim ke *database web server* menggunakan *HTTP Request* sehingga dapat dipantau secara jarak jauh pada halaman *web*.

Berdasarkan pengujian sistem penghitung populasi walet yang telah dilakukan, diperoleh kesimpulan bahwa rata-rata akurasi pendekripsi sebesar 95,1% pada 12 kali percobaan. Sedangkan untuk pengujian sistem transmisi data menggunakan LoRa, rata-rata *time on air* pengiriman data ke *gateway* pada *Node A* adalah 2,88 detik, sedangkan *Node B* adalah 2,62 detik.

**Kata kunci :** *End Devices, Node, Gateway, HTTP Request, Time on Air*

## KATA PENGANTAR

Assalamu Alaikum Wr. Wb.

Puji dan syukur penulis panjatkan atas kehadirat Allah SWT karena berkat Rahmat dan Karunia-Nya sehingga Tugas Akhir yang berjudul "**RANCANG SISTEM RUMAH WALET CERDAS BERBASIS INTERNET OF THINGS**" ini dapat diselesaikan sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penyusunan penelitian ini disajikan hasil suatu penelitian yang menyangkut judul yang telah diangkat dan telah melalui proses pencarian dari berbagai sumber baik jurnal penelitian, *prosiding* pada seminar-seminar nasional/internasional, buku maupun dari situs-situs di internet.

Penulis menyadari bahwa dalam penyusunan dan penulisan skripsi ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tugas akhir, sangatlah sulit untuk menyelesaikan tugas akhir ini. Oleh karena itu, penulis dengan senang hati menyampaikan terima kasih kepada:

- 1) Kedua Orang tua penulis, Bapak H. Muh. Yunus Nabir dan Ibu Hj. Sumiati Sri Ningsi, saudari Fachira Azzahra Yunus, serta keluarga besar penulis, yang selalu memberikan dukungan, doa, dan semangat, juga selalu sabar dalam mendidik penulis sejak kecil;
- 2) Bapak Adnan, S.T., M.T., Ph.D., selaku pembimbing I dan Bapak Ir. Christoforus Yohannes, M.T., selaku pembimbing II yang selalu

menyediakan waktu, tenaga, pikiran dan perhatian yang sangat luar biasa untuk mengarahkan penulis dalam penyusunan tugas akhir;

- 3) Bapak Dr. Ir. Amil Ahmad Ilham, ST., M.IT., selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas bimbingannya selama masa perkuliahan penulis;
- 4) Ibu Ingrid Nurtanio, M.T., selaku dosen Departemen Teknik Informatika Universitas Hasanuddin atas bimbingan serta nasehatnya;
- 5) Bapak Dr. Eng. Zulkifli Tahir, S.T., M.Sc., selaku dosen Departemen Teknik Informatika Universitas Hasanuddin atas bimbingan serta nasehatnya;
- 6) Ibu Elly Warni S.T., M.T., selaku dosen pembimbing akademik penulis atas bimbingan serta nasehatnya;
- 7) Kepada Zulfahmi S.T., David Reinhart S.T., Auliati Nisa S.T., A. Khairil Fajri, dan Tiwi Nur Safitri S.T., yang selalu membantu selama penelitian, pengambilan data dan diskusi progress penyusunan Tugas Akhir, serta menjadi kawan maupun lawan dalam suka duka penyusunan skripsi ini;
- 8) Kepada Suriani Sudirman Amd.Kep., yang tidak henti-hentinya memberikan kasih sayang, semangat, dan bantuan selama penyusunan skripsi ini;
- 9) Kepada Fitriani Sudirman, yang tidak henti-hentinya mengingatkan dan memberi semangat selama penyusunan skripsi ini;
- 10) Para stackholder dari PT. CARAKDE, yang tiada henti-hentinya memberikan dukungan, masukan dan semangat di masa-masa sulit;

- 11) Para Pajokka Squad yang selalu memberikan dukungan dan semangat di masa-masa sulit;
- 12) Keluarga angkatan 2014 Departemen Teknik Informatika FT-UH atas semua bantuan dan semangat yang diberikan selama ini;
- 13) Teman-teman angkatan RECTIFIER FT-UH atas dukungan dan semangat yang diberikan selama ini;
- 14) Para Sahabat, teman-teman, adik-adik dan kakak-kakak anggota Laboratorium Internet of Things dan Parallel Computing Universitas Hasanuddin yang telah memberikan begitu banyak bantuan selama penelitian, pengambilan data dan diskusi progress penyusunan Tugas Akhir;
- 15) Segenap Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang selalu dan tak kenal lelah membantu penulis;
- 16) Orang-orang berpengaruh lainnya yang tanpa sadar telah menjadi inspirasi penulis;

Akhir kata, penulis berharap semoga Allah SWT. berkenan membala segala kebaikan dari semua pihak yang telah banyak membantu. Semoga Tugas Akhir ini dapat memberikan manfaat bagi pengembangan ilmu. Aamiin.

*Wassalam*  
Makassar, 01 Agustus 2019

Penulis

## DAFTAR ISI

	<b>halaman</b>
<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN .....</b>	<b>ii</b>
<b>ABSTRAK .....</b>	<b>iii</b>
<b>KATA PENGANTAR .....</b>	<b>iv</b>
<b>DAFTAR ISI .....</b>	<b>vii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xi</b>
<b>DAFTAR TABEL .....</b>	<b>xiv</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
I.1 Latar Belakang.....	1
I.2 Rumusan Masalah.....	3
I.3 Tujuan Penelitian .....	3
I.4 Manfaat Penelitian .....	3
I.5 Batasan Masalah Penelitian .....	4
I.6 Sistematika Penulisan .....	4
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>6</b>
II.1 Burung Walet .....	6
II.2 Budidaya Walet .....	7
II.3 Memonitor suhu dan Kelembapan Rumah Walet .....	8
II.4 Metode Penghitung Populasi Walet .....	10
II.5 Mikrokontroler Arduino .....	12

II.5.1 Pengenalan Arduino .....	12
II.5.1.1 Arduino Uno R3 .....	13
II.5.1.2 IDE Arduino .....	15
II.5.2 ATMega328 .....	16
II.6 Sensor DHT11 .....	20
II.7 <i>Transceiver Infrared</i> (Infra Merah) .....	21
II.8 Teknologi Transmisi LoRa ( <i>Long Range</i> ) .....	23
II.8.1 LoRa <i>Shield</i> .....	24
II.9 <i>Mini PC Raspberry Pi</i> .....	27
II.9.1 Sistem Operasi Raspberry Pi .....	28
II.9.2 Raspberry Pi 3 Model B .....	29
II.9.3 Spesifikasi Raspberry Pi 3 Model B .....	30
II.10 <i>Web Server</i> .....	31
II.10.1 <i>Web Server Apache</i> .....	32
II.10.1.1 Keunggulan <i>Web Server Apache</i> .....	33
II.10.1.2 Fasilitas dari Apache .....	33
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>35</b>
III.1 Tempat dan Waktu Penelitian .....	35
III.2 Alat dan Bahan Penelitian .....	35
III.3 Kriteria Desain .....	35
III.4 Prosedur Penelitian .....	37
III.5 Tahap Persiapan .....	39
III.6 Tahap Pembuatan Sistem .....	39

III.6.1 Tahap Perancangan Sistem Hardware (Perangkat Keras).....	39
III.6.1.1 Arduino Uno R.3 .....	41
III.6.1.2 Sensor DHT11 .....	41
III.6.1.3 <i>Transceiver</i> Infra merah (Deteksi Halangan) .....	42
III.6.1.4 LoRa <i>Shield Cytron</i> .....	45
III.6.1.5 LoRa HopeRF Modul .....	46
III.6.1.6 Raspberry Pi 3 Model B .....	47
III.6.2 Tahap Perakitan / Perwujudan Alat .....	48
III.6.3 Tahap Pembuatan <i>Software</i> (Perangkat Lunak) .....	49
II.6.3.1 Deteksi Suhu dan Kelembapan .....	52
III.6.3.2 Penghitung Populasi Walet .....	52
III.6.3.3 Sistem Transmisi Data .....	54
III.6.3.4 Visualiasasi Data pada <i>Web Server</i> .....	54
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>55</b>
IV.1 Hasil Rancang Bangun Alat .....	55
IV.2 Pengujian Sistem .....	58
IV.2.1 Pengujian Sistem Pembacaan Suhu dan Kelembapan .....	58
IV.2.2 Pengujian Sistem Penghitung Populasi Walet.	59
IV.2.3 Pengujian Sistem Transmisi Data .....	62
IV.2.4 Pengujian Visualisasi Data Pada <i>Web Server</i> ..	63

IV.3 Pengambilan Data .....	65
IV.4 Hasil Pengamatan dan Data .....	68
IV.4.1 Sistem Pembacaan Suhu dan Kelembapan .....	68
IV.4.2 Sistem Penghitung Populasi Walet .....	69
IV.4.3 Sistem Transmisi Data .....	70
IV.4.4 <i>Web Server</i> .....	73
<b>BAB V PENUTUP .....</b>	<b>75</b>
V.1 Kesimpulan .....	75
V.2 Saran .....	76
<b>DAFTAR PUSTAKA .....</b>	<b>78</b>
<b>LAMPIRAN</b>	

## DAFTAR GAMBAR

<b>Nomor</b>		<b>halaman</b>
Gambar 2.1	Desain Umum Gedung Walet.....	8
Gambar 2.2	Digital <i>Thermometer</i> dan <i>Hygrometer</i> .....	9
Gambar 2.3	Sarang Walet .....	10
Gambar 2.4	<i>Hand Tally Counter / Alat Hitung Digital</i> .....	11
Gambar 2.5	<i>CCTV Infrared</i> .....	12
Gambar 2.6	Pin <i>Mapping</i> Arduino Uno R3 .....	14
Gambar 2.7	Tampilan IDE Aplikasi Arduino v.1.6.9 .....	15
Gambar 2.8	Pin <i>Mapping Chip</i> ATMega328 .....	17
Gambar 2.9	Pin <i>Mapping</i> DHT11 .....	21
Gambar 2.10	<i>Transceiver Infrared</i> .....	22
Gambar 2.11	Pin <i>Mapping LoRa Shield</i> RFM Cytron .....	25
Gambar 2.12	Komponen Raspberry Pi 3 Model B .....	30
Gambar 3.1	Gambaran Umum Cara Kerja Sistem .....	36
Gambar 3.2	Diagram Tahapan Penelitian .....	37
Gambar 3.3	Skematik Rangkaian <i>End Devices</i> .....	40
Gambar 3.4	Skematik Pendekripsi Suhu dan Kelembapan .....	42
Gambar 3.5	Rangkaian Pendekripsi Halangan .....	43
Gambar 3.6	Skematik Rangkaian Pendekripsi Halangan .....	44
Gambar 3.7	Skematik Rangkaian LoRa <i>Shield</i> dan Arduino .....	46
Gambar 3.8	Skematik Rangkaian Sistem <i>Gateway</i> .....	47
Gambar 3.9	Blok Diagram Cara Kerja Alat .....	49

Gambar 3.10 <i>Flowchart</i> Sistem <i>End Devices</i> Rumah Walet Cerdas .....	50
Gambar 3.11 <i>Flowchart</i> Sistem <i>Gateway</i> Rumah Walet Cerdas .....	51
Gambar 4.1 Box Miniatur Pintu Keluar dan Masuk Walet .....	55
Gambar 4.2 Rangkaian Sistem <i>End Devices</i> .....	56
Gambar 4.3 Rangkaian Sistem <i>Node A</i> .....	56
Gambar 4.4 Rangkaian Sistem <i>Node B</i> .....	57
Gambar 4.5 Rangkaian Sistem <i>Gateway</i> .....	57
Gambar 4.6 Tampilan Data yang dikirim <i>Node</i> pada Serial Monitor .....	58
Gambar 4.7 Penerima Infra merah Mendeteksi Halangan .....	59
Gambar 4.8 Proses Pengiriman Data pada Serial Monitor <i>Node</i> .....	62
Gambar 4.9 Proses Penerimaan Paket Data pada Sistem <i>Gateway</i> .....	62
Gambar 4.10 Visualisasi Data pada <i>Website</i> .....	65
Gambar 4.11 Proses Pengambilan Data Sensor .....	67
Gambar 4.12 Penempatan <i>Node A</i> , <i>Node B</i> dan <i>Gateway</i> .....	68
Gambar 4.13 Tampilan Data Suhu dan Kelembapan pada Halaman <i>Web</i> ..	66
Gambar 4.14 Halaman <i>Login</i> .....	73
Gambar 4.14 Halaman Rumah Walet .....	73
Gambar 4.15 Tampilan Data Rumah Walet pada Halaman <i>Web</i> .....	74

## DAFTAR TABEL

<b>Nomor</b>		<b>halaman</b>
Tabel 2.1 Spesifikasi Arduino Uno R3 .....		14
Tabel 2.2 Spesifikasi Teknis DHT11.....		21
Tabel 3.1 Daftar Komponen Sistem <i>End Device</i> .....		40
Tabel 3.2 <i>Wiring</i> DHT11 dan Arduino Uno .....		42
Tabel 3.3 <i>Wiring</i> LoRa <i>Shield</i> dan Arduino Uno .....		45
Tabel 3.4 <i>Wiring</i> LoRa HopeRF modul dengan Raspberry Pi 3 .....		46
Tabel 4.1 <i>Database</i> Data Sensor .....		64
Tabel 4.2 Hasil Pengujian Sistem Penghitung Populasi .....		69
Tabel 4.3 Hasil Pengamatan Sistem Transmisi Data pada <i>Node A</i> .....		71
Tabel 4.4 Hasil Pengamatan Sistem Transmisi Data pada <i>Node B</i> .....		72

# **BAB I**

## **PENDAHULUAN**

### **I.1 Latar Belakang**

Sarang walet adalah komoditi yang mempunyai nilai ekonomis tinggi maka banyak masyarakat saat ini berbondong-bondong membudidayakan walet dengan membangun rumah bertingkat dimana ruangan lantai atas ataupun seluruh ruangan pada satu bangunan dikhkususkan sebagai rumah walet. Namun dalam pembudidayaannya sulit dilakukan karena membutuhkan strategi dan penanganan khusus agar walet mau membuat sarang di rumah walet tersebut.

Salah satu faktor penting yang harus diperhatikan adalah kondisi ideal rumah walet yang harus menyerupai habitat aslinya. Dimana sesuai habitat asli, terdapat beberapa faktor yang sangat mempengaruhi mengapa walet memilih gua-gua sebagai tempat bersarangnya. (Rachmad Andri, 2013).

Menurut pakar (Arief Budiman, 2002), faktor yang mempengaruhi adalah habitat aslinya mempunyai suhu dan kelembapan ideal yaitu pada rentan suhu 26-29°C dan kelembapan berkisar antara 80-90% Serta faktor yang tidak kalah pentingnya adalah rumah walet jauh dari aktivitas manusia. Hal itu dikarenakan walet membutuhkan rasa aman, nyaman dan tenang dalam berkembang biak.

Saat ini pembudidaya walet sudah melakukan beberapa metode dalam memenuhi faktor-faktor penting tersebut guna mendapatkan kondisi ideal rumah walet. Salah satu metode yang digunakan pembudidaya adalah monitoring suhu dan kelembapan rumah walet yang dilakukan dengan menggunakan alat sensor suhu dan kelembapan dimana pembudidaya diharuskan memonitor suhu dan

kelembapan dari dalam rumah walet. Metode tersebut memiliki kekurangan yaitu setiap kali pengukuran harus masuk ke dalam ruangan, hal tersebut jelas menjadi permasalahan dikarenakan mengganggu kenyamanan dan keamanan walet.  
(Rachmad Andri, 2013)

Hal lain yang menjadi permasalahan bagi pembudidaya adalah penghitungan populasi walet di dalam rumah walet, dimana tujuannya adalah mengetahui jumlah populasi walet yang berkembang biak di rumah walet. Cara pembudidaya dalam melakukan penghitungan populasi masih dilakukan dengan cara manual yaitu dengan cara menghitung jumlah sarang walet, dimana setiap sarang walet diperkirakan memiliki 1 pasang walet dan 2 ekor anak walet. Cara lain adalah sekitar jam 5 pagi hingga jam 11 siang diluar rumah walet pembudidaya menghitung jumlah walet yang keluar dari rumah walet yang melalui pintu masuk rumah walet menggunakan *hand counter*. Cara-cara tersebut tidak akurat dan kurang efektif dikarenakan dapat mengganggu kenyamanan walet dan mempersulit pembudidaya dalam memonitor populasi walet.

Berdasarkan uraian di atas, peniliti tertarik untuk membuat suatu sistem berbasis *Internet of Things* yang dapat memudahkan pembudidaya dalam memonitor suhu dan kelembapan serta jumlah populasi walet di rumah walet tanpa perlu masuk di dalam rumah walet. Oleh karena itu, tugas akhir yang akan dilakukan berjudul “**Rancang Sistem Rumah Walet Cerdas Berbasis Internet of Things**”.

## **I.2 Rumusan Masalah**

Berdasarkan latar belakang, maka rumusan masalah pada tugas akhir ini adalah:

1. Berapa tingkat akurasi sistem penghitung jumlah populasi walet?
2. Bagaimana data-data sensor pada *end devices* dapat dikirim ke *database web server*?
3. Berapa lama *time on air* pengiriman data dari *node* ke *gateway*?

## **I.3 Tujuan Penelitian**

Tujuan dari tugas akhir ini adalah :

1. Untuk menghitung tingkat akurasi dari sistem penghitung jumlah populasi walet.
2. Untuk mengetahui metode pengiriman data dari *end devices* ke *database web server* sehingga dapat dipantau pada halaman web.
3. Untuk mengukur kecepatan pengiriman data dari *node* ke *gateway*.

## **I.4 Manfaat Penelitian**

Manfaat dari tugas akhir ini adalah :

- a. Manfaat untuk Instansi Penelitian

Menjadi salah satu sistem yang dapat dikembangkan untuk penelitian-penelitian selanjutnya.

- b. Manfaat untuk pembudidaya Walet

1. Memudahkan pembudidaya dalam memantau suhu dan kelembapan rumah walet secara jarak jauh.

2. Memudahkan pembudidaya dalam memantau jumlah populasi walet yang tinggal di rumah walet secara jarak jauh.
3. Mengurangi aktivitas manusia di dalam rumah walet.

## I.5 Batasan Masalah Penelitian

Yang menjadi batasan masalah dalam tugas akhir ini adalah :

1. Sistem penghitung populasi walet tidak dapat membedakan obyek yang dideteksi sehingga data yang terdeteksi akan tetap dianggap sebagai walet.
2. *Receiver* (penerima) infra merah tidak mampu mendeteksi dua atau lebih walet yang bersamaan masuk ataupun keluar.
3. Sensor yang digunakan adalah sensor DHT11 untuk mengukur suhu dan kelembapan, sedangkan pemancar dan penerima infra merah untuk menghitung populasi walet di dalam rumah walet.

## I.6 Sistematika Penulisan

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu :

## **BAB I PENDAHULUAN**

Bab ini menguraikan secara umum mengenai hal yang menyangkut latar belakang, perumusan masalah dan batasan masalah, tujuan, manfaat, dan sistematika penulisan.

## **BAB II TINJAUAN PUSTAKA**

Bab ini berisi teori-teori tentang hal-hal yang berhubungan dengan budidaya walet, pembacaan sensor pada mikrokontroler arduino, *Internet of Things*, teknologi transmisi LoRa, pembuatan *web server* pada Raspberry Pi dan metode yang digunakan.

## **BAB III METODOLOGI PENELITIAN**

Bab ini membahas tentang jenis penelitian, alat dan bahan penelitian, metode perancangan sistem, metode pengumpulan data, metode pengujian dan hasil penelitian.

## **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil perancangan sistem dan pengolahan data hasil penelitian.

## **BAB V PENUTUP**

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan lebih lanjut.

## **BAB II**

### **TINJAUAN PUSTAKA**

Seperti pada penjelasan sistematika penulisan pembuatan tugas akhir ini, dalam bab ini akan dijelaskan beberapa landasan-landasan teori sebagai hasil dari studi literature yang berhubungan dalam perancangan dan pembuatan alat. Pada bab ini penulis akan membahas dasar-dasar dari beberapa bagian penting yang terdapat pada sistem rumah walet cerdas berbasis IoT.

#### **II.1 Burung Walet**

Burung Walet (*Collocalia fuciphaga*) merupakan ternak unggas yang dibudidayakan dengan sarang sebagai produksi utama yang bernilai ekonomi yang sangat tinggi. Sarang yang dihasilkan *Collocalia fuciphaga* merupakan sarang putih yang sebagian besar terbentuk dari air liur burung tersebut. Produksi sarang burung walet dipengaruhi oleh berbagai faktor, salah satunya yaitu faktor kondisi lingkungannya yang idealnya menyerupai habitat aslinya yaitu di gua-gua (Turaina Ayuti, 2016).

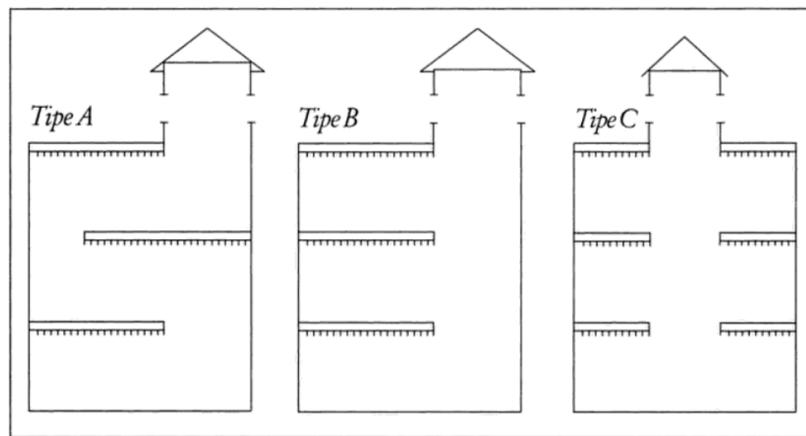
Menurut pakar budidaya walet (Arief Budiman, 2002), faktor yang mempengaruhi adalah habitat aslinya mempunyai suhu dan kelembapan ideal yaitu pada rentan suhu 26-29°C dan kelembapan berkisar antara 80-90% Serta faktor yang tidak kalah pentingnya adalah rumah walet jauh dari aktivitas manusia. Hal itu dikarenakan walet membutuhkan rasa aman, nyaman dan tenang dalam berkembang biak.

Burung walet memiliki beberapa ciri khas yang tidak dimiliki oleh burung lain. Ciri khas tersebut diantaranya melakukan hampir segala aktivitasnya di udara seperti makan dan bereproduksi dengan kecepatan terbang rata-rata 25 mil/jam atau 11,176 m/detik, sehingga burung walet sering disebut dengan burung layang-layang.

## **II.2 Budidaya Walet**

Budidaya walet di Indonesia telah ada sejak abad ke-18 dan banyak dikembangkan di luar habitat aslinya, yaitu pada rumah burung walet. Indonesia merupakan penyedia sarang burung walet dunia. Tempat penyebaran pembudidayaan rumahannya telah tersebar diberbagai pulau di Indonesia. Ekspor sarang burung walet dilakukan ke berbagai negara di Asia dan Eropa, serta Australia dan Amerika Serikat (Turaina Ayuti, 2016).

Menurut pakar budidaya walet (Arief Budiman, 2002), dalam membangun gedung walet sebenarnya tidak ada desain khusus. Setiap orang bias membuat rancangan sendiri. Sebab utamanya adalah yang dibutuhkan walet sebenarnya bukan desain yang bagus, melainkan kondisi iklim mikro dalam gedung yang sesuai dengan habitat mikro walet. Dengan demikian, yang lebih penting adalah kondisi gedung yang benar. Semua desain, asalkan memenuhi kebutuhan habitat aslinya. Berikut ini tiga desain ruang gedung walet yang umum dipakai bisa dilihat pada gambar 2.1



**Gambar 2.1 Desain Umum Gedung Walet**

(Arief Budiman, 2002)

Di habitat asalnya walet menghuni gua-gua alam yang bersuhu dingin, lembab, memiliki ruang gelap, keamanan terjamin dan memiliki tempat bersarang yang mudah. Gua alam yang tidak memiliki kondisi seperti itu selamanya tidak akan dipilih walet sebagai tempat untuk berkembang biak. Kondisi yang sesuai dengan kondisi gua alam yang dipilih sebagai tempat berkembang biak itu bias dipakai sebagai patokan dalam membangun gedung walet. Membangun gedung walet yang ideal haruslah mampu menduplikasi kondisi iklim mikro gua-gua alam yang dihuni walet. Dengan demikian, gedung ideal untuk budidaya walet adalah gedung yang suhu, kelembapan, intensitas cahaya, dan keamanannya menyerupai kondisi gua alam tempat tinggal walet.

### **II.3 Memonitor Suhu dan Kelembapan Rumah Walet**

Salah satu faktor walet dapat berkembangbiak dengan baik adalah Suhu dan kelembapan rumah walet menyerupai kondisi habitat aslinya, dimana rentang suhu  $26^{\circ}\text{C} - 29^{\circ}\text{C}$  dan kelembapan relatif 80% - 90%. Dalam menyelesaikan

permasalahan tersebut, pembudidaya telah melakukan beberapa metode dalam memonitor suhu dan kelembapan rumah walet agar tetap stabil.

Dalam memonitor suhu dan kelembapan rumah walet, beberapa pembudidaya menggunakan alat pengukur suhu dan kelembapan untuk mendeteksi keadaan ruangan rumah walet, sehingga pembudidaya dapat mengambil keputusan dalam mengatur kondisi suhu dan kelembapan rumah walet. Salah satu alat yang digunakan pembudidaya untuk *monitoring* suhu dan kelembapan adalah Digital *Thermometer & Hygrometer*, gambar alat dapat dilihat pada gambar 2.2.



**Gambar 2.2** Digital *Thermometer* dan *Hygrometer*  
(waletcentral.com)

Metode pembudidaya memiliki kekurangan yaitu pembudidaya diharuskan untuk masuk kerumah walet dalam *monitoring* suhu dan kelembapan. Hal tersebut dapat mengganggu kenyamanan dan keamanan walet dalam berkembang biak.

## **II.4 Metode Penghitung Populasi Walet**

Pembudidaya walet sering mengalami kesulitan dalam menentukan metode terbaik yang digunakan dalam budidaya walet, dikarenakan sulitnya mengetahui apakah walet sudah betah tinggal di rumah walet menggunakan metode yang dipilih.

Salah satu cara pembudidaya walet dalam mengetahui betah tidaknya walet tinggal di rumah walet adalah dengan menghitung populasi walet yang ada di dalam rumah walet. Saat ini metode pembudidaya walet dalam melakukan penghitungan populasi walet yang tinggal di dalam rumah walet masih dilakukan secara manual. Beberapa metode penghitung populasi walet yang digunakan pembudidaya walet :

1. Metode pertama yaitu menghitung Sarang Walet. Metode ini digunakan pembudidaya dalam menghitung populasi walet dengan cara, menghitung jumlah sarang walet yang ada di dalam rumah, lalu mengasumsikan bahwa satu sarang dihuni dua induk dan dua anakan walet. Metode ini kurang akurat dikarenakan hanya berupa asumsi pembudidaya saja. Contoh sarang walet dapat dilihat pada gambar 2.3.



**Gambar 2.3 Sarang Walet**

(waletcentral.com)

2. Metode pembudidaya selanjutnya menggunakan adalah alat bantu *hand tally counter* / alat hitung yang dapat dilihat pada gambar 2.4. Metode tersebut mengharuskan pembudidaya melihat pintu masuk rumah walet selama berjam-jam di luar rumah walet atau menggunakan CCTV lalu menghitung walet yang masuk sehingga memiliki tingkat resiko kesalahan yang tinggi.



**Gambar 2.4 Hand Tally Counter / Alat Hitung Digital**  
(lafayettelifeosciences.com)

3. Metode yang ketiga, pembudidaya tetap menggunakan alat penghitung manual namun dikombinasikan dengan CCTV. Jadi metode ini tidak mengharuskan pembudidaya berada diluar rumah walet dan memperhatikan pintu masuk walet selama berjam-jam. Cukup menghitung melalui layar monitor yang terhubung ke CCTV dimana CCTV itu sendiri menampilkan pintu masuk walet. Contoh CCTV yang digunakan pembudidaya, bisa dilihat pada gambar 2.5.



**Gambar 2.5** CCTV *Infrared*

(diy-security-cameras.com)

Metode metode yang digunakan pembudidaya dalam menghitung populasi walet ini kurang efektif dikarenakan masih dilakukan secara manual.

## **II.5 Mikrokontroler Arduino**

### **II.5.1 Pengenalan Arduino**

Arduino dikatakan sebagai sebuah *platform* dari *physical computing* yang bersifat *open source*. Pertama - tama perlu dipahami bahwa kata “*platform*” disini adalah sebuah pilihan kata yang tepat. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah kombinasi dari *hardware*, bahasa pemrograman dan *Integrated Development Environment* (IDE) yang canggih. IDE sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam *memory mikrokontroler*. Ada banyak projek dan alat – alat yang dikembangkan oleh akademisi dan profesional dengan menggunakan Arduino, selain itu juga ada banyak modul – modul pendukung (sensor, tampilan,

penggerak dan sebagainya) yang dibuat oleh pihak lain untuk bisa disambungkan dengan Arduino. Arduino berevolusi menjadi sebuah *platform* karena ia menjadi pilihan dan acuan bagi banyak praktisi.

Salah satu yang membuat arduino memikat hati banyak orang adalah karena sifatnya *open source*, baik untuk *hardware* maupun *software*-nya. Komponen utama didalam papan Arduino adalah sebuah mikrokontroller 8 bit dengan merk Atmega yang dibuat oleh perusahaan Atmel *Corporation*. Berbagai papan Arduino menggunakan tipe Atmega yang berbeda-beda, sebagai contoh Arduino Uno menggunakan Atmega328, Arduino Nano, Arduino Mega, Arduino Zero, Arduino Yun dan Arduino Due. Yang membedakan adalah spesifikasinya. (Safari Nurliana, 2016).

Bahasa pemrograman Arduino merupakan bahasa C yang sudah disederhanakan syntax bahasa pemrogramannya sehingga mempermudah pengguna dalam mempelajari dan mendalami mikrokontroler.

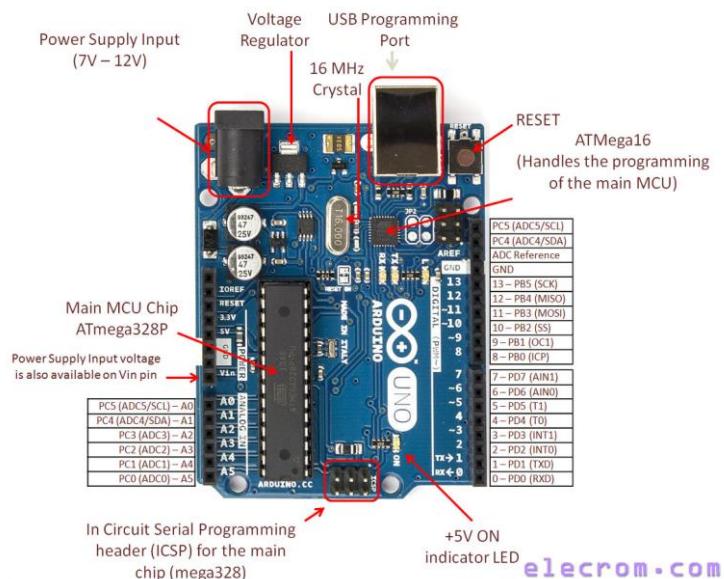
### **II.5.1.1 Arduino Uno R3**

Arduino Uno Revisi 3 adalah papan pengembangan mikrokontroler yang berbasis chip ATmega328P. Arduino Uno memiliki 14 digital pin *input / output* (atau biasa ditulis I/O, dimana 14 pin diantaranya dapat digunakan sebagai *output* PWM antara lain pin 0 sampai 13), 6 pin *input* analog, menggunakan *crystal* 16 MHz antara lain pin A0 sampai A5, koneksi USB, jack listrik, *header* ICSP dan tombol *reset*. Hal tersebut adalah semua yang diperlukan untuk mendukung sebuah rangkaian mikrokontroler. Spesifikasi Arduino Uno R3 dapat dilihat pada

tabel 2.1 dan Arduino Uno R3 dapat dilihat pada gambar 2.6. (Wicaksono & Sigit N., 2017).

**Tabel 2.1** Spesifikasi Arduino Uno R3

Mikrokontroler	ATmega328P
Operasi Tegangan	5 Volt
<i>Input</i> Tegangan	7-12 Volt
Pin I/O Digital	14
Pin Analog	6
Arus DC tiap pin I/O	6
Arus DC ketika 3.3V	50 mA
Memori <i>flash</i>	50 mA
SRAM	2 KB
EEPROM	1 KB
Kecepatan <i>clock</i>	16 MHz

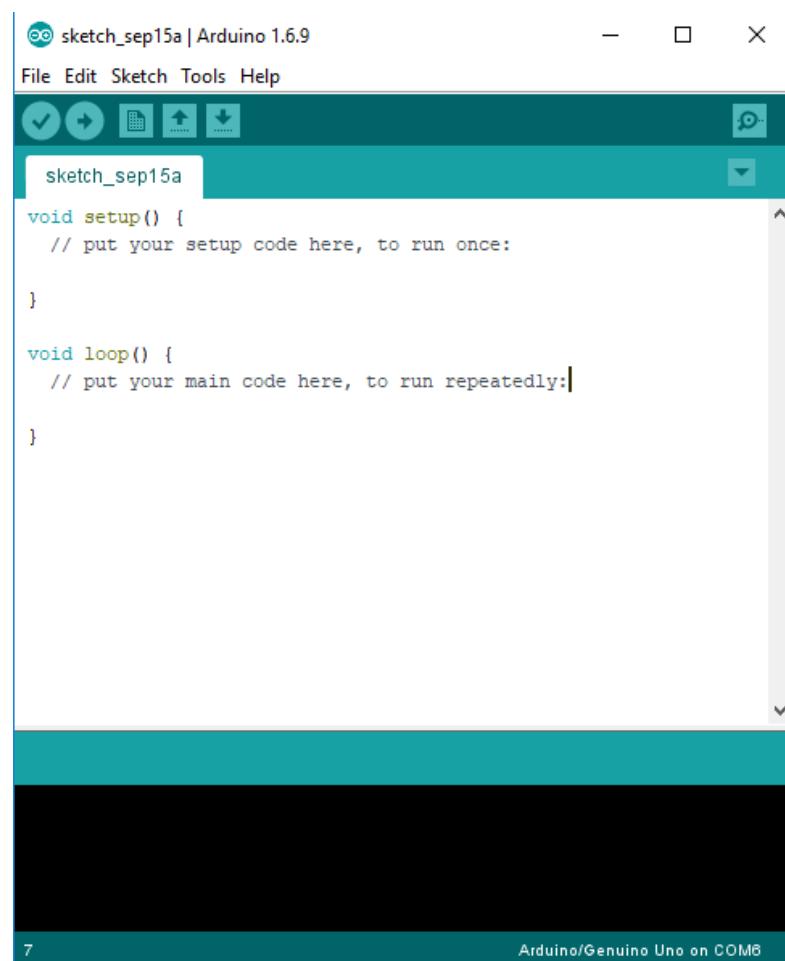


**Gambar 2.6** Pin Mapping Arduino Uno R3

([www.elecrom.com](http://www.elecrom.com))

### II.5.1.2 IDE Arduino

IDE (*Integrated Development Environment*) adalah sebuah perangkat lunak yang digunakan untuk mengembangkan aplikasi mikrokontroler mulai dari menuliskan *source* program, kompilasi, *upload* hasil kompilasi dan uji coba secara terminal serial. IDE Arduino dapat dilihat pada gambar 2.7.



**Gambar 2.7** Tampilan IDE Aplikasi Arduino v.1.6.9

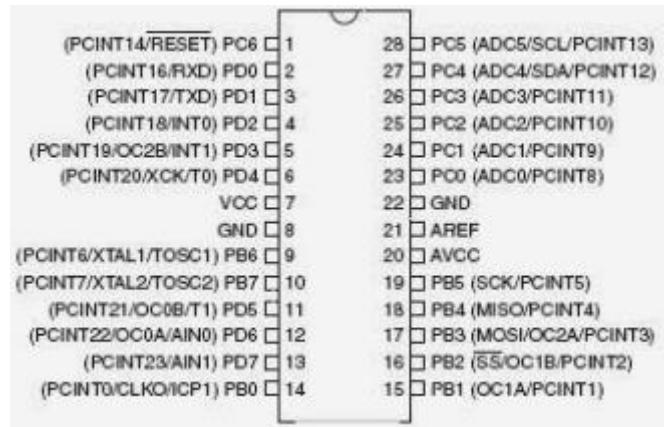
- a. Ikon menu *verify* yang bergambar tanda centang berfungsi untuk mengecek program yang ditulis apakah ada salah atau *error*.

- b. Ikon menu ***upload*** yang bergambar panah ke arah kanan berfungsi untuk memuat / *transfer* program yang dibuat di *software* Arduino ke *hardware* Arduino.
- c. Ikon menu ***New*** yang bergambar sehelai kertas berfungsi untuk membuat halaman baru dari dalam pemrograman.
- d. Ikon menu ***Open*** yang bergambar panah ke arah atas berfungsi untuk membuka program yang disimpan atau membuka program yang sudah dibuat dari pabrikan *software* Arduino.
- e. Ikon menu ***Save*** yang bergambar panah ke arah bawah berfungsi untuk menyimpan program yang telah dibuat atau dimodifikasi.
- f. Ikon menu ***Serial Monitor*** yang bergambar kaca pembesar berfungsi untuk mengirim atau menampilkan serial komunikasi data saat dikirim dari *hardware* Arduino.

### II.5.2 ATMega328

ATMega328 merupakan mikrokontroler keluarga AVR 8 bit. Beberapa tipe mikrokontroler yang sama dengan ATMega8 ini antara lain ATMega8535, ATMega16, ATMega32, ATMega328, yang membedakan antar mikrokontroler ini adalah, ukuran memori, banyaknya GPIO (Pin *Input/Output*), *peripheral* (USART, *timer*, *counter*, dll). Dari segi ukuran fisik, ATMega328 memiliki ukuran fisik lebih kecil dibandingkan dengan beberapa *mikrokontroler* diatas. Namun untuk segi memori dan *peripheral* lainnya ATMega328 tidak kalah dengan yang lainnya karena ukuran memori dan *peripheral*-nya relatif sama

dengan ATMega8535, ATMega32, hanya saja jumlah GPIO lebih sedikit dibandingkan *mikrokontroler* diatas.



**Gambar 2.8** Pin *Mapping Chip* ATMega328

(learningaboutelectronics.com)

Dapat dilihat pada gambar 2.8, ATMega328 memiliki 3 buah *port* utama yaitu *Port B*, *Port C*, dan *Port D* dengan total pin *input/output* sebanyak 23 pin. *Port* tersebut dapat difungsikan sebagai *input/output* digital atau difungsikan sebagai *peripheral* lainnya (Wicaksono & Sigit N., 2017).

### 1. Port B

*Port B* merupakan jalur data 8 bit yang dapat difungsikan sebagai *input/output*. Selain itu PortB juga dapat memiliki fungsi alternatif seperti di bawah ini:

- a. ICP1 (PB0), berfungsi sebagai *Timer Counter 1 input capture* pin.
- b. OC1A (PB1), OC1B (PB2) dan OC2 (PB3) dapat difungsikan sebagai keluaran PWM (*Pulse Width Modulation*).

- c. SS (PB2), MOSI (PB3), MISO (PB4), SCK (PB5) merupakan jalur komunikasi SPI (*Serial Peripheral Interface*).
- d. Selain itu pin ini juga berfungsi sebagai jalur pemrograman serial (ISP).
- e. TOSC1 (PB6) dan TOSC2 (PB7) dapat difungsikan sebagai sumber *clock* eksternal untuk *timer*.
- f. XTAL1 (PB6) dan XTAL2 (PB7) merupakan sumber *clock* utama mikrokontroler.

## 2. *Port C*

*Port C* merupakan jalur data 7 bit yang dapat difungsikan sebagai input/output digital. Fungsi alternatif *Port C* antara lain sebagai berikut:

- a. ADC6 *channel* (PC0, PC1, PC2, PC3, PC4, PC5) dengan resolusi sebesar 10 bit. ADC (*Analog to Digital Converter*) dapat kita gunakan untuk mengubah *input* yang berupa tegangan analog menjadi data digital.
- b. I2C (SDA dan SDL) merupakan salah satu fitur yang terdapat pada *Port C*. I2C digunakan untuk komunikasi dengan sensor atau *device* lain yang memiliki komunikasi data tipe I2C seperti sensor kompas, *Accelerometer nunchunk*.

## 3. *Port D*

*Port D* merupakan jalur data 8 bit yang masing-masing pinnya juga dapat difungsikan sebagai *input/output*. Sama seperti *Port B* dan *Port C*, *Port D* juga memiliki fungsi alternatif di bawah ini:

- a. USART (TXD dan RXD) merupakan jalur data komunikasi serial dengan level sinyal TTL. Pin TXD berfungsi untuk mengirimkan data serial, sedangkan RXD kebalikannya yaitu sebagai pin yang berfungsi untuk menerima data serial.
- b. *Interrupt* (INT0 dan INT1) merupakan pin dengan fungsi khusus sebagai eksternal *event* yang memiliki prioritas lebih tinggi dari pada program utama sehingga ketika dipicu akan dieksekusi terlebih dahulu. Dengan kata lain, program utama yang seharusnya dijalankan secara berurutan tersebut akan ditunda untuk mengerjakan fungsi interupsi yang disebut *Interrupt Service Routine* (ISR) terlebih dahulu. Setelah ISR dijalankan, sistem akan kembali mengerjakan program utama yang telah ditunda tadi secara berurutan.
- c. XCK dapat difungsikan sebagai sumber *clock* eksternal untuk USART, namun kita juga dapat memanfaatkan *clock* dari CPU, sehingga tidak perlu membutuhkan eksternal *clock*.
- d. T0 dan T1 berfungsi sebagai masukan *counter* eksternal untuk *timer 0* dan *timer 1*.
- e. AIN0 dan AIN1 keduanya merupakan *input* untuk *analog comparator*.

ATMega328 adalah mikrokontroler yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*) yang mana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*). Mikrokontroler ini memiliki beberapa fitur antara lain (Wicaksono & Sigit N., 2017):

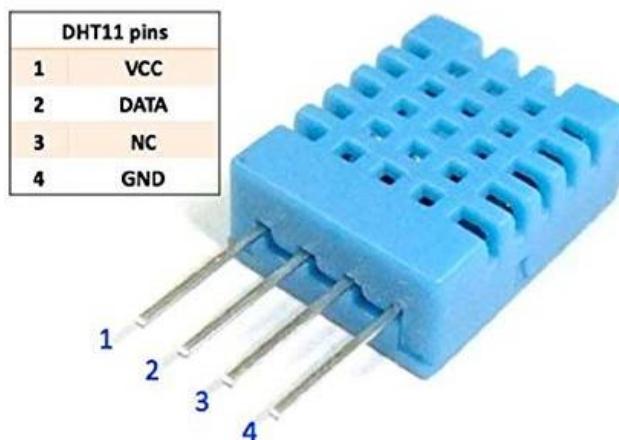
1. Memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1KB sebagai tempat penyimpanan data semi permanen karena EEPROM tetap dapat menyimpan data meskipun catu daya dimatikan.
2. Memiliki SRAM (*Static Random Access Memory*) sebesar 2KB.
3. Memiliki pin I/O digital sebanyak 14 pin 6 diantaranya PWM (*Pulse Width Modulation*) *Output*.
4. 32 x 8-bit register serba guna.
5. Dengan clock 16 MHz kecepatan mencapai 16 MIPS.
6. 32 KB *Flash* memori dan pada Arduino memiliki *bootloader* yang menggunakan 2KB dari *flash* memori sebagai *bootloader*.
7. 130 macam instruksi yang hamper semuanya dieksekusi dalam satu siklus *clock*.

## II.6 Sensor DHT11

DHT11 adalah sensor suhu dan kelembapan yang memiliki *output* sinyal digital yang telah terkalibrasi. Sensor DHT11 ini menggunakan teknik *Digital Signal Acquisition* dan teknologi penginderaan suhu dan kelembapan, sensor ini memiliki komponen pengukuran suhu NTC (*Negative Temperature Coefisien*) yang terhubung pada *mikrokontroler* 8 bit sehingga menawarkan kualitas yang baik, respon cepat dan kemampuan anti gangguan. Adapun spesifikasi dari sensor DHT11 ini bisa dilihat pada tabel 2.2 dan untuk pin *mapping* bisa dilihat pada gambar 2.9.

**Tabel 2.2** Spesifikasi Teknis DHT11

Sensor	Rentang Pengukuran	Toleransi Kelembapan	Toleransi Suhu	Resolusi	Pin
DHT11	20% – 95% & 0 – 50°C	+5%	+2°C	1	4 Pin Single Row



**Gambar 2.9** Pin Mapping DHT11

(alselectro.com)

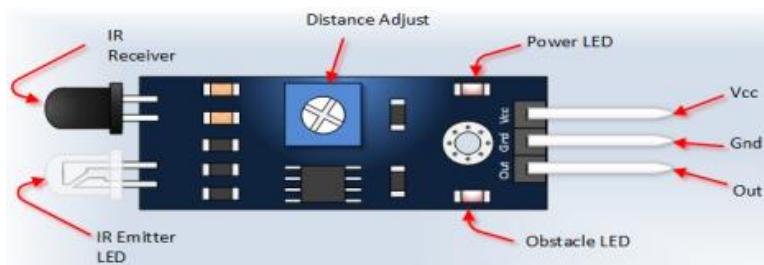
Dari gambar 2.9 terdapat 4 pin pada sensor DHT11 yaitu pin VCC dengan tegangan 3,5 – 5 volt DC yang dihubungkan ke pin *power* Arduino, pin DATA yang dihubungkan ke salah satu pin digital Arduino, pin NC (*Not Connection*) tidak digunakan, dan pin GND dihubungkan ke pin *ground* Arduino (Rachmad Andri, 2013).

## II.7 Transceiver Infrared (Infra Merah)

*Transceiver Infrared* / infra merah merupakan sebuah modul yang terdiri dari LED infra merah dan photodioda yang berfungsi sebagai pendekripsi halangan

atau objek di depannya, berikut adalah komponen-komponen yang ada pada modul tersebut beserta gambar dapat dilihat pada gambar 2.10 (Okta Dwi H., 2017):

1. Komponen utamanya terdiri dari LED *Infrared* dan *Infrared receiver* / phototransistor.
2. Ketika power-up, IR emitter akan memancarkan cahaya *infrared* yang kasat mata.
3. Cahaya tersebut kemudian dipantulkan oleh objek yang ada di depannya. Cahaya terpantul ini kemudian diterima oleh IR *receiver*.
4. Terdapat Op-Amp LM363 yang berfungsi sebagai komparator antara resistansi IR *receiver* dan resistansi trimpot pengatur sensitivitas.
5. Output Op-Amp ini juga terhubung dengan pin “*Out*” yang dihubungkan ke Arduino.



**Gambar 2.10 Transceiver Infrared**  
(playembedded.org)

Konfigurasi pin :

- VCC tegangan eksternal 3,3 – 5 Volt
- GND ground

- OUT digital *output interface* (0 dan 1)

## II.8 Teknologi Transmisi LoRa (*Long Range*)

LoRa (*Long Range*) adalah suatu format modulasi yang unik dan mengagumkan yang dibuat oleh Semtech. Modulasi yang dihasilkan menggunakan modulasi FM. Inti pada pemrosesan menghasilkan nilai frekuensi yang stabil. Metode transmisi juga bisa menggunakan PSK (*Phase Shift Keying*), FSK (*Frequency Shift Keying*) dan lainnya. Nilai frekuensi pada LoRa bermacam-macam sesuai daerahnya, jika di Asia frekuensi yang digunakan yaitu 433 MHz, di Eropa nilai frekuensi yang digunakan yaitu 868 MHz, sedangkan di Amerika Utara frekuensi yang digunakan yaitu 915 MHz.

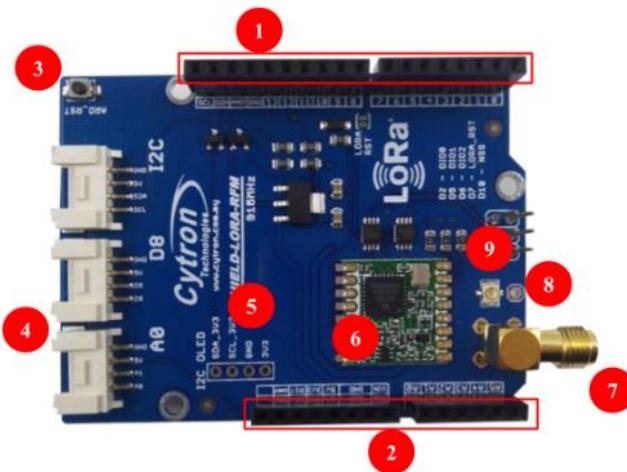
LoRa adalah teknologi nirkabel yang dikembangkan untuk menciptakan jaringan *wide-area low-power* (LPWAN) yang dibutuhkan untuk aplikasi dari mesin ke mesin (M2M) dan *Internet of Things* (IoT). Teknologi ini menawarkan perpaduan jangka panjang yang sangat menarik, konsumsi daya rendah dan transmisi data yang aman dan mendapatkan daya tarik yang signifikan dalam jaringan IoT yang digunakan oleh operator jaringan nirkabel. Jaringan berbasis teknologi nirkabel LoRa dapat memberikan jangkauan yang lebih besar dibandingkan dengan jaringan seluler yang ada. Sebenarnya, banyak operator jaringan seluler telah memilih untuk melengkapi jaringan seluler / nirkabel yang ada dengan LPWAN berdasarkan teknologi LoRa karena mudah dipasang ke infrastruktur yang ada dan juga memungkinkan mereka menawarkan solusi kepada pelanggan mereka untuk melayani lebih banyak baterai IoT-aplikasi yang dioperasikan (Saqiras Nurbaitil, 2018).

## **II.8.1 LoRa *Shield***

LoRa *Shield* adalah *transceiver* jarak jauh pada *form factor* perisai Arduino dan berdasarkan pada *open source library*. *Shield* memungkinkan pengguna untuk mengirim data dan mencapai rentang yang sangat panjang pada data-data rendah. Ini menyediakan spektrum penyebaran spektrum ultra jarak jauh dan tinggi. kekebalan interferensi sambil meminimalkan konsumsi saat ini. LoRa *Shield* berbasis pada *chip Semtech SX1276 / SX1278*, ini menargetkan aplikasi jaringan sensor nirkabel profesional seperti sistem irigasi, *metering* cerdas, kota cerdas, deteksi *smartphone*, otomasi bangunan, dan sebagainya. Menggunakan teknik modulasi LoRaTM yang dipatenkan oleh *Hope RF Shield* Cytron yang menampilkan teknologi LoRa dapat mencapai sensitivitas diatas -148dBm menggunakan kristal biaya rendah dan *bill of material*. Sensitivitas tinggi yang dikombinasikan dengan penguat daya +20 dBm terintegrasi menghasilkan anggaran link terdepan industri sehingga bisa optimal untuk aplikasi yang membutuhkan jangkauan atau ketahanan. LoRa juga memberikan keuntungan yang signifikan baik pada *blocking* maupun selektivitas terhadap teknik modulasi konvensional, memecahkan kompromi desain tradisional antara rentang, kekebalan interferensi dan konsumsi energi (Saqiras Nurbaitil, 2018).

Perangkat ini juga mendukung mode FSK kinerja tinggi (G) untuk sistem termasuk WMBus, IEEE 802.15.4g. Itu Dragino *Shield* yang menampilkan teknologi LoRa® memberikan *noise*, selektivitas, linearitas *receiver* dan IIP3 yang luar biasa untuk konsumsi arus yang jauh lebih rendah daripada perangkat bersaing. Fitur dari LoRa *Shield* ini sebagai berikut (Cytron techonologies, 2017) :

1. Kompatibel dengan papan Arduino 3.3v atau 5v I/O.
2. Terintegrasi dengan RFM95W LoRa *Module* yang menggunakan Frekuensi 915 MHz.
3. Maksimal sensitivitas *receiver* : down to -146 dBm.
4. TX Power : +14 dBm efisiensi tinggi PA.
5. FSK, GFSK, MSK, GMSK, LoRa and OOK *module*.
6. Arus RX rendah 10.3 mA, retensi *register* 200 nA.
7. Dilengkapi dengan SMA 90° antena konektor dan U.FL SMD antena konektor.
8. *Interface* via SPI dengan Arduino *main board*.



**Gambar 2.11 Pin Mapping LoRa Shield RFM Cytron**  
 (Cytron techonologies, 2017)

Pada gambar 2.11 dapat dilihat letak pin yang terdapat pada board dengan penjelasan sebagai berikut:

1. Pin Digital I/O

Merupakan pin digital *input/output* yang terpasang pada pin digital Arduino *main board*.

## 2. Pin Analog I/O, Pin POWER, pin RESET dan pin IOREF

Terdapat 6 pin analog *input/output* yang terpasang pada pin analog Arduino *main board* yaitu terdiri dari pin A0-A5. Terdapat juga pin VIN yang berfungsi sebagai input daya untuk *board* Arduino. Sedangkan pin POWER terdiri dari pin output 5v dan 3,3v serta pin Ground. Dan terdapat juga pin RESET dan pin IOREF.

## 3. Tombol *Reset* Arduino

Tombol ini digunakan pengguna untuk meng-*reset* Arduino *main board*.

## 4. *Grove* Konektor

Konektor untuk sensor yang kompatibel dengan *device grove* konektor. Terdapat 3 *grove* yang terkoneksi ke A0, D8, dan I2C.

## 5. I2C OLED

Pilihan *slot* untuk pengguna dalam menginstall OLED yang berfungsi sebagai *visual display*.

## 6. RFM95W LoRa *Module*

*Low-Power Long Range Transceiver* merupakan teknologi yang beroperasi pada frekuensi 915 MHz yang terintegrasi dengan RF.

## 7. SMA 90<sup>0</sup> Antena Konektor

Konektor untuk SMA antena, biasanya menggunakan antena 915 MHz 50ohm.

#### 8. Lubang Antena Konektor

Lubang antena yang bersifat *single wire*. Pengguna harus menyolder antena pada lubang antena konektor.

#### 9. U.FL SMD Antena Konektor

Pengguna dapat menggunakan U.FL SMD antena dengan menghubungkan antena dengan *pigtail* antena.

### **II.9 Mini PC Raspberry Pi**

Raspberry Pi merupakan sebuah computer berukuran mini sebesar kartu kredit dengan harga yang relatif murah. Raspberry ini memiliki dua model yaitu Model A dan Model B. Perbedaan diantara keduanya hanya terletak pada keberadaan Ethernet yang absen pada model A dan jumlah *port* USB yang menjadi dua kali pada model B. Walaupun kecil dan murah, tetapi RasPi tidak bisa disebut murahan. Pasalnya, banyak karya dan kegunaan yang bias dihasilkan dari RasPi, mulai dari fungsi utamanya sebagai computer yang memungkinkan kita untuk *browsing*, membuat laporan tugas, membuat *slide* presentasi, bermain *game*, atau sekedar mendengarkan musik dan menonton video. RasPi pun sangat membantu dalam pembuatan karya-karya inovatif, seperti penggunaan RasPi dalam robot mata-mata yang dilengkapi kamera. Bahkan ada yang menjadikan RasPi sebagai pengendali mobil otomatis. Ada pula yang membuat RasPi ini menjadi super computer.

Raspberry terdiri dari beberapa model yaitu :

1. Raspberry Pi Model A
2. Raspberry Pi Model A+

3. Raspberry Pi Model B
4. Raspberry Pi Model B+
5. Raspberry Pi 2
6. Raspberry Pi 3

### **II.9.1 Sistem Operasi Raspberry Pi**

Ini adalah daftar sistem operasi yang berjalan pada Raspberry Pi.

1. Full OS :
  - a. AROS
  - b. Haiku
  - c. LINUX :
    - Android : Android 4.0 (*Ice Cream Sandwich*)
    - Arch Linux ARM
    - R\_Pi Bodhi Linux
    - Debian Squeeze
    - Firefox OS
    - Gentoo Linux
    - Google Chrome OS : Chromium OS
    - PiBang Linux
    - Raspberry Pi Fedora
    - Raspbian
    - Slackware ARM
    - QtonPi
    - WebOS : Open webOS

d. Plan 9 from bell labs

e. RISC OS

f. UNIX :

- FreeBSD

- NETBSD

2. *Multi-purpose light distributions:*

a. Moebius, ARMHF distribusi berdasarkan Debian.

b. Squeezed Arm Puppy, versi Puppy Linux untuk ARM v6

3. *Single-purpose light distributions:*

a. IPfire

b. OpenELEC

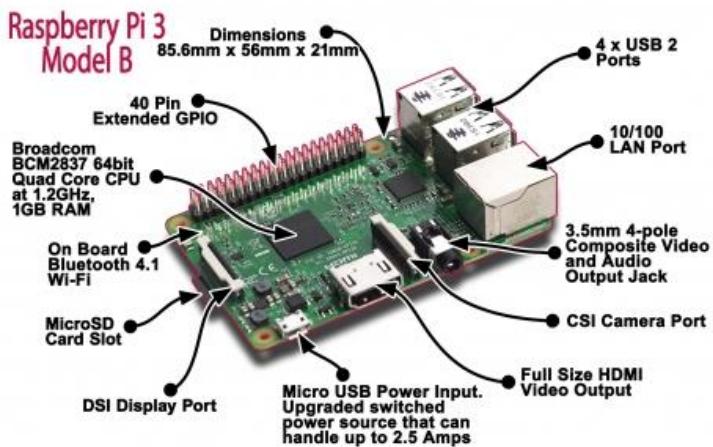
c. Raspbmc

d. XBMC

e. XBian

## **II.9.2 Raspberry Pi 3 Model B**

Raspberry Pi 3 Model B adalah generasi ketiga dari Raspberry Pi. Raspberry Pi model ini dapat digunakan untuk banyak aplikasi dan menggantikan Model Raspberry Pi B+ dan Raspberry Pi 2 Model B. Raspberry Pi 3 Model B membawa prosesor yang lebih kuat dan 10 kali lebih cepat dibandingkan dengan generasi pertama dari Raspberry Pi. Selain itu Raspberry Pi model ini juga menambahkan konektivitas *Wireless LAN* dan *Bluetooth* (Damayanti & Fradita, 2017). Komponen pada raspberry pi 3 dapat dilihat pada gambar 2.12.



**Gambar 2.12 Komponen Raspberry Pi 3 Model B**

([www.element14.com](http://www.element14.com))

### II.9.3 Spesifikasi Raspberry Pi 3 Model B

Adapun spesifikasi dari Raspberry Pi 3 Model B sebagai berikut :

1. *Chipset* Broadcom BCM2387
2. 1.2GHz *Quad-Core* ARM Cortex-A53
3. 802.11 bgn *Wireless LAN* dan *Bluetooth* 4.1 (*Bluetooth Classic* dan LE)
4. 1 GB RAM
5. 64 Bit CPU 64
6. 4 x *USB Ports*
7. 4 Pole *Stereo output* dan *Composite video port*
8. *Full Size HDMI*
9. 10/100 *BaseT Ethernet Socket*
10. *Port* kamera CSI untuk menghubungkan kamera Raspberry Pi
11. *Port* tampilan DSI untuk menghubungkan tampilan layar sentuh Raspberry Pi

12. Port Micro SD untuk memuat system operasi dan menyimpan data
13. Sumber daya *micro USB*

## **II.10 Web Server**

*Web* adalah tampilan pada *browser* dengan alamat domain khusus untuk sistem penelitian ini. *Web* dapat dibangun dengan menggunakan bahasa HTML dan PHP dengan *style* tampilan menggunakan bahasa CSS. *Web* tersebut disimpan pada satu komputer yang disebut *server*. *Server* menyimpan *program web* dan *database* untuk dapat diakses oleh *admin* atau *client* dari *browser*. *Website* dapat dibangun menggunakan program *notepad/notepad++*, *adobe dreasweaver*, atau *Sublime*.

*Web server* adalah sebuah *software* yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP atau HTTPS pada *client* yang dikenal dan biasanya kita kenal dengan nama *web browser* dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman *web* dan pada umumnya akan berbentuk dokumen HTML. Dalam bentuk sederhana *web server* akan mengirim data HTML kepada permintaan *web browser* sehingga akan terlihat seperti pada umumnya yaitu sebuah tampilan *website* (Damayanti & Fradita, 2017).

Fungsi utama *web server* adalah untuk melakukan atau akan transfer berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sedemikian rupa. Halaman *web* yang diminta terdiri dari berkas teks, video, gambar, file dan banyak lagi. pemanfaatan *web server* berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman *web* termasuk

yang di dalam berupa teks, video, gambar atau banyak lagi (Damayanti & Fradita, 2017).

Beberapa jenis *Web Server* diantaranya adalah:

1. Apache *Web Server / The HTTP Web Server*
2. Apache Tomcat
3. Microsoft windows *Server 2008 IIS (Internet Information Services)*
4. Lighttpd
5. Zeus *Web Server*
6. Sun Java System *Web Server*

Dari daftar list di atas yang paling dominan digunakan hanya Apache dan Microsoft *Windows Server*.

### **II.10.1 *Web Server Apache***

*Apache* merupakan *web server* yang paling banyak dipergunakan di Internet. Program ini pertama kali didesain untuk sistem operasi lingkungan UNIX. Namun demikian, pada beberapa versi berikutnya *Apache* mengeluarkan programnya yang dapat dijalankan di Windows NT. *Apache* mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunanya (Damayanti & Fradita, 2017).

Beberapa dukungan Apache sebagai berikut :

1. Kontrol Akses
2. CGI (*Common Gateway Interface*)
3. PHP (*Personal Home Page / PHP Hypertext Processor*)
4. SSI (*Server Side Includes*)

### **II.10.1.1 Keunggulan *Web Server Apache***

*Web Server Apache* mempunyai kelebihan dari beberapa pertimbangan di atas (Damayanti & Fradita, 2017):

1. Apache termasuk dalam kategori *freeware*.
2. Apache mudah sekali proses instalasinya jika dibandingkan *web server* lainnya seperti NCSA, IIS, dan lain-lain.
3. Mampu beroperasi pada berbagai platform sistem operasi.
4. Mudah mengatur konfigurasinya, Apache mempunyai hanya empat file konfigurasi.
5. Mudah dalam menambahkan *peripheral* lainnya ke dalam *platform web server*-nya.

### **II.10.1.2 Fasilitas dari *Apache***

Fasilitas atau ciri khas dari *web server Apache* adalah (Damayanti & Fradita, 2017) :

1. Dapat dijadikan pengganti bagi NCSA *Web server*.
2. Perbaikan terhadap kerusakan dan *error* pada NCSA v1.3 dan v1.4.
3. Apache merespon *web client* sangat cepat jauh melebihi NCSA.
4. Mampu di kompilasi sesuai dengan spesifikasi HTTP yang sekarang.
5. Apache menyediakan fitur untuk *multihomed* dan *virtual server*.
6. Kita dapat menetapkan respon error yang akan dikirim *web server* dengan menggunakan file atau skrip.
7. *Server apache* dapat otomatis berkomunikasi dengan *client browser*nya untuk menampilkan tampilan terbaik pada *client browser*nya. *Web*

*server* Apache secara otomatis menjalankan file *index.html*, halaman utama untuk ditampilkan secara otomatis pada *client*-nya.

8. *Web server* Apache mempunyai level-level pengamanan.
9. Apache mempunyai komponen dasar terbanyak di antara *web server* lainnya.
10. Ditinjau dari segi sejarah perkembangan dan prospeknya, Apache *web server* mempunyai prospek yang cerah. Apache berasal dari *web server* NCSA yang kemudian dikembangkan karena NCSA masih mempunyai kekurangan di bidang kompatibilitasnya dengan sistem operasi lain. Sampai saat ini, *web server* Apache terus dikembangkan oleh tim dari *apache.org*.
11. Performasi dan konsumsi sumber daya dari *web server* Apache tidak terlalu banyak, hanya sekitar 20MB untuk file-file dasarnya dan setiap *daemon*-nya hanya memerlukan sekitar 950KB memori per *child*.
12. Mendukung transaksi yang aman (*Secure Transaction*) menggunakan SSL (*Secure Socket Layer*).
13. Mempunyai dukungan teknis melalui *web*.
14. Mempunyai kompatibilitas *platform* yang tinggi.
15. Mendukung *third party* berupa modul-modul tambahan.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **III.1 Tempat dan Waktu Penelitian**

Penelitian ini dilaksanakan di Laboratorium *Internet of Things* dan *Parallel Computing* (IoT-PC), Departemen Teknik Informatika Fakultas Teknik, Universitas Hasanuddin. Kelurahan Romang Lompoa, Kecamatan Bontomarannu, Kabupaten Gowa. Sedangkan waktu penelitian dilaksanakan pada bulan April 2018 – Juni 2019.

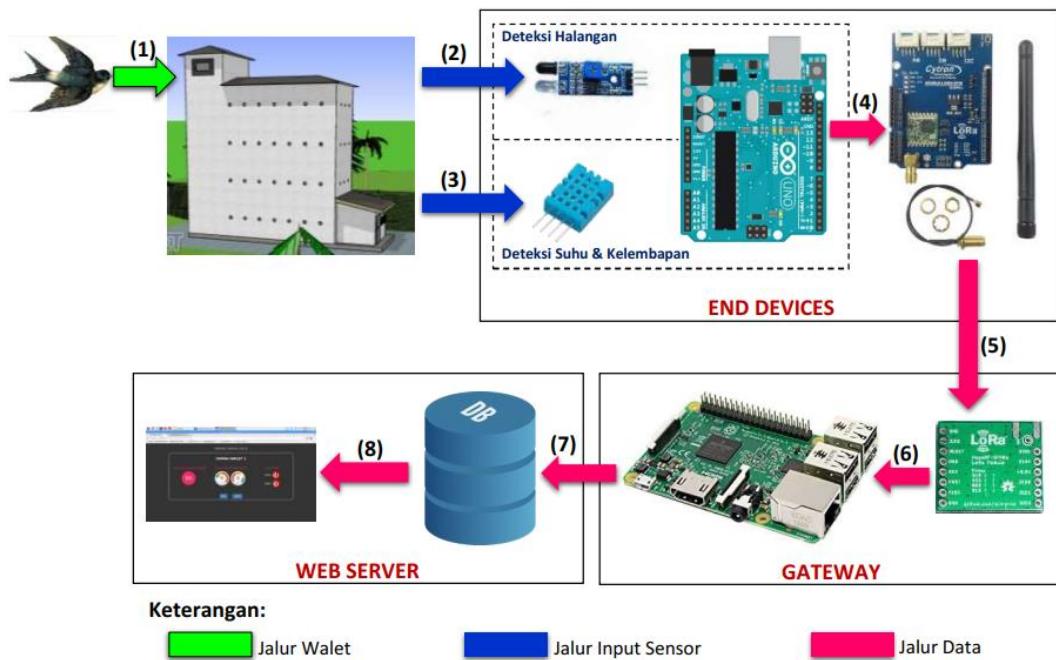
#### **III.2 Alat dan Bahan Penelitian**

Alat yang digunakan pada penelitian ini yaitu *solder* listrik, obeng, lem tembak, perangkat komputer, perangkat lunak arduino, perangkat lunak *browser*, *database* mysql, Apache, sistem operasi Raspberry Pi, alat tulis, dan *Multimeter*.

Sedangkan bahan yang digunakan yaitu Arduino Uno R3, Raspberry Pi 3 Model B, LoRa *Receiver* dan *Transmitter*, Kabel *Jumper*, *Micro SD*, Sensor DHT11 (suhu dan kelembapan udara), *transmitter* Infra merah (pemancar), *receiver* infra merah (penerima), Dioda IN4148, PCB, Timah.

#### **III.3 Kriteria Desain**

Sistem rumah walet cerdas berbasis *Internet of Things* ini dirancang untuk dapat memantau kondisi suhu dan kelembapan rumah walet sekaligus memantau jumlah populasi walet di rumah walet secara jarak jauh.



**Gambar 3.1** Gambaran Umum Cara Kerja Sistem

Dari Gambar 3.1 gambaran umum penelitian ini dapat dilihat alur kerja secara umum sebagai berikut:

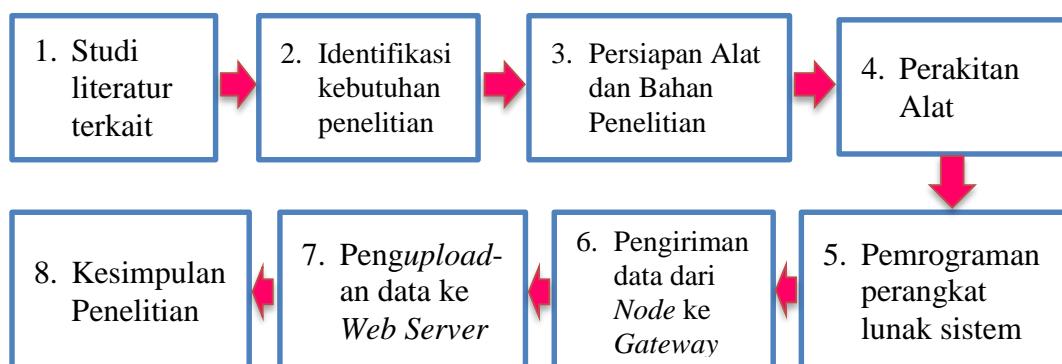
- (1). Walet keluar masuk rumah walet.
- (2). Penerima infra merah akan mendeteksi halangan yang ditimbulkan walet lalu akan diproses menggunakan Arduino untuk didapatkan jumlah populasi walet di rumah walet.
- (3). Sensor DHT11 akan mendeteksi suhu dan kelembapan rumah walet.
- (4). Data populasi, suhu dan kelembapan yang telah diproses akan dikirim ke LoRa menggunakan komunikasi data Serial SPI (*Serial Peripheral Interface*).

- (5). LoRa *Node* akan mengirim data ke LoRa *Gateway* secara nirkabel menggunakan Gelombang Radio dengan frekuensi 915 MHz.
- (6). Paket yang diterima oleh LoRa gateway akan diteruskan ke Mini PC Raspberry Pi menggunakan jalur komunikasi data Serial SPI.
- (7). Data yang telah diolah akan dikirim ke *database Web Server* menggunakan metode *HTTP Request*.
- (8). Data terakhir yang masuk di *database* akan di tampilkan pada halaman *Web*.

Dengan alur kerja diatas, data suhu, kelembapan, dan populasi didalam rumah walet akan ditampilkan pada halaman *web* sehingga dapat diakses secara jarak jauh. Pembudidaya dapat memantau data suhu, kelembapan dan populasi walet di dalam rumah walet yang dapat diakses pada PC ataupun *gadget smartphone* melalui *Web Browser*.

#### **III.4 Prosedur Penelitian**

Tahapan pada penelitian ini sebagaimana ditunjukkan pada gambar 3.2.



**Gambar 3.2** Diagram tahapan penelitian

Tahapan secara garis besar dijelaskan sebagai berikut :

1. Pada studi literatur, pencarian penelitian dilakukan terkait rancang bangun sistem rumah wallet cerdas berbasis *Internet of Things*. Pada tahap ini juga dilakukan pencarian dokumentasi hasil penelitian-penelitian sebelumnya yang terkait dengan topik penelitian.
2. Pada tahap ini, dilakukan penetapan berbagai kebutuhan penelitian dan disiapkan guna untuk menunjang penelitian dan perancangan sistem serta pembuatan aplikasi.
3. Pada tahap ini dilakukan pembelian alat dan bahan yang dibutuhkan untuk penelitian, seperti *transmitter* dan *receiver* infra merah, kabel *jumper*, dan lain-lain.
4. Pada tahap ini, dilakukan pembuatan dan perangkaian alat pada dua sistem yaitu sistem pembacaan sensor dan pengiriman data atau biasa disebut *end devices* yang terdiri dari Arduino Uno R3, LoRa *Shield* Cytron, sensor DHT11, *transmitter* dan *receiver* Infra merah, sedangkan sistem penerima data atau biasa disebut perangkat *gateway* terdiri dari Raspberry Pi 3 Model B dan Lora HopeRF.
5. Pada tahap ini, pemrograman sistem perangkat lunak menggunakan bahasa pemrograman C, C++, Python, PHP, SQLi dengan tahapan pemrograman mulai dari pembacaan data sensor dari *End Devices* lalu dikirim ke *Gateway* untuk ditampilkan pada halaman *Web*.
6. Pada tahap ini, dilakukan proses pengiriman data yang telah diolah, LoRa *Node* akan mengirim paket data ke LoRa *Gateway* setiap ±60 detik.
7. Pada tahap ini, data yang telah diterima oleh sistem *Gateway* akan disimpan pada *database web server* lalu data akan ditampilkan pada halaman *web*.

8. Setelah melakukan tahapan-tahapan di atas, diperolehlah kesimpulan berdasarkan penelitian yang telah dilakukan.

### **III.5 Tahap Persiapan**

Tahap persiapan merupakan tahapan awal dalam melakukan penelitian, pada tahap ini penulis melakukan studi literatur dengan mencari berbagai acuan baik melalui buku, jurnal, tugas akhir maupun artikel dengan nara sumber yang jelas dan terpercaya dengan tujuan untuk melengkapi literature mengenai penelitian ini. Penulis juga melakukan identifikasi masalah pada penelitian ini, membaca dan memahami kelebihan dan kekurangan dari penelitian yang sudah dilakukan sebelumnya. Serta penulis menyiapkan alat dan bahan yang diperlukan dalam penelitian ini untuk mempersiapkan menuju ke tahap selanjutnya.

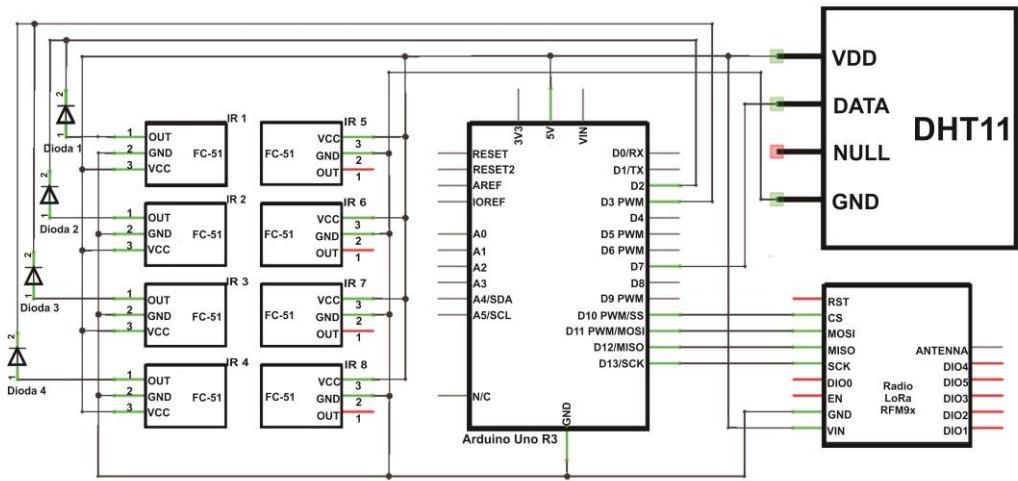
### **III.6 Tahap Pembuatan Sistem**

Tahap pembuatan sistem dibagi menjadi tiga tahap, yakni tahap perancangan sistem *Hardware* (Perangkat Keras), tahap perakitan / perwujudan alat, dan tahap pembuatan *Software* (Perangkat Lunak). Berikut penjabaran dari masing-masing tahapan.

#### **III.6.1 Tahap Perancangan Sistem Hardware (Perangkat Keras).**

Tahap perancangan sistem perangkat keras terdiri dari dua perancangan yaitu :

1. Perancangan rangkaian *End Devices*. Skematik dari rangkaian *end devices* ini bisa dilihat pada gambar 3.3 berikut.



**Gambar 3.3 Skematik Rangkaian *End Devices***

Perancangan skematik rangkaian *end devices* ini menggunakan aplikasi *Fritzing* dimana bertujuan untuk mempermudah saat proses perangkaian alat. dari Gambar 3.3, dalam menjalankan sistem terdapat beberapa perangkat elektronik yang memiliki fungsinya masing-masing. Daftar komponen elektronik yang digunakan bisa dilihat pada table 3.1 berikut.

**Tabel 3.1** Daftar Komponen Sistem *End Device*

No.	Nama Komponen	Kuantitas
1	Arduino Uno R3	1
2	Sensor DHT11	1
3	Transceiver Infra Merah	8
4	LoRa Shield Cytron	1
5	Dioda IN4148	4

2. Perancangan rangkaian sistem *Gateway*. Sistem *gateway* berfungsi sebagai penerima data dan pengolah data sehingga data dapat divisualisasikan pada *web server*. Perangkat yang digunakan pada sistem *gateway* ini adalah Mini *PC* Raspberry Pi 3 Model B dan LoRa *HopeRF*. Skematik rangkaian sistem *gateway* ini dapat dilihat pada gambar 3.9.

### **III.6.1.1 Arduino Uno R.3**

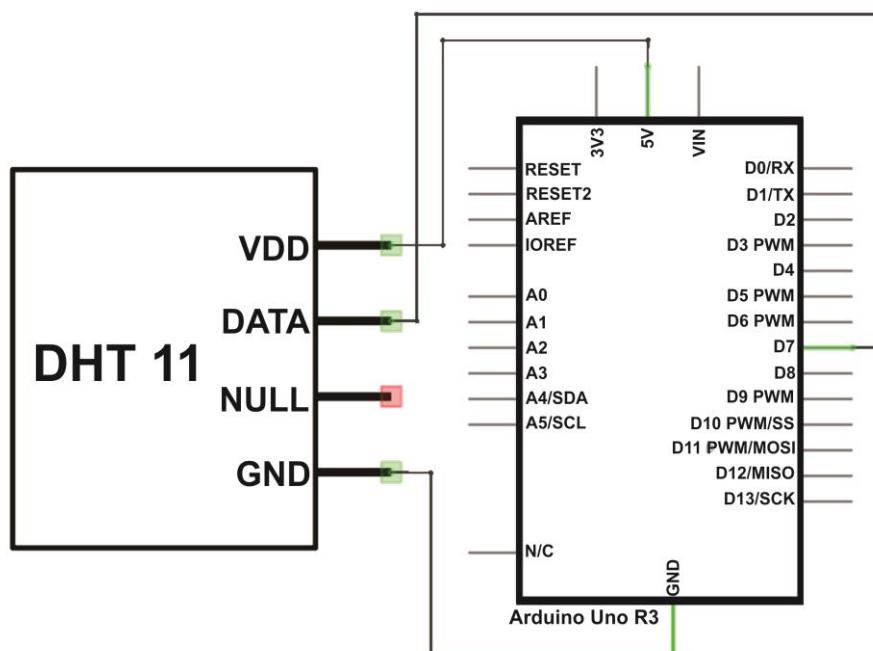
Mikrokontroler Arduino Uno Revisi 3 adalah otak atau pengendali seluruh perangkat dari *end devices* yang digunakan untuk mengontrol, dan memproses data sebelum dikirim ke *gateway* serta menjadi sumber daya untuk perangkat lain yang terdapat pada *end devices* seperti sensor DHT11, pemancar dan penerima Infra merah , dan *LoRa Shield Cytron*. Sifat programnya yang menggunakan metode *looping* memudahkan penggunaan mikrokontroler ini dalam pembuatan sistem automatisasi dalam penelitian ini.

### **III.6.1.2 Sensor DHT11**

Sensor DHT11 digunakan sebagai alat pengukur suhu dan kelembapan udara di rumah walet. Sensor DHT11 yang digunakan memiliki 4 kaki yang 3 kaki diantaranya di hubungkan ke pin Arduino Uno. *Wiring* antara DHT11 dan Arduino dapat dilihat pada table 3.2 dan skematik penghubungannya dapat dilihat pada gambar 3.4.

**Tabel 3.2 Wiring DHT11 dan Arduino Uno**

DHT11	Arduino Uno
VCC	VCC (5v)
DATA	D7
GND	GND

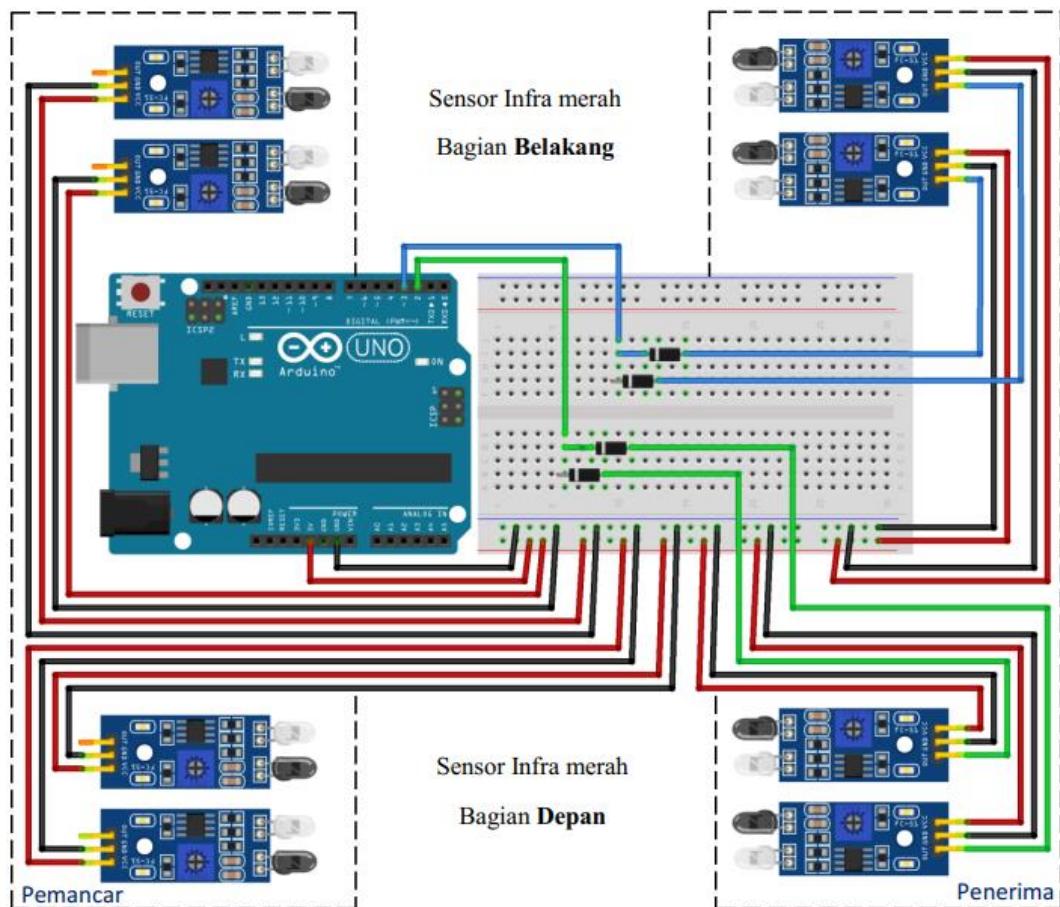


**Gambar 3.4 Skematik deteksi Suhu dan Kelembapan**

### III.6.1.3 Transceiver Infra merah (Deteksi Halangan)

Pada penelitian ini *Transceiver* Infra merah digunakan sebagai alat pendeteksi halangan yang dihasilkan walet saat keluar dan masuk rumah walet. *Transmitter* infra merah memancarkan infra merah ke arah *receiver* infra merah sehingga ketika walet melewati sinar infra merah tersebut, maka *receiver* infra merah tidak menerima sinar infra merah sehingga dapat ditentukan arah walet masuk ataupun keluar. Dari proses tersebut didapatkan jumlah populasi walet

dalam rumah walet. Rangkaian penghubungan antara pemancar serta penerima infra merah pada Arduino dapat dilihat pada gambar 3.5.

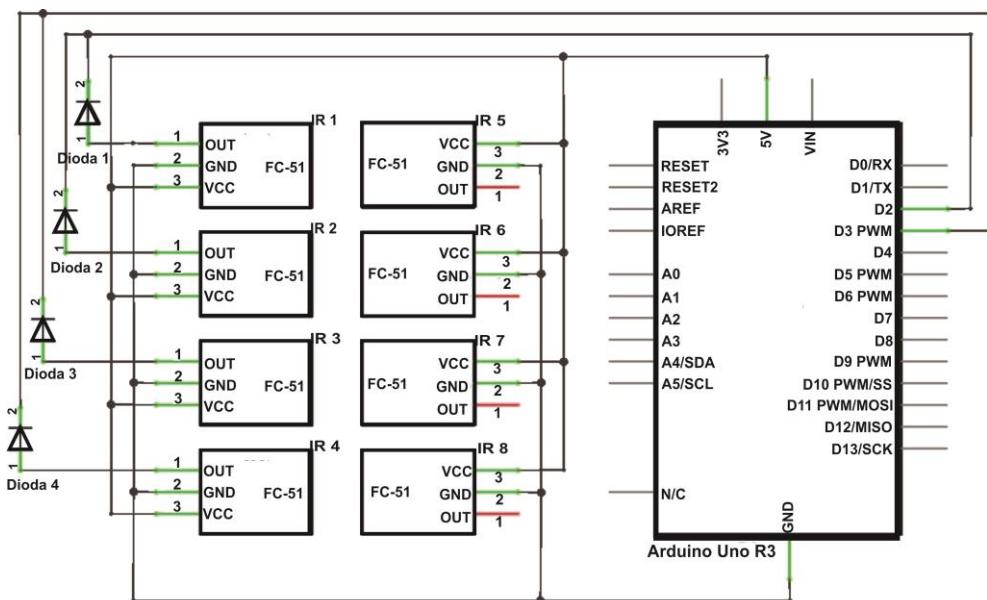


**Gambar 3.5** Rangkaian pendeksi Halangan

Dari gambar 3.5 dapat dilihat bahwa alat yang digunakan 4 buah *transmitter* infra merah (pemancar), bisa dilihat pada bagian kiri gambar 3.5. Dan 4 buah digunakan sebagai *receiver* infra merah (penerima), bisa dilihat pada bagian kanan gambar 3.5.

Untuk 4 pemancar infra merah, pin yang dihubungkan ke Arduino adalah pin VCC ke VCC 5 Volt dan GND ke GND Arduino Uno. Sedangkan 4 penerima

infra merah, pin VCC di hubungkan ke VCC 5 Volt, pin GND dihubungkan ke GND Arduino Uno dan pin DATA untuk 2 penerima infra merah bagian depan di hubungkan ke dioda IN4148 tiap pin datanya, lalu kedua dioda IN4148 dihubungkan ke pin D2 Arduino Uno yang dimana merupakan pin INTERRUPT0. Sedangkan pin DATA untuk dua penerima infra merah bagian belakang di hubungkan juga ke dioda IN4148 tiap pin datanya lalu kedua dioda IN4148 di hubungkan ke pin D3 Arduino Uno yang dimana merupakan pin INTERRUPT1. Dioda IN4148 berfungsi sebagai gerbang logika *OR* untuk sinyal keluaran dua penerima infra merah yang hasil dari keluarannya dihubungkan ke satu pin input *interrupt*. Hal tersebut dilakukan dikarenakan Arduino Uno yang digunakan hanya memiliki dua pin *Interrupt* yaitu D2 dan D3, sedangkan penerima infra merah yang digunakan ada empat buah. Skematik rangkaian pendekksi halangan dapat dilihat pada gambar 3.6 berikut.



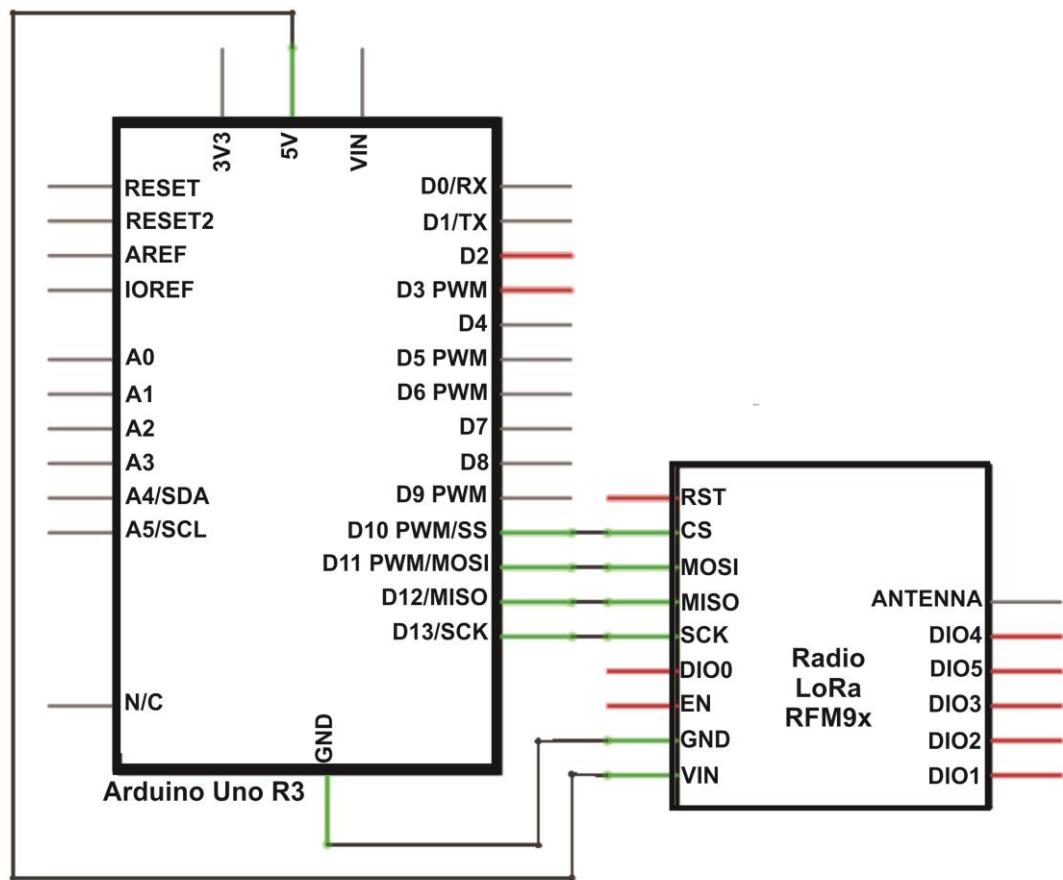
**Gambar 3.6** Skematik Rangkaian pendekksi Halangan

### **III.6.1.4 LoRa *Shield Cytron***

*LoRa Shield Cytron* adalah *board* teknologi LoRa dari Cytron yang dapat dipasang langsung pada papan Arduino. Pada penelitian ini *LoRa Shield* ini digunakan sebagai pengirim data suhu, kelembapan dan data populasi walet secara jarak jauh dengan data *rate* yang sangat rendah dengan frekuensi yang digunakan adalah 915MHz. *Wiring* yang digunakan dapat dilihat pada tabel 3.3 dan skematik rangkaian untuk *LoRa Shield Cytron* dan Arduino dapat dilihat pada gambar 3.8.

**Tabel 3.3** Wiring *LoRa Shield* dan Arduino Uno

<b>LoRa Shield Cytron</b>	<b>Arduino Uno</b>
VCC	VCC (5v)
GND	GND
SCK	D13 / SCK
MISO	D12 / MISO
MOSI	D11 / MOSI
D10 / SS	D10 / SS



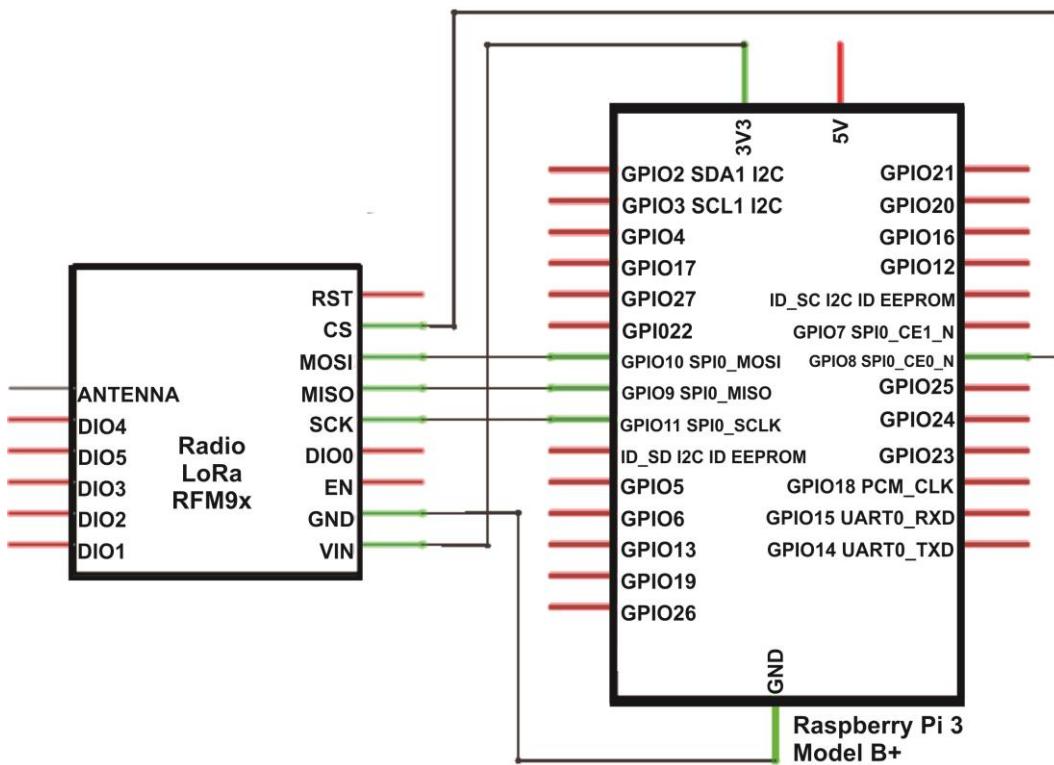
**Gambar 3.7** Skematik Rangkaian LoRa *Shield* dan Arduino

### III.6.1.5 LoRa HopeRF Modul

LoRa HopeRF Modul merupakan modul LoRa yang digunakan pada penelitian ini sebagai penerima data dari LoRa *Node*. LoRa HopeRF ini terhubung dengan Raspberry pi sebagai perangkat *gateway* pada penelitian ini. Untuk *Wiring* pada LoRa HopeRF ini dapat dilihat pada tabel 3.4 dan skematik rangkaian antara LoRa HopeRF dan Raspberry Pi dapat dilihat pada gambar 3.8.

**Tabel 3.4** Wiring LoRa HopeRF modul dengan Raspberry Pi 3

LoRa HopeRF Modul	Raspberry Pi 3
VCC (+3.3v)	VCC (+3.3v)
GND	GND
MOSI	GPIO 10 / MOSI
MISO	GPIO 9 / MISO
SS	GPIO 8 / SPI_CE0
SCK	GPIO 11 / SCLK



**Gambar 3.8** Skematik Rangkaian Sistem *Gateway*

### III.6.1.6 Raspberry Pi 3 Model B

Raspberry Pi merupakan komputer kecil yang memiliki input output digital port. GPIO merupakan terminal *bi-directional* yang ditambahkan ke Raspi

berjumlah 40 pin pada Raspberry Pi 3 Model B. pin GPIO (*General Purpose Input Output*) ini memiliki bagian-bagian yang mewakili 40 pin pada Raspberry Pi tersebut, antara lain : Power Supply (3.3Volt / 5Volt, 2 set), UART (*Universal asynchronous receiver/transmitter*, 1 set), SPI (*Serial Peripheral interface*), I2C (*Inter-integrated Circuit*). Tetapi untuk penelitian ini pin yang digunakan hanya beberapa yang dihubungkan ke LoRa HopeRF, pin yang digunakan adalah pin untuk komunikasi data serial (SPI). Wiring raspberry pi dengan perangkat lain dapat dilihat pada tabel 3.4.

Pada sistem *gateway* ini, Raspberry Pi 3 digunakan sebagai pengolah data yang diterima dari LoRa HopeRF dan memiliki fungsi sebagai *Web Server*.

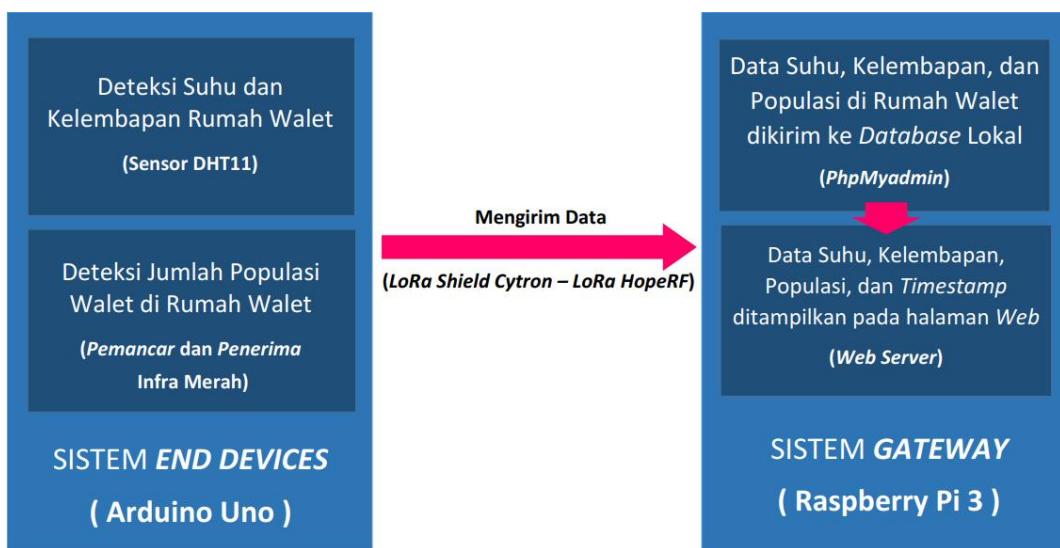
### **III.6.2 Tahap Perakitan / Perwujudan Alat**

Tahap perakitan / perwujudan alat ini merupakan tahap realisasi dari tahap perancangan alat. Cara perealisasianya yakni menghubungkan komponen-komponen sistem *end devices* serta sistem *gateway* menjadi satu kesatuan tiap sistem agar dapat di jalankan sesuai fungsi tertentu. Setelah itu pembuatan miniatur pintu masuk walet dimana berfungsi sebagai tempat atau wadah untuk sistem *end devices* yang telah di rakit.

Setelah miniatur pintu masuk walet dibuat, tahap selanjutnya adalah penempatan 4 pemancar infra merah dan 4 penerima infra merah dengan jarak antara tiap pemancar infra merah adalah 12 cm, begitupun jarak antara tiap penerima adalah 12 cm. Sedangkan jarak antara pemancar dan penerima infra merah adalah 40 cm.

### III.6.3 Tahap Pembuatan *Software* (Perangkat Lunak)

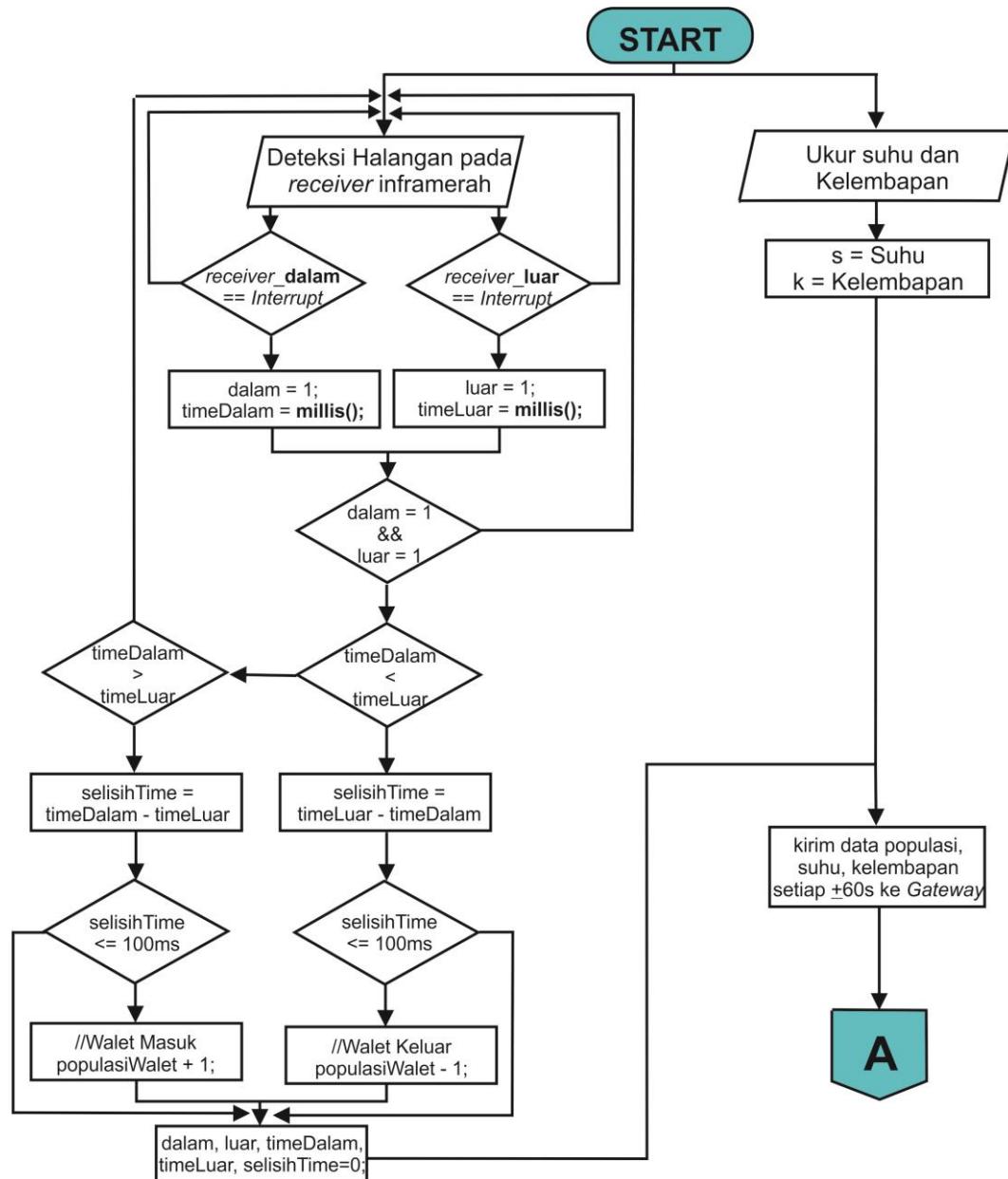
Tahap pembuatan perangkat lunak ini merupakan tahap pembuatan program untuk mengeksekusi rancangan perangkat keras yang telah dirakit. Dimana pembuatan program ini mengikuti blok diagram cara kerja alat yang bisa dilihat pada gambar 3.9 berikut.



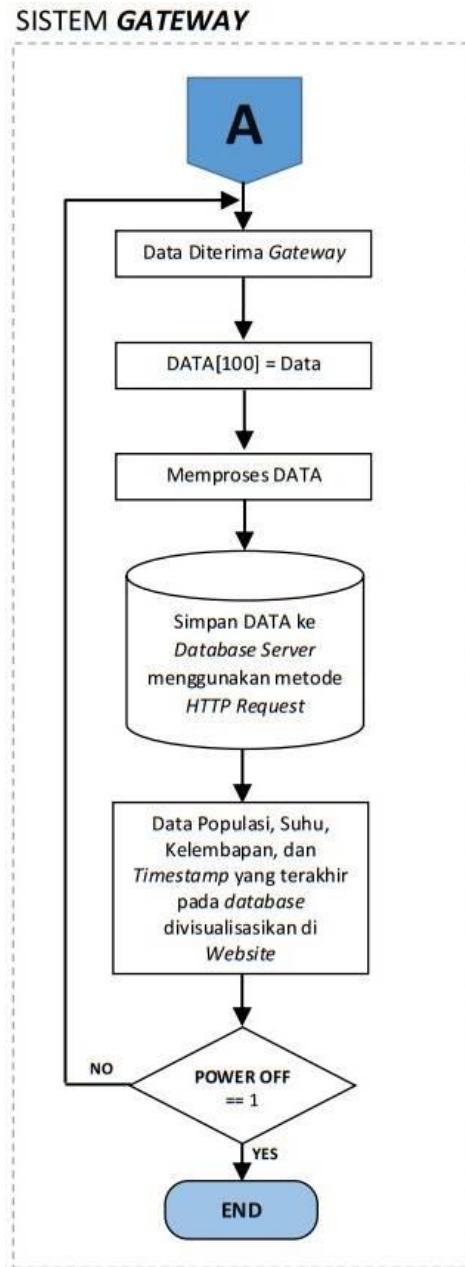
**Gambar 3.9** Blok Diagram Cara Kerja Alat

Pada gambar 3.9, dapat dilihat alur program yang dibuat, dimana terdapat dua sistem yaitu pertama sistem *End devices* yang menggunakan Arduino Uno dengan menggunakan bahasa pemrograman C serta menggunakan program *Arduino IDE* untuk mengeksekusi dan meng-*upload* program yang dibuat ke Mikrokontroler Arduino. Sedangkan sistem kedua adalah sistem *Gateway* yang menggunakan *Mini PC* Raspberry Pi 3 dengan Sistem operasi *Raspbian* dan menggunakan beberapa *library*. Bahasa pemrograman yang digunakan adalah

*Python*, C++, dan terdapat pula beberapa Bahasa program untuk mengeksekusi *web server* yaitu PHP, HTML, CSS, dan *Mysqli*.



**Gambar 3.10** Flowchart Sistem *End Devices* Rumah Walet Cerdas



**Gambar 3.11 Flowchart Sistem Gateway Rumah Walet Cerdas**

Pada gambar 3.10 dan 3.11 merupakan *flowchart* sistem rumah walet cerdas. Dapat dilihat proses awal pada *flowchart* adalah proses pembacaan data suhu dan kelembapan ruangan dan proses pendekripsi halangan sedangkan proses penghitung populasi dan pengiriman data ke *gateway* terjadi pada proses

*looping* program Arduino. Artinya dalam selang waktu 60 detik, proses-proses tersebut akan di eksekusi satu kali. Sedangkan untuk proses pendekripsi keluar atau masuknya walet menggunakan metode interupsi pada program Arduino. Dimana cara kerja interupsi ini adalah ketika program berjalan dan terjadi interupsi yang ditimbulkan karena adanya halangan yang terdeteksi oleh penerima infra merah, maka program yang sedang berjalan akan di hentikan sementara lalu akan menjalankan fungsi interupsi tersebut.

### **III.6.3.1 Deteksi Suhu dan Kelembapan**

Pada gambar 3.10 *flowchart* sistem *End Devices*, dapat dilihat *flowchart* untuk mendekripsi suhu dan kelembapan, dimana program akan menjalankan proses deteksi setiap selang waktu  $\pm 60$  detik. Proses pendekripsi menggunakan sensor DHT11, setelah itu data suhu dan kelembapan akan disimpan pada variabel.

### **III.6.3.2 Penghitung Populasi Walet**

*Flowchart* untuk penghitung populasi walet dapat dilihat pada gambar 3.10 dengan proses pertama adalah melakukan pendekripsi halangan yang disebabkan walet ketika melewati penerima infra merah. Ketika penerima infra merah mendekripsi adanya halangan maka terjadi proses interupsi. Proses interupsi ini akan menjalankan sebuah *function* diluar proses *loop* pada program arduino, ketika terjadi proses interupsi maka program yang sedang berjalan akan dihentikan sementara untuk menjalankan program interupsi tersebut. Program interupsi yang dijalankan berisikan pendeklarasian variabel dimana variabel ini

digunakan untuk menangkap total waktu dari ketika mikrokontroler arduino di nyalakan hingga terjadinya interupsi. Setelah program interupsi selesai dijalankan maka program akan melanjutkan kembali program yang telah dihentikan sebelumnya.

Program akan kembali ke proses pendekripsi halangan ketika salah satu atau tidak ada sama sekali penerima infra merah yang mendekripsi halangan dan melanjutkan ke proses selanjutnya ketika terdapat dua penerima infra merah mendekripsi adanya halangan maka proses selanjutnya adalah program untuk menentukan arah walet apakah itu masuk atau keluar, dimana kedua variabel waktu yang didapatkan dari proses interupsi akan saling dibandingkan. Ketika variabel waktu penerima infra merah bagian depan lebih besar dari pada variabel waktu penerima infra merah bagian belakang maka program akan melakukan proses pengurangan untuk mengetahui selisih waktu yang didapatkan. Proses selanjutnya adalah menghitung selisih waktu yang didapatkan dari dua penerima infra merah yang mendekripsi adanya halangan. Apabila selisih waktu kurang dari 100ms maka program akan melanjutkan ke proses selanjutnya yaitu variabel populasi akan ditambah 1 atau walet terdeteksi masuk. Begitupun sebaliknya, ketika variabel waktu penerima infra merah bagian depan lebih kecil dari pada variabel waktu penerima infra merah bagian belakang maka program akan melakukan proses pengurangan untuk mengetahui selisih waktu yang didapatkan. Apabila selisih waktu kurang dari 100ms maka program akan melanjutkan ke proses selanjutnya yaitu variabel populasi akan dikurang 1 atau walet terdeteksi

keluar. Ketika selisih waktu yang didapatkan melebihi 100ms maka seluruh variabel yang digunakan akan dikembalikan ke nilai awal.

### **III.6.3.3 Sistem Transmisi Data**

Pada gambar 3.10 dan gambar 3.11 dapat dilihat *flowchart* sistem transmisi data, dimana data populasi, suhu dan kelembapan yang didapatkan dari proses sebelumnya akan dikirim ke *gateway* setiap  $\pm 60$  detik. Proses selanjutnya adalah data diterima *gateway*. Program transmisi data ini menggunakan *library SX1272* pada Arduino dan Raspberry Pi.

### **III.6.3.4 Visualisasi Data pada Web Server**

*Flowchart* visualisasi data yang diterima oleh *gateway* ke *web server* dapat dilihat pada gambar 3.11. Dimana proses pertama adalah menyimpan paket data yang bertipe *string* ke variabel bertipe *array* satu dimensi agar dapat dilanjutkan pada proses selanjutnya. Proses selanjutnya adalah melakukan pemotongan elemen *array* data. Data yang telah diproses akan dikirim ke *database web server* menggunakan metode *HTTP Request*. Data terakhir pada *database* akan ditampilkan pada halaman web. Halaman web akan otomatis memuat ulang setiap 20 detik.

## BAB IV

### HASIL DAN PEMBAHASAN

#### IV.1 Hasil Rancang Bangun Alat

Setelah semua desain rancangan dan komponen penyusunnya telah selesai dibuat maka tahap selanjutnya adalah perakitan komponen-komponen sistem *end devices*, sistem *node*, serta sistem *gateway* menjadi satu kesatuan tiap sistemnya pada *box* miniatur pintu masuk walet yang telah dibuat. Sehingga dapat menjalankan fungsi sebagaimana blok diagram pada gambar 3.9.



**Gambar 4.1** Box Miniatur Pintu Keluar dan Masuk Walet

Pada gambar 4.1, *Box* miniatur pintu masuk walet ini berfungsi sebagai tempat atau wadah untuk rangkaian serta sensor-sensor yang digunakan. Sehingga dapat berfungsi sebagaimana mestinya.



**Gambar 4.2** Rangkaian Sistem *End Devices*

Pada gambar 4.2 dapat dilihat merupakan rangkaian sistem *end devices*, dimana terdapat 4 penerima infra merah yang sejajar dengan 4 pemancar infra merah dimana perangkat-perangkat tersebut serta sensor DHT11, LoRa *Shield Cytron* terhubung ke Arduino Uno sebagai *Node A*.

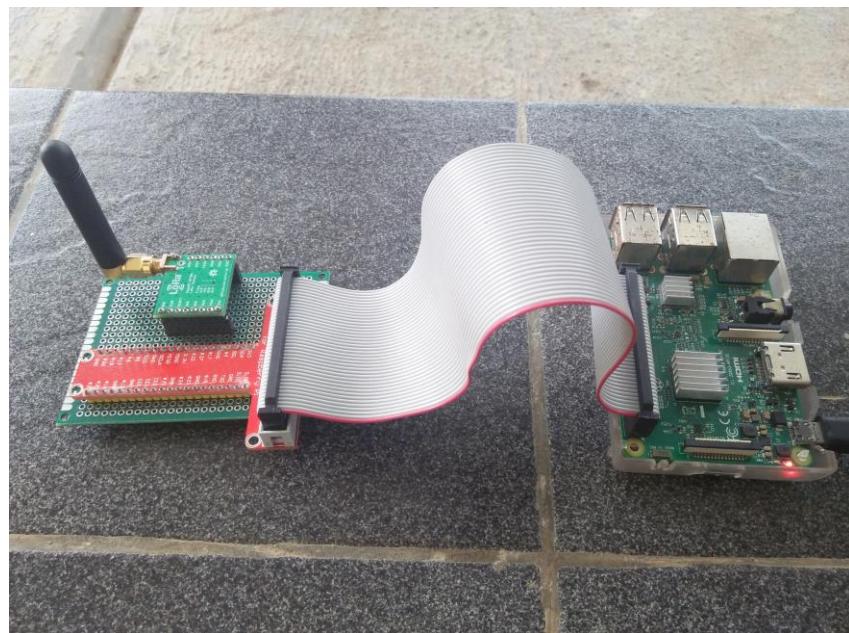


**Gambar 4.3** Rangkaian Sistem *Node A*



**Gambar 4.4** Rangkaian Sistem *Node B*

Pada gambar 4.3 dapat dilihat merupakan rangkaian sistem *Node A*, dimana terdapat LoRa HopeRF yang terhubung pada Arduino Mega. Sedangkan pada gambar 4.4 merupakan sistem *Node B*, dengan menggunakan LoRa HopeRF yang terhubung pada Arduino Uno.



**Gambar 4.5** Rangkaian Sistem *Gateway*

Gambar 4.5 merupakan rangkaian sistem *gateway* yang menunjukkan rangkaian raspberry pi 3 dengan LoRa HopeRF.

## IV.2 Pengujian Sistem

Pengujian sistem yang dilakukan meliputi pengujian sistem pembacaan suhu dan kelembapan menggunakan sensor DHT11, kemudian pengujian sistem penghitung populasi walet menggunakan pemancar dan penerima Infra merah, dan pengujian sistem transmisi data serta pengujian visualisasi data pada halaman *web*. Pengujian sistem ini bertujuan untuk mengetahui sistem dapat berjalan sesuai *listing* program yang telah diprogramkan.

### IV.2.1 Pengujian Sistem Pembacaan Suhu dan Kelembapan

Pengujian sistem pembacaan suhu dan kelembapan ini menggunakan sensor DHT11.

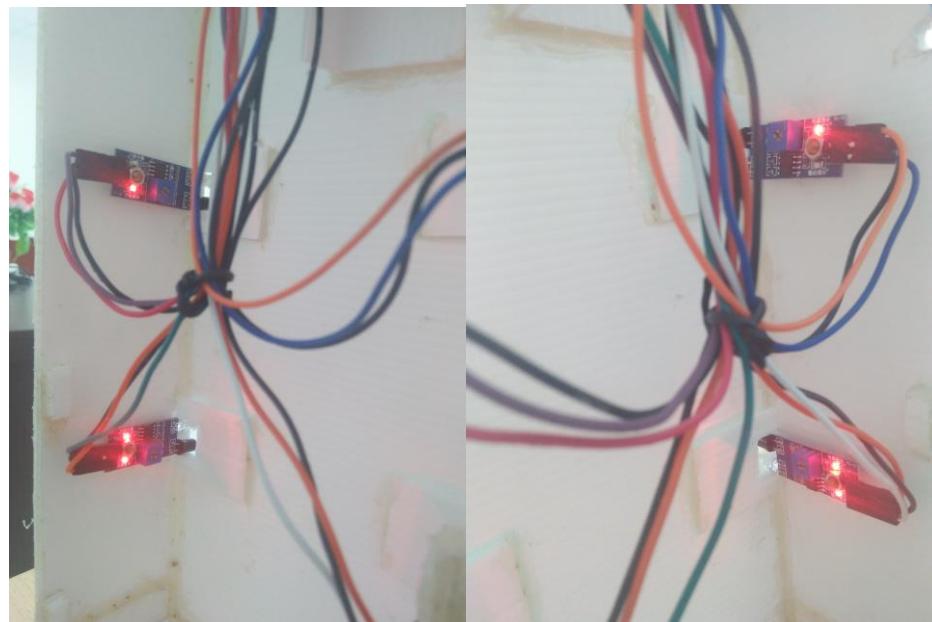
**Sending** /new\_record.php?IR=11&TR=25&HU=67//

**Gambar 4.6** Tampilan Data yang dikirim *Node* pada Serial Monitor

Pada gambar 4.5 menunjukkan pembacaan data suhu dan kelembapan dari sensor DHT11 pada sistem *node* yang dikirim ke *gateway*. Dimana “TR” merupakan label dari *Temperature* dengan data suhu yaitu 25<sup>0</sup>C, sedangkan “HU” merupakan label dari *Humidity* dengan data kelembapan 67%.

#### IV.2.2 Pengujian Sistem Penghitung Populasi Walet

Pengujian pada sistem penghitung populasi walet ini dilakukan dengan cara melemparkan papan berwarna hitam dengan ukuran 10 x 6 cm ke arah dalam maupun ke arah luar.



**Gambar 4.7** Penerima Infra merah Mendeteksi Halangan

Pada gambar 4.7 dapat dilihat bahwa dua penerima infra merah bagian atas mendeteksi adanya halangan sehingga program akan menjalankan proses interupsi untuk mengetahui total waktu dimulainya program hingga terjadinya interupsi, sehingga dapat diketahui halangan yang terdeteksi keluar atau masuk rumah walet.

```
pinMode (interruptPin2, INPUT);
attachInterrupt (digitalPinToInterrupt (interruptPin2), sensorDepan, RISING);

pinMode (interruptPin3, INPUT);
attachInterrupt (digitalPinToInterrupt (interruptPin3), sensorBelakang, RISING);
```

Kode program diatas merupakan kode program yang digunakan untuk mendeteksi adanya kenaikan sinyal listrik pada siklus jam ketika penerima infra merah mendeteksi halangan. Ketika pin interupsi D2 atau D3 pada Arduino Uno mendeteksi halangan maka akan menjalankan fungsi *sensorDepan()* atau *sensorBelakang()*, dimana isi dari fungsi tersebut bisa dilihat pada kode program dibawah.

```
void sensorDepan() {  
    timeSensorDepan = millis();  
    dataA=1;  
}  
  
void sensorBelakang() {  
    timeSensorBelakang = millis();  
    dataB=1;  
}
```

Ketika interupsi terjadi, maka program akan menangkap waktu dimulainya program hingga terjadinya interupsi dan pemberian nilai “1” pada variabel *dataA* atau *dataB*. Waktu yang dibutuhkan sebuah proses interupsi untuk mengeksekusi fungsi tersebut adalah kurang dari 1 millidetik. Proses interupsi ini terjadi diluar program utama pada Arduino.

```

unsigned long currentMillis1 = millis();

if (currentMillis1 - previousMillis1 >= interval1){
    previousMillis1 = currentMillis1;

    if (dataA==1 && dataB==1){
        if (timeSensorDepan < timeSensorBelakang){
            selisih_time=timeSensorBelakang-timeSensorDepan;
            if (selisih_time<=100){
                terdeteksi=terdeteksi+1;
                timeSensorDepan = 0;
                timeSensorBelakang = 0;
                selisih_time = 0;
                dataA = 0;
                dataB = 0;
            } else {
                timeSensorDepan = 0;
                timeSensorBelakang = 0;
                selisih_time = 0;
                dataA = 0;
                dataB = 0;
            }
        } if (timeSensorBelakang < timeSensorDepan){
            selisih_time=timeSensorDepan-timeSensorBelakang;
            if (selisih_time<=100){
                if (terdeteksi > 0){
                    terdeteksi=terdeteksi-1;
                } else {
                    terdeteksi=0;
                }
                timeSensorDepan = 0;
                timeSensorBelakang = 0;
                selisih_time = 0;
                dataA = 0;
                dataB = 0;
            } else {
                timeSensorDepan = 0;
                timeSensorBelakang = 0;
                selisih_time = 0;
                dataA = 0;
                dataB = 0;
            }
        }
    }
}

```

Kode program diatas digunakan untuk mendeteksi walet keluar atau masuk rumah walet. Proses tersebut terjadi didalam fungsi *Loop*.

Dapat dilihat pada gambar 4.6 tampilan pengiriman data pada serial monitor *node*, hasil dari pembacaan data pada sistem penghitung populasi walet. Dimana

sensor infra merah mendeteksi proses halangan yang ditimbulkan selama selang waktu 1 menit, dimana data populasi didalam rumah walet yang terdeteksi adalah 11 dengan label “IR”. Selanjutnya data-data tersebut akan dikirim ke *gateway*.

#### IV.2.3 Pengujian Sistem Transmisi Data

Proses pengiriman data dari sistem *node* dilakukan setiap selang waktu +60s. Proses pengiriman dapat dilihat pada gambar 4.8.

```
Reading
Sending /new_record.php?IR=11&TR=25&hu=67//
Real payload size is 35
--> CS1
--> CAD 549
OK1
--> RSSI -127
LoRa pkt seq 126
LoRa Sent in 252
LoRa Sent w/CAD in 803
Packet sent, state 0
130490
Will send next value at
192485
```

**Gambar 4.8** Proses Pengiriman Paket Data pada Serial Monitor *Node*

Sedangkan pada sistem *gateway* akan mendeteksi ketika terdapat paket data yang dikirim oleh sistem *node* pada frequensi 915MHz, dan menerima data tersebut untuk diolah lalu dikirim ke *database*.

```
^$Low-level gw status ON
--- rxlora. Dst=1 type=0x12 src=8 seq=120 len=38 SNR=8 RSSIpkt=-43 BW=125 CR=4/5 SF=12
^p1,18,8,120,38,8,-43
^r125,5,12
^t2018-09-21T14:09:19.045
??/new_record.php?IR=0&TR=26&HU=66// ← A      /*Data Mentah*/
/new_record.php?IR=0&TR=26&HU=66//    ← B      /*Data Setelah di Proses*/
BEGIN (200 OK)
COMPLETE (1 bytes)
```

**Gambar 4.9** Proses Penerimaan Paket Data pada Sistem *Gateway*

Pada gambar 4.9 dapat dilihat proses penerimaan data dimana *Point A* merupakan paket data mentah yang diterima, sedangkan *Point B* merupakan data mentah yang telah di proses menjadi data yang siap dikirim ke *database web server*.

#### IV.2.4 Pengujian Visualisasi Data Pada Web Server

Pengujian visualisasi data pada *web server* berupa pengujian penampilan data pada halaman *web*. Data yang diterima oleh sistem *gateway* akan di proses dengan menghapus elemen *array* pada data mentah yang diterima dari *node*, proses tersebut untuk menghapus karakter tidak penting yang ikut dalam paket data yang di kirim oleh sistem *node*. Berikut kode program untuk menghapus elemen *array* data.

```
const char * str = my_data;

for (int d=3; d<38; ++d){
    my_data[d]=my_data[d+1];
}
for (int e=2; d<37; ++e){
    my_data[e]=my_data[e+1];
}
for (int f=1; f<36; ++f){
    my_data[f]=my_data[f+1];
}
for (int g=0; g<35; ++g){
    my_data[g]=my_data[g+1];
}

char y[100];
y[0]=^0';
strncpy(y,my_data,35);
PRINT_STR("%s",y);
PRINTLN;
```

Data yang telah melalui proses penghapusan elemen *array*, selanjutnya akan dikirim ke *database* menggunakan metode *HTTP Request*. *HTTP Request* untuk Bahasa pemrograman C++ menggunakan *library HappyHTTP*. Berikut kode program untuk proses pengiriman data ke *database* menggunakan metode *HTTP Request*.

```

happyhttp::Connection conn( "localhost", 80 );
conn.setcallbacks( OnBegin, OnData, OnComplete, 0 );

conn.request( "GET", y );

while (conn.outstanding())
conn.pump();

```

Data berupa *Link* HTTP yang di kirim ke *database* menggunakan *HTTP Request*. Yang selanjutnya akan melalui proses *Parsing* oleh API PHP. Untuk mendapatkan data *integer* yang akan disimpan pada *database*. Berikut kode program untuk program *parsing* menggunakan API PHP.

```

<?php
    $db = new PDO('mysql:host=localhost;dbname=db_smartwalet;charset=utf8mb4',
    'root', '12');
    $stmt = $db->prepare("INSERT INTO `data_sensor`(`id`, `IR`, `TR`, `HU`, `timestamp`)
VALUES (NULL, :IR, :TR, :HU, CURRENT_TIMESTAMP);");
    $stmt->bindValue(':IR', $_GET['IR'], PDO::PARAM_INT);
    $stmt->bindValue(':TR', $_GET['TR'], PDO::PARAM_INT);
    $stmt->bindValue(':HU', $_GET['HU'], PDO::PARAM_INT);
    $stmt->execute();
?>

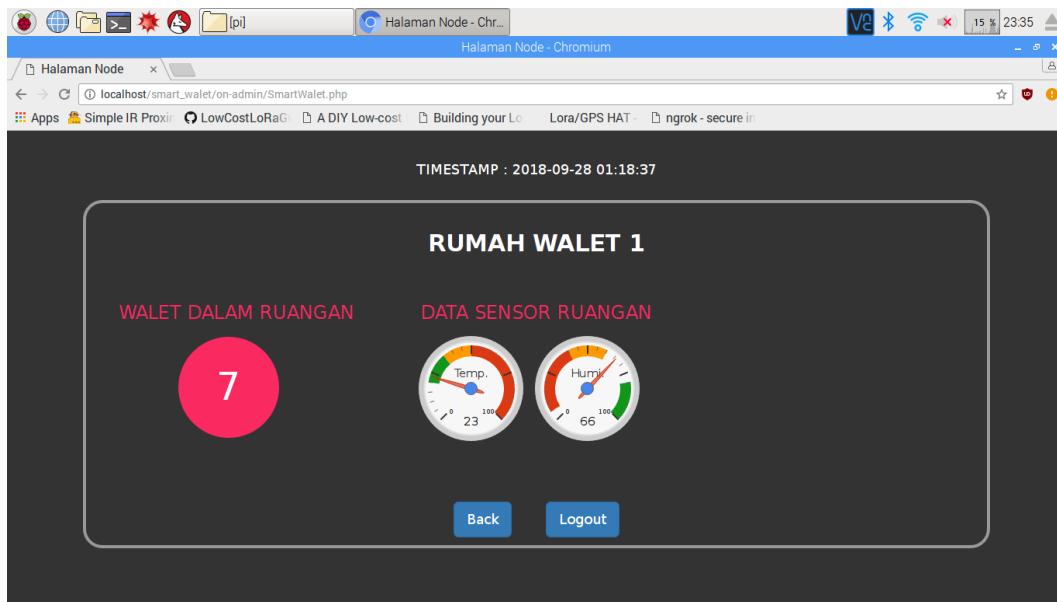
```

Data terakhir yang telah dikirim ke *database* akan ditampilkan pada *web* menggunakan Bahasa pemrograman PHP. Berikut data yang diterima *database*.

**Tabel 4.1** Database Data Sensor

<b>id</b>	<b>Populasi (IR)</b>	<b>Suhu / TR (°C)</b>	<b>Kelembapan / HU (%)</b>	<b>Time Stamp</b>
1	0	31	64	2018-09-27 15:39:22
2	14	31	64	2018-09-27 15:40:27
3	20	31	64	2018-09-27 15:41:30
4	5	31	64	2018-09-27 15:42:34
5	5	31	64	2018-09-27 15:43:39
6	0	24	67	2018-09-28 01:13:26
7	15	24	66	2018-09-28 01:14:29
8	20	24	66	2018-09-28 01:15:31
9	7	24	66	2018-09-28 01:16:33
10	7	23	66	2018-09-28 01:18:37

Pada tabel 4.1 merupakan *database* dari data sensor yang telah dikirim beserta *timestamp* ketika data tersimpan pada *database*. Data 1-5 dilakukan saat sore hari sedangkan data 6-10 dilakukan saat malam hari.



**Gambar 4.10** Visualisasi Data pada *Website*

Pada gambar 4.10 dapat dilihat data terakhir pada *database* ditampilkan pada halaman *web* yang akan di muat ulang setiap 20 detik. Data yang ditampilkan berupa data walet dalam ruangan, data suhu dan kelembapan dengan menggunakan *google chart*, serta *timestamp* ketika data tersimpan pada *database*. Berikut potongan kode program halaman *web* Data.

```

$sql = 'SELECT * FROM data_sensor ORDER BY id DESC LIMIT 1';
$query = mysqli_query($conn, $sql);
$row = mysqli_fetch_array($query);
$infra = $row['IR']; // Data Infrared (Populasi)
$temp = $row['TR']; // Data Temperature (suhu)
$shumi = $row['HU']; // Data Humidity (kelembapan)
$timestamp = $row['timestamp']; // Data Timestamp
$page = $_SERVER['PHP_SELF'];
$sec = "20"; ?> // Reload Page setiap 20 detik

        function drawChartTemp() {
            var data = google.visualization.arrayToDataTable([
                ['Label', 'Value'],
                ['Temp.'],
                <?php
                echo $temp;
                ?> ]);
            var options = {
                width: 650, height: 150,
                redFrom: 50, redTo: 100,
                yellowFrom: 35, yellowTo: 50,
                greenFrom: 20, greenTo:35,
                minorTicks: 5 };
            var chart = new
                google.visualization.Gauge(document.getElementById('chart_div_temp'));
            chart.draw(data, options); }

<h2 style="color:White; padding-bottom: 30px; text-align: center; font-weight: bold;">RUMAH
WALET 1</h2>
<div class="col-sm-4" style="padding-bottom: 20px;">

    <h3 style="color:#FA2A61; text-align: center; ">WALET DALAM RUANGAN</h3>
    <table class="columns">
        <tr><td><div class="col-sm-1" style="width:130px; height: 130px; border-radius:
        180px; background-color: #FA2A61; align: center; padding-top: 17px; margin-left: 80%;
        margin-top: 8px;">
        <?php if (!$query) { die ('SQL Error: ' . mysqli_error($conn)); }
        echo '<h1 style="color:White; text-align: center; font-size: 50px;">'.$infra.'</h1>';
        mysqli_free_result($query);
        mysqli_close($conn);?>
        </div></td></tr></table></div>
        <div class="col-sm-4" style="padding-bottom: 20px;">

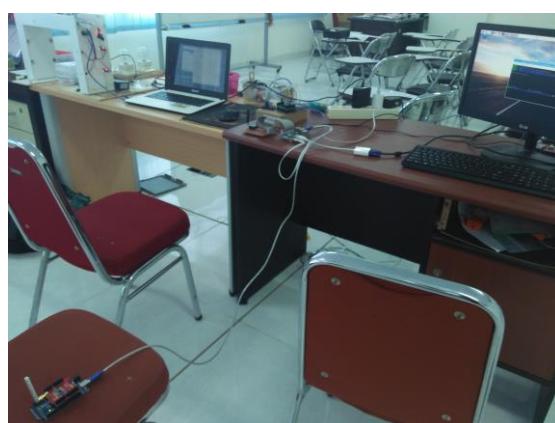
            <h3 style="color:#FA2A61;text-align: center;">DATA SENSOR RUANGAN</h3>
            <table class="columns">
                <tr><td><div id="chart_div_temp" style="width: 150px; height: 180px; margin-left:
                20px;"></div> </td>
                    <td><div id="chart_div_humi" style="width: 150px; height: 180px;"></div> </td>
                </tr></table></div>

```

### **IV.3 Pengambilan Data**

Pengambilan data dilakukan di dalam ruangan Laboratorium *Internet of Things & Parallel Computing*, Gedung Teknik Elektro, Fakultas Teknik Universitas Hasanuddin, Kab. Gowa dan di Perumahan Mutiara Indah Village, Samata, Kab. Gowa. Pemilihan lokasi penelitian ini berdasarkan pada kondisi suhu dan kelembapan yang berbeda serta kemudahan peneliti dalam menjangkau lokasi pengambilan data.

Pada penelitian ini peneliti melakukan dua kali pengambilan data, pengambilan data pertama merupakan pengambilan data sensor-sensor yang digunakan. Metode mengumpulkan data dengan cara manual yaitu meletakkan miniatur pintu burung walet. Lalu peneliti menggunakan alat bantu sebuah papan berukuran 10x6 cm berwarna hitam dengan ketebalan 0.5 cm, papan tersebut diumpamakan sebagai burung walet. Proses pengambilan data untuk menghitung populasi yaitu dengan cara melempar papan yang sesuai ukuran walet dewasa melewati miniatur pintu. Berikut proses pengambilan data sensor dapat dilihat pada gambar 4.11.



**Gambar 4.11** Proses Pengambilan Data Sensor

Untuk pengambilan data yang kedua dilakukan untuk mengukur *time on air*, serta *packet lost* pada sistem transmisi data. Proses pengambilan data transmisi dapat dilihat pada gambar 4.3 untuk *Node A*, gambar 4.4 untuk *Node B*, dan gambar 4.5 untuk *Gateway*. Metode yang digunakan dengan meletakkan masing-masing *node (end device)* dengan jarak  $\pm 125\text{m}$  dari *gateway*. Penempatan *nodes* dan *gateway* dapat dilihat pada gambar 4.12 berikut.



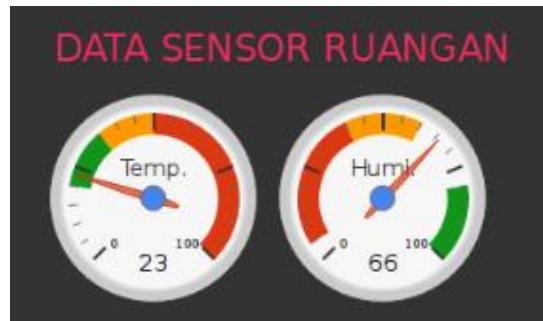
**Gambar 4.12** Penempatan *Node A*, *Node B* dan *Gateway*

#### **IV.4 Hasil Pengamatan dan Data**

Setelah dilakukan pengujian sistem, peniliti melakukan pengamatan pada sistem-sistem serta data yang dihasilkan.

##### **IV.4.1 Sistem Pembacaan Suhu dan Kelembapan**

Sistem pembacaan suhu dan kelembapan ruangan walet menghasilkan data yang dapat dilihat pada Tabel 4.1. data yang berlabel "TR" yaitu Suhu dan data berlabel "HU" yaitu Kelembapan.



**Gambar 4.13** Tampilan Data Suhu dan Kelembapan pada Halaman Web

Pada gambar 4.13 merupakan tampilan data suhu dan kelembapan pada halaman Web. Dimana data suhu dan kelembapan menggunakan fitur *google chart*.

#### IV.4.2 Sistem Penghitung Populasi Walet

Pada Tabel 4.1 dapat dilihat database populasi berlabel “IR” dimana data tersebut di dapatkan dari perhitungan antara keluar dan masuknya walet melalui miniatur *box* pintu masuk rumah walet. Pengamatan sistem penghitung populasi walet ini dilakukan saat pengambilan data. Dengan data tiap hasil pengujian dapat dilihat pada Tabel 4.2 berikut.

**Tabel 4.2** Hasil Pengujian Sistem Penghitung Populasi

Percobaan	Masuk	Keluar	Populasi Terdeteksi	Tidak Terdeteksi	Akurasi (%)
1	20	0	18	2	90
2	25	0	41	2	92
3	10	0	51	0	100
4	0	15	38	2	86,7
5	0	10	29	1	90
6	0	15	14	0	100

7	15	15	<b>13</b>	<b>1</b>	<b>96,7</b>
8	10	10	<b>12</b>	<b>1</b>	<b>95</b>
9	15	15	<b>11</b>	<b>1</b>	<b>96,7</b>
10	20	20	<b>11</b>	<b>0</b>	<b>100</b>
11	50	0	<b>58</b>	<b>3</b>	<b>94</b>
12	0	40	<b>20</b>	<b>2</b>	<b>95</b>

Pada pengujian sistem penghitung populasi dilakukan dengan 12 kali percobaan. Proses pengujian dilakukan di sore hari, dimana percobaan 1-3 dilakukan dengan melempar masuk miniatur walet ke arah miniatur pintu. Pada percobaan 4-6 dilakukan dengan melempar kearah luar. Serta pada percobaan 7-9 dilakukan dengan melempar Kearah masuk dan keluar dengan jumlah lemparan yang sama. Setiap percobaan pada percobaan 1-9 dilakukan pada interval pengiriman yaitu setiap 1 menit tiap percobaannya. Sedangkan pada percobaan 10-12 dilakukan diluar interval pengiriman atau menghiraukan proses pengiriman sesuai jumlah lemparan. Didapatkan akurasi pendektsian yang terendah yaitu 86,7%, sedangkan akurasi tertinggi mencapai 100%.

#### IV.4.3 Sistem Transmisi Data

Sistem transmisi data dengan menggunakan dua *node* dimana tiap *node* mengirim paket data ke *gateway* tiap +60 detik. Berikut hasil pengamatan pada sistem transmisi data antara *node* dan *gateway* dapat dilihat pada tabel 4.3.

**Tabel 4.3** Hasil Pengamatan Sistem Transmisi Data pada *Node A*

Pengiriman	Spreading Factor (SF)	Coding Rate (CR)	Bandwidth (kHz)	Real Payload Size (bytes)	Packet Lost	Time on Air (s)
1	12	4/5	125	35	x	2.86
2	12	4/5	125	35	x	3.17
3	12	4/5	125	35	x	2.81
4	12	4/5	125	35	x	3.06
5	12	4/5	125	35	x	3.01
6	12	4/5	125	35	x	2.8
7	12	4/5	125	35	x	2.99
8	12	4/5	125	35	x	2.66
9	12	4/5	125	35	x	2.95
10	12	4/5	125	35	x	2.75
11	12	4/5	125	35	x	2.88
12	12	4/5	125	35	x	3.04
13	12	4/5	125	35	x	2.93
14	12	4/5	125	35	x	2.6
15	12	4/5	125	35	x	2.68

Pada tabel 4.3, dapat dilihat hasil pengamatan sistem transmisi pada *node A* dengan 15 kali pengiriman data ke *gateway*. Dimana *node A* menggunakan SF = 12, CR = 4/5, dan BW = 125 kHz. Pada 15 kali pengiriman data, rata-rata *real payload size* yang dikirim adalah 35 bytes dan data yang dikirim lalu diterima oleh *gateway* tanpa adanya paket hilang. Sedangkan waktu saat diudara yang dibutuhkan saat pengiriman data dari *node A* ke *gateway* rata-rata 2,88 detik dengan *time on air* tercepat adalah saat pengiriman ke-14 yaitu 2,6 detik, serta *time on air* terlama adalah saat pengiriman ke-2 dengan waktu yang dibutuhkan adalah 3,17 detik.

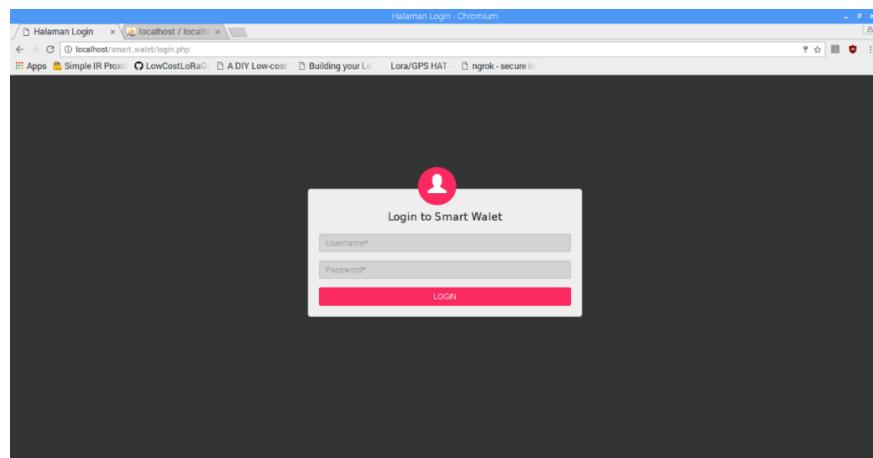
**Tabel 4.4** Hasil Pengamatan pada Sistem Transmisi Data pada *Node B*

Pengiriman	Spreading Factor [SF]	Coding Rate [CR]	Bandwidth [BW] (kHz)	Real Payload Size (bytes)	Packet Lost	Time on Air (s)
1	12	4/5	125	23	x	2.68
2	12	4/5	125	23	x	2.72
3	12	4/5	125	23	x	2.77
4	12	4/5	125	23	x	2.59
5	12	4/5	125	23	x	2.75
6	12	4/5	125	23	x	2.79
7	12	4/5	125	23	x	2.52
8	12	4/5	125	23	x	2.7
9	12	4/5	125	23	x	2.59
10	12	4/5	125	23	x	2.63
11	12	4/5	125	23	x	2.81
12	12	4/5	125	23	x	2.62
13	12	4/5	125	23	x	2.47
14	12	4/5	125	23	x	2.34
15	12	4/5	125	23	x	2.38

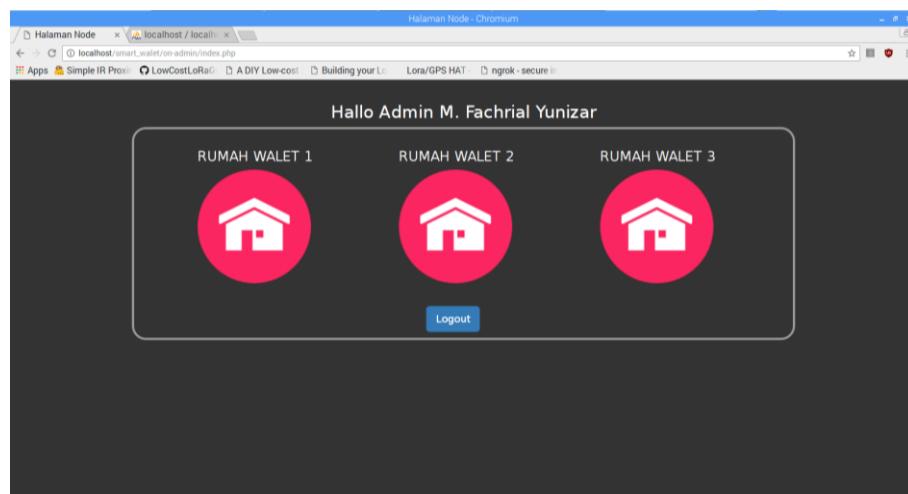
Pada tabel 4.4, dapat dilihat hasil pengamatan sistem transmisi pada *node B* dengan 10 kali pengiriman data ke *gateway*. Dimana *node B* juga menggunakan SF = 12, CR = 4/5, dan BW = 125 kHz. Pada 15 kali pengiriman data, yang berbeda dengan *node A* adalah rata-rata *real payload size* yang dikirim yaitu 23 bytes. Serta data yang dikirim *node B* lalu diterima oleh *gateway* tanpa adanya paket hilang. Sedangkan waktu saat diudara yang dibutuhkan saat pengiriman data dari *node B* ke *gateway* rata-rata 2,62 detik dengan *time on air* tercepat adalah saat pengiriman ke-14 yaitu 2,34 detik, serta *time on air* terlama adalah saat pengiriman ke-11 dengan waktu yang dibutuhkan adalah 2,81 detik.

#### IV.4.4 Web Server

Web server digunakan sebagai visualisasi dari implementasi *Internet of Things*. Dimana data suhu, kelembapan, populasi, dan *timestamp* dapat dipantau di komputer ataupun *Smartphone* dimana saja dan kapan saja oleh pengguna. Berikut tampilan halaman *Web*-nya dengan halaman awal merupakan halaman *login* yang dapat dilihat pada gambar 4.14.

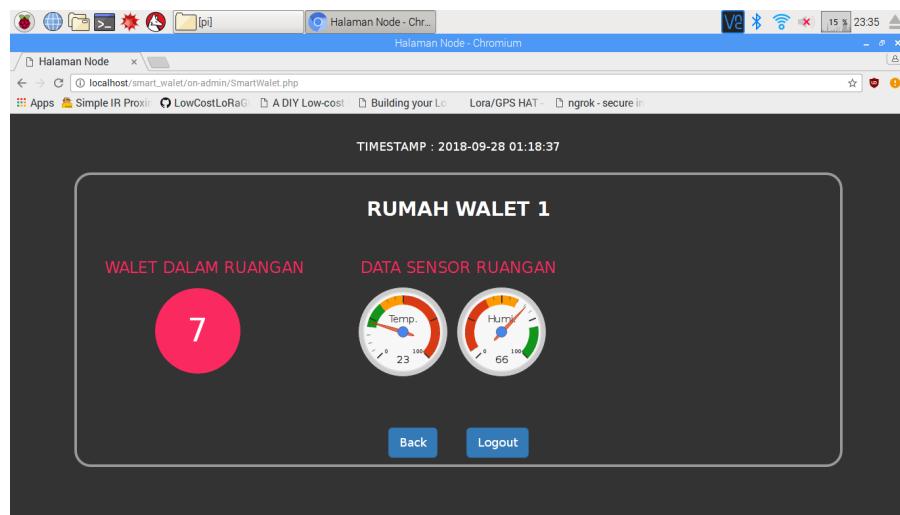


Gambar 4.14 Halaman *Login*



Gambar 4.14 Halaman Rumah Walet

Pada gambar 4.14 merupakan halaman pemilihan rumah walet yang terpasang *end devices*. Sistem *end devices* dapat dipasangkan pada lebih dari satu rumah sehingga tiap *end devices* yang terdapat pada rumah walet dapat dipantau dihalaman berbeda.



**Gambar 4.15** Tampilan Data Rumah Walet pada Halaman Web

Dapat dilihat pada gambar 4.15 merupakan tampilan data rumah walet. Yang terdiri dari data populasi walet dalam ruangan, data suhu dan kelembapan, serta *timestamp* data saat diterima pada *database*.

## **BAB V**

### **PENUTUP**

#### **V.1 Kesimpulan**

Berdasarkan penelitian yang telah dilakukan dalam pengujian sistem rumah walet cerdas berbasis *internet of things* , maka dapat disimpulkan bahwa:

1. Pada pengujian sistem penghitung populasi walet didapatkan rata-rata akurasi pendekripsi sebesar 95,1% pada 12 kali percobaan. Dengan akurasi pendekripsi yang terendah yaitu 86,7%, sedangkan akurasi tertinggi mencapai 100%. Rata-rata Akurasi pendekripsi tidak mencapai 100% dikarenakan spesifikasi pemancar infra merah dan penerima infra merah yang dipengaruhi dengan keadaan suhu panas disekitarnya.
2. Sistem rumah walet cerdas mengimplementasikan *internet of things* dalam memantau suhu, kelembapan dan populasi pada halaman *web* sehingga berhasil di pantau dengan jarak jauh. Hal tersebut dapat terealisasikan dikarenakan pada sistem ini menggunakan media transmisi LoRa untuk pengiriman data dari *end devices* ke *gateway* setiap ±60 detik menggunakan jaringan radio pada frekuensi 915MHz, lalu data tersebut dikirim ke *database web server* menggunakan *HTTP Request* pada bahasa pemrograman C++ dan proses terakhir adalah menampilkan data terakhir *database* pada halaman *Web* menggunakan

bahasa pemrograman PHP sehingga data-data dapat dipantau pada halaman *Web*.

3. Pada 15 kali pengujian pengiriman data dari *node* ke *gateway* membandingkan 2 sistem transmisi dengan parameter yang berbeda. Dimana kedua sistem transmisi data ini menggunakan *Spreading Factor* 12, *Coding Rate* 4/5, dan *Bandwidth* 125kHz yang sama tapi dengan *real payload size* / panjang data yang berbeda yaitu *Node A* = 35 bytes dan *Node B* = 23 bytes. Hasil dari 15 kali pengujian tersebut adalah rata-rata *time on air* pengiriman data dari *node A* ke *gateway* membutuhkan 2,88 detik tanpa adanya paket hilang. Sedangkan rata-rata *time on air* pengiriman data dari *node B* ke *gateway* membutuhkan 2,62 detik tanpa adanya paket hilang. Dari kedua pengujian sistem transmisi data tersebut dapat disimpulkan bahwa pengiriman data pada *node B* lebih cepat dari pada *node A* walaupun menggunakan *Spreading Factor*, *Coding Rate*, *Bandwidth*, dan jarak ke *gateway* yang sama. Hal itu dapat terjadi dikarenakan *real payload size* data dari *node B* lebih kecil dari pada *node A*.

## V.2 Saran

Sehubungan dengan selesainya proses pembuatan skripsi ini, penulis bermaksud menyampaikan beberapa saran kepada para pembaca yakni:

1. Peneliti berharap sistem pemantauan suhu dan kelembapan dapat dikembangkan lagi dengan menambahkan fitur pengaturan otomatis

suhu dan kelembapan rumah walet sehingga dapat membantu para pembudidaya walet dalam mengatur kondisi ruangan secara otomatis.

2. Pada proses pendektsian halangan yang disebabkan oleh walet, masih belum optimal dalam akurasi deteksi menggunakan penerima infra merah. Dikarenakan sensitivitas perangkat penerima infra merah yang dipengaruhi panas suhu disekitarnya.
3. Sistem yang telah dibuat peneliti belum bisa mengenali benda yang lewat, sehingga apapun yang melewati sensor akan terdeteksi sebagai walet. Dari hal tersebut peneliti berharap sistem yang telah di bangun dapat dikembangkan lagi agar akurasi pendektsian menjadi 100%.
4. Masih terdapat banyak variabel-variabel pada budidaya walet yang dapat di manipulasi. Dimana peneliti berharap kedepannya variabel-variabel tersebut dapat di manipulasi sehingga tercipta satu sistem budidaya walet menggunakan teknologi-teknologi modern.

## DAFTAR PUSTAKA

- Anonim1. (15. 8 2018). *Arduino Reference*. [Online]  
*Available At* Arduino.cc:  
<https://www.arduino.cc/reference/en/language/functions/externalinterrupts/attachinterrupt/>
- Anonim2. (27. 8 2018). *HTTP Request and Response*. [Online]  
*Available At* Bertzzie.com:  
<https://bertzzie.com/knowledge/serversidenodejs/RequestDanResponse.html>
- Atmoko, R. A. (2013). Sistem Monitoring dan Pengendalian Suhu dan Kelembaban Ruang pada Rumah Walet Berbasis Android, Web, SMS. *Seminar Nasional Teknologi Informasi & Komunikasi Terapan*, 283-290.
- Ayuti, T.;Garnida, D.;& Asmara, I. Y. (2016). Identifikasi Habitat dan Produksi Sarang Burung Walet (*Collocalia fuciphaga*) di Kabupaten Lampung Timur.
- Budiman, A. (2002). *Pedoman Membangun Gedung Walet*. Jakarta: AgroMedia.
- Cahyati, S. N. (2018). *Rancang Bangun Miniatur Stasiun Cuaca*. Makassar: Universitas Hasanuddin.
- Cytron Technologies. (2017). *Cytron RFM LoRa Shield*. Malaysia: Cytron Technologies Sdn. Bhd.
- Damayanti, F. (2017). *Rancang Bangun Web Server Sebagai Data Storage and Display dari Raspberry Pi*. Sriwijaya: Politeknik Negeri Sriwijaya.
- Ferly;Uzlianda, M.;& Rachmansyah. Rancang Bangun Alat Penghitung Jumlah Burung Walet yang Keluar Masuk Sarang. 1-10.
- Handika, O. D. (2017). *Miniatur Sistem Palang Pintu Kereta Api Otomatis Berbasis Arduino*. Yogyakarta: STMIK AKAKOM Yogyakarta.
- Naushad, A. (11. 8 2018). *How to Make a Raspberry Pi Web Server*. [Online]  
*Available At* Maker.pro:  
<https://maker.pro/raspberry-pi/projects/raspberry-pi-web-server>

- Nisa, A. (2018). *Pemanfaatan Teknologi Internet of Things Untuk Monitoring Konsentrasi CO dan CO2 Dalam Upaya Mendeteksi Kebakaran.* Makassar: Universitas Hasanuddin.
- Nurliana, S. (2016). *Rancang Bangun Alat Pemberi Isyarat Kecepatan Maksimum Melalui Sms Gateway Berbasis Mikrokontroler Pada Helm.* Palembang: Politeknik Negeri Sriwijaya.
- Patel, A. (15. 8 2018). *Using IR Sensor (TCRT 5000) With Arduino and Program to Remove Noise.* [Online] Available At Instructables.com:  
<https://www.instructables.com/id/Using-IR-Sensor-TCRT-5000-With-Arduino-and-Program>
- Wicaksono, S. N. (2017). *Aplikasi Kran Otomatis Berbasis Arduino.* Yogyakarta: AKAKOM Yogyakarta.
- Wpnsmith. (25. 7 2018). *Raspberry Pi Temperature & Humidity Network Monitor : 11 Steps.* [Online] Available At Instructables.com:  
<https://www.instructables.com/id/Raspberry-Pi-Temperature-Humidity-Network-Monitor/>
- Wulansari, U. (2014). *Prototipe Penghitung Jumlah Burung Walet Yang Keluar Masuk Sarang Dengan Komunikasi Data Wireless Berbasis Mikrokontroler.* Palembang: Politeknik Negeri Sriwijaya.

# **LAMPIRAN**

### 1. Psudo Code Program Node\_RumahWaletCerdas.ino

```
#define LORAMODE 1
uint8_t node_addr=8;
unsigned int idlePeriod = 1000;
unsigned int nCycle = idlePeriod*60/LOW_POWER_PERIOD;
sx1272config my_sx1272config;

dht DHT;
#define DHT11_PIN 7
int act_cooling = 8;
int act_heating = 9;
const byte interruptPin2 = 2;
const byte interruptPin3 = 3;
unsigned long timeSensorDepan=0;
unsigned long timeSensorBelakang=0;
unsigned long previousMillis1 = 0;
int selisih_time=0;
int dataA=0,dataB=0;
int terdeteksi;
const long interval1 = 101;
void setup()
{
    pinMode(interruptPin2, INPUT);
    attachInterrupt(digitalPinToInterrupt(interruptPin2), sensorDepan, RISING);
    pinMode(interruptPin3, INPUT);
    attachInterrupt(digitalPinToInterrupt(interruptPin3), sensorBelakang, RISING);
    pinMode(act_cooling, OUTPUT);
    pinMode(act_heating, OUTPUT);
    Serial.begin (38400);
    digitalWrite(8, HIGH);
    int e;
#define LOW_POWER
    bool low_power_status = IS_LOWPOWER;
    // Power ON the module
    sx1272.ON();

    void loop(void)
    {
        unsigned long currentMillis1 = millis();
        if (currentMillis1 - previousMillis1 >= interval1){
            previousMillis1 = currentMillis1;
            if (dataA==1 && dataB==1){
                if (timeSensorDepan > timeSensorBelakang){
```

```
selisih_time=timeSensorDepan-timeSensorBelakang;
if (selisih_time<=100){
    terdeteksi=terdeteksi+1;
    timeSensorDepan = 0;
    timeSensorBelakang = 0;
    selisih_time = 0;
    dataA = 0;
    dataB = 0;
} else {
    timeSensorDepan = 0;
    timeSensorBelakang = 0;
    selisih_time = 0;
    dataA = 0;
    dataB = 0;
}
}

if (timeSensorDepan < timeSensorBelakang){
    selisih_time=timeSensorBelakang-timeSensorDepan;
    if (selisih_time<=100){
        if (terdeteksi > 0){
            terdeteksi=terdeteksi-1;
        }else{
            terdeteksi=0;
        }
        timeSensorDepan = 0;
        timeSensorBelakang = 0;
        selisih_time = 0;
        dataA = 0;
        dataB = 0;
    } else {
        timeSensorDepan = 0;
        timeSensorBelakang = 0;
        selisih_time = 0;
        dataA = 0;
        dataB = 0;
    }
}

}if (timeSensorDepan == timeSensorBelakang) {
    timeSensorDepan = 0;
    timeSensorBelakang = 0;
    selisih_time = 0;
    dataA = 0;
    dataB = 0;
}
}
```

```

}

long startSend;
long endSend;
uint8_t app_key_offset=0;
int e;
#ifndef LOW_POWER
// 600000+random(15,60)*1000
if (millis() > nextTransmissionTime) {
#endif
#ifndef WITH_APPKEY
    app_key_offset = sizeof(my_appKey);
    // set the app key in the payload
    memcpy(message,my_appKey,app_key_offset);
#endif
    int temp;
    uint8_t r_size;
    int chk = DHT.read11(DHT11_PIN);
    int dht_temp = DHT.temperature;
    int dht_humi = DHT.humidity;
    char final_str[80] = "\\!";
    char aux[6] = "";
    char dataWalet[5];
    char dataTemp[5];
    char dataHumi[5];
    String str1;
    String str2;
    String str3;
    str1 = String(terdeteksi);
    str1.toCharArray(dataWalet,5);
    str2 = String(dht_temp);
    str2.toCharArray(dataTemp,5);
    str3 = String(dht_humi);
    str3.toCharArray(dataHumi,5);
    for (int i=0; i<number_of_sensors; i++) {
        if (sensor_ptrs[i]->get_is_connected() || sensor_ptrs[i]->has_fake_data()) {
            ftoa(aux, sensor_ptrs[i]->get_value(), 0);
            if (i==0) {
                sprintf(final_str, "/new_record.php?IR=%s&%s=%s",dataWalet,sensor_ptrs[i]-
>get_nomenclature(),dataTemp);
            }
            else {
                sprintf(final_str, "%s&%s=%s//", final_str, sensor_ptrs[i]->get_nomenclature(),dataHumi);
            }
        }
    }
}

```

```

}

r_size=sprintf((char*)message+app_key_offset, final_str);

PRINT_CSTSTR("%s","Reading ");
PRINTLN;
delay(100);
PRINT_CSTSTR("%s","Sending ");
PRINT_STR("%s",(char*)(message+app_key_offset));
PRINTLN;
PRINT_CSTSTR("%s","Real payload size is ");
PRINT_VALUE("%d", r_size);
PRINT_CSTSTR("%s","LoRa pkt seq ");
PRINT_VALUE("%d", sx1272.packet_sent.packnum);
PRINTLN;
PRINT_CSTSTR("%s","LoRa Sent in ");
PRINT_VALUE("%ld", endSend-startSend);
PRINTLN;
PRINT_CSTSTR("%s","LoRa Sent w/CAD in ");
PRINT_VALUE("%ld", endSend-sx1272._startDoCad);
PRINTLN;
PRINT_CSTSTR("%s","Packet sent, state ");
PRINT_VALUE("%d", e);
PRINTLN;

nCycle = idlePeriodInMin*60/LOW_POWER_PERIOD + random(2,4);
PRINT_VALUE("%ld", nextTransmissionTime);
PRINTLN;
PRINT_CSTSTR("%s","Will send next value at\n");
// use a random part also to avoid collision
nextTransmissionTime=millis()+(unsigned long)idlePeriodInMin*60+(unsigned long)random(15,60);
PRINT_VALUE("%ld", nextTransmissionTime);
PRINTLN;
}

#endif
}

void sensorDepan() {
    timeSensorDepan = millis();
    dataA=1;
}

void sensorBelakang() {
    timeSensorBelakang = millis();
    dataB=1;
}

```

## 2. Psudo Code Program Lora\_Gateway.cpp

```
void OnBegin( const happyhttp::Response* r, void* userdata )
{
    printf( "BEGIN (%d %s)\n", r->getstatus(), r->getreason() );
    count = 0;
}

void OnData( const happyhttp::Response* r, void* userdata, const unsigned char* data, int n )
{
    fwrite( data,1,n, stdout );
    count += n;
}

void OnComplete( const happyhttp::Response* r, void* userdata )
{
    printf( "COMPLETE (%d bytes)\n", count );
}

//#include "rapidjson/httpplib.h"
#include <iostream>
#endif
#ifndef ARDUINO
#include <SPI.h>
#define PRINTLN      Serial.println("")
#define PRINT_CSTSTR(fmt,param)  Serial.print(F(param))
#define PRINT_STR(fmt,param)   Serial.print(param)
#define PRINT_VALUE(fmt,param)  Serial.print(param)
#define PRINT_HEX(fmt,param)   Serial.print(param,HEX)
#define FLUSHOUTPUT     Serial.flush();
#else
#define PRINTLN      printf("\n")
#define PRINT_CSTSTR(fmt,param) printf(fmt,param)
#define PRINT_STR(fmt,param)   PRINT_CSTSTR(fmt,param)
#define PRINT_VALUE(fmt,param) PRINT_CSTSTR(fmt,param)
#define PRINT_HEX(fmt,param)   PRINT_VALUE(fmt,param)
#define FLUSHOUTPUT     fflush(stdout);
#endif
#ifndef DEBUG
#define DEBUGLN      PRINTLN
#define DEBUG_CSTSTR(fmt,param) PRINT_CSTSTR(fmt,param)
#define DEBUG_STR(fmt,param)   PRINT_CSTSTR(fmt,param)
#define DEBUG_VALUE(fmt,param) PRINT_VALUE(fmt,param)
#else
#define DEBUGLN
#define DEBUG_CSTSTR(fmt,param)
#endif
```

```

#define DEBUG_STR(fmt,param)
#define DEBUG_VALUE(fmt,param)

#elif defined BAND900
#define ENDING_CHANNEL 12
uint8_t loraChannelIndex=12;

loraChannelArray[MAX_NB_CHANNEL]={CH_00_900,CH_01_900,CH_02_900,CH_03_900,CH_04_900,CH_05_900,CH_06_900,CH_07_900,CH_08_900,CH_09_900,CH_10_900,CH_11_900,CH_12_900};

// the special mode to test BW=125MHz, CR=4/5, SF=12
#define LORA_ADDR 1
// to unlock remote configuration feature
#define UNLOCK_PIN 1234

// has customized LoRa settings
e = sx1272.setSF(optSF);
PRINT_CSTSTR("%s","^$LoRa SF ");
PRINT_VALUE("%d", optSF);
PRINT_CSTSTR("%s",": state ");
PRINT_VALUE("%d", e);
PRINTLN;

e = sx1272.setBW( (optBW==125)?BW_125:(optBW==250)?BW_250:BW_500 );
PRINT_CSTSTR("%s","^$LoRa BW ");
PRINT_VALUE("%d", optBW);
PRINT_CSTSTR("%s",": state ");
PRINT_VALUE("%d", e);
PRINTLN;

// indicate that we have a custom setting
loraMode=0;
void setup()
{
    int e;
    e = sx1272.ON();
    PRINT_CSTSTR("%s","^$*****Power ON: state ");
    PRINT_VALUE("%d", e);
    PRINTLN;
    e = sx1272.getSyncWord();
    char my_data[100];
    for (; a<tmp_length; a++,b++) {
        PRINT_STR("%c",(char)sx1272.packet_received.data[a]);
        my_data[a]=(char)sx1272.packet_received.data[a];
        if (b<MAX_CMD_LENGTH)
            cmd[b]=(char)sx1272.packet_received.data[a];
    }
    // strlen(cmd) will be correct as only the payload is copied
    cmd[b]='\0';
    PRINTLN;
}

```

```

my_data[a]='\0';

const char * str = my_data;

for (int d=3; d<38; ++d){
    my_data[d]=my_data[d+1];
    for (int e=2; e<37; ++e){
        my_data[e]=my_data[e+1];
        for (int f=1; f<36; ++f){
            my_data[f]=my_data[f+1];
            for (int g=0; g<35; ++g){
                my_data[g]=my_data[g+1];
            }
        }
    }
}

char y[100];
y[0]='\0';
strncpy(y,my_data,35);
PRINT_STR("%s",y);
PRINTLN;

happyhttp::Connection conn( "localhost",80);
conn.setcallbacks( OnBegin, OnData, OnComplete, 0 );
conn.request( "GET", y);
while (conn.outstanding())
    conn.pump();

PRINTLN;
FLUSHOUTPUT;

#if not defined ARDUINO && defined WININPUT
if (keyIndex) {
    PRINT_CSTSTR("%s","keyboard input : ");
    PRINT_STR("%s",keyPressBuff);
    PRINTLN;
}
#endif
#endif
}

if (receivedFromSerial || receivedFromLoRa) {
    i=0;
    if (cmd[i]=='/' && cmd[i+1]=='@') {
        PRINT_CSTSTR("%s","^$Parsing command\n");
        i=2;
        PRINT_CSTSTR("%s", "$");
        PRINT_STR("%s",cmd);
        PRINTLN;
    }
}

```

**3. Kode Program New\_record.php**

```
<?php
    $db = new PDO('mysql:host=localhost;dbname=db_smartwalet;charset=utf8mb4', 'root', '12');
    $stmt = $db->prepare("INSERT INTO `data_sensor` (`id`, `IR`, `TR`, `HU`, `timestamp`) VALUES
        (NULL, :IR, :TR, :HU, CURRENT_TIMESTAMP);");
    $stmt->bindValue(':IR', $_GET['IR'], PDO::PARAM_INT);
    $stmt->bindValue(':TR', $_GET['TR'], PDO::PARAM_INT);
    $stmt->bindValue(':HU', $_GET['HU'], PDO::PARAM_INT);
    $stmt->execute();
?>
```

**4. Kode Program SmartWalet.php**

```
<?php
$db_host = 'localhost'; // Nama Server
$db_user = 'root'; // User Server
$db_pass = '12'; // Password Server
$db_name = 'db_smartwalet'; // Nama Database
$conn = mysqli_connect($db_host, $db_user, $db_pass, $db_name);
if (!$conn) {
    die ('Gagal terhubung MySQL: ' . mysqli_connect_error());
}
$sql = 'SELECT * FROM data_sensor ORDER BY id DESC LIMIT 1';
$query = mysqli_query($conn, $sql);
$row = mysqli_fetch_array($query);
$infra = $row['IR'];
$temp = $row['TR'];
$shumi = $row['HU'];
$timestamp = $row['timestamp'];
$page = $_SERVER['PHP_SELF'];
$sec = "60";
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="refresh" content="<?php echo $sec?>; URL='<?php echo $page?>'>">
<title>Halaman Node</title>
<link href="../assets/css/bootstrap.min.css" rel="stylesheet">
<link href="../assets/css/style.css" rel="stylesheet">
```

```
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
  google.charts.load('current', {'packages':['gauge']});
  google.charts.setOnLoadCallback(drawChartTemp);
  google.charts.setOnLoadCallback(drawChartHumi);
  function drawChartTemp() {
    var data = google.visualization.arrayToDataTable([
      ['Label', 'Value'],
      ['Temp.',
        <?php
        echo $temp;
        ?>]
    ]);
    var options = {
      width: 650, height: 150,
      redFrom: 50, redTo: 100,
      yellowFrom: 35, yellowTo: 50,
      greenFrom: 20, greenTo:35,
      minorTicks: 5
    };
    var chart = new google.visualization.Gauge(document.getElementById('chart_div_temp'));
    chart.draw(data, options);
  }
  function drawChartHumi() {
    var data = google.visualization.arrayToDataTable([
      ['Label', 'Value'],
      ['Humi.',
        <?php
        echo $humi;
        ?>]
    ]);
    var options = {
      width: 650, height: 150,
      greenFrom: 80, greenTo: 100,
      yellowFrom: 40, yellowTo: 60,
      redFrom: 5, redTo: 40,
      minorTicks: 5 };
    var chart = new google.visualization.Gauge(document.getElementById('chart_div_humi'));
    chart.draw(data, options);
  }
</script>
</head>
<body>
<?php
```

```

session_start();
/**
 * Jika Tidak login atau sudah login tapi bukan sebagai admin
 * maka akan dibawa kembali ke halaman login atau menuju halaman yang seharusnya.
 */
if ( !isset($_SESSION['user_login']) || 
    ( isset($_SESSION['user_login']) && $_SESSION['user_login'] != 'admin' ) ) {
    header('location:../../login.php');
    exit();?>
<h4 style="color:White; text-align: center; margin-top: 40px;"><?php
if (!$query) {
    die ('SQL Error: ' . mysqli_error($conn));
}
echo 'TIMESTAMP : '.$timestamp. '</br/>';
mysqli_free_result($query);
mysqli_close($conn);
?></h4><

```

```

        </table>
    </div>
    <div class="col-sm-4" style="padding-bottom: 20px;">
        <h3 style="color:#FA2A61;text-align: center;">ACTUATOR</h3>
        <table class="columns">
            <tr>
                <td><h4 style="color:White;padding-left: 105px;">
                    <?php
                    if ($temp < 26 && $humi > 90) {
                        $act_cooler = '';
                        $act_heater = '';
                    }else if ($temp > 29 && $humi < 80) {
                        $act_cooler = '';
                        $act_heater = '';
                    } else {
                        $act_cooler = '';
                        $act_heater = '';
                    }
                    echo 'Cooler : '.$act_cooler. '<br/><br/>';
                    echo 'Heater : '.$act_heater. '<br/>';
                ?></h4></td>
                <td> </td>
            </tr>
        </table>
    </div>
    </div>
    <div style="padding-top: 20px;padding-bottom: 10px;text-align: center;">
        <button type="button" class="btn btn-primary btn-lg"> <a href="index.php" style="color:White;"> Back
        </a></button> .....
        <button type="button" class="btn btn-primary btn-lg"> <a href=".../logout.php"
        style="color:White;">Logout</a></button>
    </div>
    </div>
    <script src=".../assets/js/jquery.min.js"></script>
    <script src=".../assets/js/bootstrap.min.js"></script>
</body>
</html>

```