

## **TUGAS AKHIR**

# **RANCANG BANGUN SISTEM PENGUKURAN DAN PEMANTAUAN SUHU UDARA SECARA *REAL TIME* BERBASIS IoT MENGGUNAKAN KAA PLATFORM**

*Sebagai salah satu syarat menyelesaikan  
Program Studi Diploma III Jurusan Teknik Informatika*



**Oleh :**

**AL MUTAQHIN GUSTIAN ARHEF**

**6103151005**

**PROGRAM STUDI D-III TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BENGKALIS**

**BENGKALIS**

**JANUARI, 2018**

## DAFTAR ISI

	<b>Halaman</b>
<b>HALAMAN SAMPUL</b>	
<b>HALAMAN JUDUL</b>	i
<b>HALAMAN PENGESAHAN</b>	ii
<b>ABSTRAK</b>	iii
<b>DAFTAR ISI</b>	iv
<b>DAFTAR TABEL</b>	v
<b>DAFTAR GAMBAR</b>	vi
<b>BAB I     PENDAHULUAN</b>	
1.1 Latar Belakang	1
1.2 Batasan Masalah	2
1.3 Tujuan	2
1.4 Manfaat	3
1.5 Metode Penyelesaian Masalah	3
<b>BAB II    TINJAUAN PUSTAKA</b>	
2.1 Kajian Terdahulu	7
2.2 Landasan Teori	9
<b>BAB III   METODE PENELITIAN</b>	
3.1 Data dan Alat Penelitian	14
3.2 Perancangan Sistem	16

<b>BAB IV</b>	<b>JADWAL PENELITIAN</b>	
4.1	Hasil	28
4.2	Pengujian	28
<b>BAB V</b>	<b>PENUTUP</b>	
5.1	Kesimpulan	54
5.2	Saran	54
<b>DAFTAR PUSTAKA</b>		56

## DAFTAR TABEL

	<b>Halaman</b>
Tabel 3.1 Perancangan <i>Database</i>	23
Tabel 4.1 Nilai Suhu Pada Lobi Menggunakan Sensor Suhu	44

## DAFTAR GAMBAR

	Halaman
Gambar 1.1 Prosedur Penelitian	4
Gambar 2.1 <i>Java</i>	11
Gambar 2.2 <i>Python</i>	11
Gambar 2.3 <i>Twitterbots</i>	12
Gambar 3.1 Gambaran Perancangan Sistem	16
Gambar 3.2 <i>Flowchart</i> dari Sistem yang Dibuat	18
Gambar 3.3 <i>Flowchart</i> Sistem Sensor	19
Gambar 3.4 Diagram Kasus Penggunaan Sistem	20
Gambar 3.5 Perancangan Pada Sensor	21
Gambar 3.6 Perancangan <i>Gateway</i>	21
Gambar 3.7 Perancangan <i>Database</i>	22
Gambar 3.8 Perancangan Tampilan <i>Dashboard</i>	25
Gambar 3.9 Perancangan Tampilan Suhu Udara dan Baterai	26
Gambar 3.10 Perancangan Tampilan Suhu Udara dan Baterai <i>real-time</i>	27
Gambar 4.1 Persiapan Sensor	29
Gambar 4.2 Persiapan Mini PC	30
Gambar 4.3 <i>Coding</i> Sensor Suhu Udara	32
Gambar 4.4 <i>Instalasi Library</i> Pembacaan <i>Port</i>	33
Gambar 4.5 <i>Source Code</i> Baca <i>Port</i> dan Koneksi Mikrokontroler	35
Gambar 4.6 <i>Source Code</i> <i>Parsing</i> data	36
Gambar 4.7 <i>Source Code</i> Pengiriman Data ke <i>Database</i>	39
Gambar 4.8 <i>Source Code</i> Jeda Waktu dan <i>Looping</i> Pengiriman Data	39
Gambar 4.9 <i>Source Code</i> Koneksi ke <i>Database</i>	39
Gambar 4.10 <i>Source Code</i> <i>Input</i> Data Sensor dari Mini PC ke <i>Database</i>	41
Gambar 4.11 Koneksi Dari Sensor ke Mikrokontroler	42

Gambar 4.12 Tampilan Nilai Suhu Pada Serial Waspmote PRO	42
Gambar 4.13 Pemantau Suhu Pada Lobi Menggunakan Sensor Suhu	43
Gambar 4.14 Grafik Pemeriksaan Suhu Udara di GU dan GC	45
Gambar 4.15 Tampilan <i>Dasboard</i>	46
Gambar 4.16 Tampilan Grafik dan Rekap	47
Gambar 4.17 Tampilan Grafik <i>Real-Time</i>	47
Gambar 4.18 sudo apt-get install Python-setuptools	48
Gambar 4.19 sudo easy_install pip	48
Gambar 4.20 Sudo pip install twython	49
Gambar 4.21 sudo apt-get install Python-setuptools	49
Gambar 4.22 Source Code MyTweeter.py	50
Gambar 4.23 Source Code MyTweeter.py Pada Java	51
Gambar 4.24 Status Di Twitter Sesuai Dengan Waktu Di Database	52

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dengan adanya suhu atau temperatur udara maka suatu tempat dapat dibedakan, antara tempat yang satu dengan tempat yang lain. Dimana semakin tinggi suhu maka semakin panas tempat tersebut dan begitu sebaliknya, semakin rendah suhu maka semakin dingin tempat tersebut.

Pada tempat-tempat tertentu pasti memiliki tempat yang sangat penting seperti lobi, karena di lobi semua orang melakukan aktifitas termasuk juga pengunjung, oleh karena itu suhu udara lobi harus di perhatikan juga, bahkan perlu di-*monitoring* supaya kenyamanan pengunjung dapat terjaga, sehingga pengunjung betah dan nyaman ketika melakukan aktifitas selama di lobi.

Pada umumnya melakukan monoting dengan menggunakan alat pengukur temperatur biasa, dan melakukan pemantauan secara manual bisa di lakukan tapi itu masi kurang efektif dalam mengatasi masalah tersebut. Apalagi pada era teknologi sekarang setiap urusan diharapkan dapat dilakukan dengan cepat dan dengan hasil yang maksimal.

Oleh karena itu, penulisan mengusulkan sistem pengukuran dan pemantauan suhu udara secara *real time* dimana data suhu yang didapat dari sensor suhu udara dikirim menggunakan sistem jaringan *nirkabel*. Dimana pada sistem jaringan *nirkabel* menggunakan alat yang bernama *Xbee Pro*. Karena *Xbee Pro* memiliki kelebihan yaitu daya yang rendah dalam proses penggunaannya serta memiliki jangkauan yang luas. Data yang dikirim ke *raspberry pi* nantinya akan dikirim lagi ke *database kaa platform*.

Disini menggunakan *kaa platform* karena memiliki beberapa keunggulan dari pada MQTT (*Message Queuing Telemetry Transport*) adapun keunggulan *kaa platform* sebagai beriku:

kaa adalah *multi platform middleware* yang merupakan produk *open source* yang berguna dalam pengembangan IoT (*Internet of Things*) dan memiliki banyak keunggulan dimana programmer bebas dalam hal melakukan suatu pengembangan, dan didukung sangat banyak *platform* seperti *consumer elektronik, agriculuter, healteare*, maupun *weareble gadget*.

Berdasarkan dari latar belakang permasalahan yang telah dijelaskan, maka perlu direncanakan suatu penelitian untuk membuat sistem yang mampu menangani permasalahan tersebut. Oleh sebab itu, perancangan penelitian tersebut dituangkan dalam judul, **“RANCANG BANGUN SISTEM PENGUKURAN DAN PEMANTAUAN SUHU UDARA SECARA *REAL TIME* BERBASIS IoT MENGGUNAKAN KAA PLATFORM”**

## **1.2 Batasan Masalah**

Berdasarkan rumusan masalah diatas, adapun yang menjadi batasan masalah pada penelitian adalah :

1. Hanya melakukan pengukuran pada suhu udara.
2. Menggunakan *cloud kaa platform*
3. Hasil pengukuran hanya ditampilkan pada sebuah halaman *website* dan file keluaran berupa format CSV.
4. Notifikasi peringatan dini hanya ditampilkan pada media sosial.

## **1.3 Tujuan**

Sesuai dengan rumusan dan batasan masalah diatas, maka tujuan yang ingin dicapai dari penelitian adalah untuk membangun sistem pengukur dan memantau suhu udara secara *real time*, dimana data tersebut dapat ditampilkan dalam halaman *website* dan bisa menampilkan notifikasi melalui media sosial dan file keluaran berupa CSV yang berguna sebagai rekap nantinya.



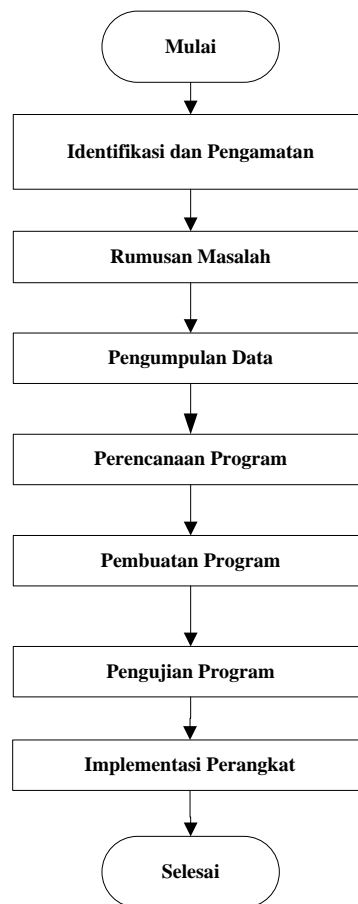
#### **1.4 Manfaat**

Adapun beberapa manfaat yang diharapkan pada pembuatan proposal akhir ini adalah :

1. Memudahkan dalam melakukan pemantauan suhu udara pada suatu tempat secara *real time*.
2. Dalam pembuatannya menggunakan *cloud* kaa *platform* sebagai media penyimpanan.
3. Memudahkan pengguna dalam membuat rekap mengenai data suhu udara, karena pada website kita bisa langsung mengunduh file rekap dengan format CSV.
4. Pengguna dapat mengetahui apa bila ada perubahan suhu diatas rata-rata, karena apa bila suhu melewati dari batas yang telah di tentukan, maka akan mengirim notifikasi ke media sosial secara otomatis.

#### **1.5 Metode Penyelesaian Masalah**

Metode penyelesaian masalah diperlukan agar penelitian yang dilakukan dapat terlaksana dengan baik dan teratur. Prosedur penelitian dimulai dari langkah-langkah menggunakan suatu alat yang berguna untuk mengumpulkan data dan menjawab pertanyaan – pertanyaan dalam suatu penelitian. Di dalam prosedur penelitian ini, penulis membahas tentang metode dan teknik pengumpulan data, sample penelitian, penyusunan alat, langkah - langkah pengumpulan data dan prosedur pengolahan data.



**Gambar 1.1 Prosedur Penelitian**  
(sumber: Data Olahan)

### **1.5.1 Identifikasi dan Pengamatan**

Melakukan identifikasi terhadap masalah yang ada dimana ketika akan melakukan pemantauan suhu udara disuatu tempat, seseorang tidak bisa hanya mengandalkan indra praba ataupun alat termometer biasa saja, karena hasil dan waktu yang dihabiskan tidak sangat efisien. Jadi akan berdampak dalam jangka pendek maupun jangka panjang nantinya.

### **1.5.2 Rumusan Masalah**

Dari permasalahan yang ada nantinya akan diambil inti dari suatu permasalahan didalam masalah tersebut, mengenai apa – apa yang terjadi sehingga dapat menemukan suatu kesimpulan yang tepat.

### **1.5.3 Pengumpulan Data**

Dalam prose pengumpulan data menggunakan metode pengumpulan studi pustaka, dimana metode pengumpulan studi pustaka adalah dengan mempelajari, dan melakukan pencarian maupun mengumpulkan data yang berhubungan dengan penelitian yang akan dibuat nantinya, contoh: buku, jurnal dan akses internet.

### **1.5.4 Perancangan Program**

Perancangan program yang akan dibuat berupa sensor suhu udara yang diletakkan pada suatu tempat tertentu, dimana pada tempat tersebut nantinya akan dilakukan pemantauan secara *real time*, dimana datanya akan dikirim secara terus menerus ke mini PC, dan di mini PC tersebut data diproses sebelum nanti dilakukan proses sinkronisasi ke *cloud*, dan kemudian data tersebut akan ditampilkan dalam suatu halaman *website*.

### **1.5.5 Pembuatan Program**

Dimana nantinya akan dilakukan desain terhadap program yang telah dirancang sebelumnya, dan memasukkan beberapa pengkodean yang diperlukan dengan bahasa pemrograman yang telah ditentukan sebelumnya diperancangan, dan menambah beberapa hal yang diperlukan.

### **1.5.6 Pengujian Program**

Akan dilakukan pengujian terhadap program yang telah dibuat sebelumnya, yang bertujuan untuk melihat kualitas dari program yang telah dibuat dan melakukan beberapa penambahan yang bertujuan untuk pengembangan terhadap program yang telah dibuat dan memastikan bahwa program yang telah dibuat dapat berjalan dengan baik.

### **1.5.7 Hasil Program**

Hasil akhir dari program yang akan dibuat adalah berupa sensor suhu udara yang diletakkan pada tempat tertentu, yang bertujuan untuk melakukan pemantauan *real time* pada tempat tersebut dan kemudian data tersebut dikirim pada suatu alat untuk diproses yang kemudia disinkronkan pada *cloud*, dan dapat dilihat pada halaman *website*.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Kajian Terdahulu**

Menurut Indriani (2014) “Sistem pengendalian suhu digunakan di banyak fasilitas seperti kontrol suhu ruangan, kontrol kualitas makanan dan minuman, kontrol peralatan elektronik dan lain-lain. Pengontrolan suhu dapat dilakukan dengan menggunakan mikrokontroler sebagai media pengolahan dan proses perubahan suhu yang terjadi pada tegangan. Salah satu komponen pendukung pada mikrokontroler untuk mendeteksi perubahan suhu adalah sensor suhu LM 35 yang dalam penelitian ini kita akan fokus pada sensor suhu LM 35 untuk mendeteksi perubahan suhu pada air laut dalam skala kecil. Tujuan penelitian ini adalah untuk mengetahui kemampuan dan kinerja sensor temperatur LM 35 untuk mendeteksi perubahan suhu air laut dan memeriksa kondisi sensor temperatur LM 35 yang digunakan pada air laut. Hasil penelitian menunjukkan bahwa sensor suhu LM 35 dapat digunakan untuk mengukur perubahan suhu air laut tanpa memiliki perbedaan yang lebih tinggi pada nilai aktual dan hasil eksperimen. kisaran perbedaan lebih kecil dan sensor suhu LM 35 dapat digunakan untuk mengukur perubahan suhu air laut”.

Menurut Nasrullah (2015) “Penelitian ini bertujuan untuk mengukur dan menganalisis kondisi temperatur dan kelembaban relatif udara diluar bangunan berdasarkan data klimatik daerah tropis lembab. Data yang digunakan untuk menganalisis adalah temperatur dan kelembaban relatif udara yang diperoleh dari alat monitor cuaca ( *weather monitoring* ) Vaisala, RTU ( *Remote Terminal Unit* ), yang merupakan bagian dari perangkat AWS ( *Automatic Weather Stations* ) pada stasiun data kampus baru Fakultas Teknik Unhas Gowa. Data diolah dan dianalisis melalui komputer dengan menggunakan program Microsoft Excel dalam bentuk tabel dan diagram. Hasil penelitian menunjukkan kondisi temperatur udara luar pada bulan September 2013-Agustus 2014 yang

masuk dalam zona kenyamanan termal sesuaistandar SNI (Standar Nasional Indonesia) berdasarkan nilai rata-rata harian diatas 50% terjadi pada bulan Oktober, Desember, Januari, Februari, Maret, Mei, Juli dan Agustus. Nilai rata-rata dibawah 50% pada bulan September, November, April dan Juni. Temperatur udara maksimum tertinggi terjadi pada bulan September 2013 dan nilai minimum terendah pada bulan Agustus 2014. Untuk kelembaban relatif udara pada bulan September, Oktober, Maret, Mei, Juli dan Agustus masuk dalam zona kenyamanan termal diatas 50%, dan bulan November, Desember, Januari, April serta Juni berada dibawah 50%. Nilai rata-rata maksimum tertinggi terjadi pada bulan Desember 2013 dan minimum terendah pada bulan September 2013. Data temperatur dan kelembaban relatif udara di ruang luar ( outdoor ) bangunan kampus baru Fakultas Teknik Unhas Gowa tidak sepenuhnya mendukung penerapan system penghawaan pasif untuk mencapai kenyamanan termal dalam ruangan ( indoor )”.

Menurut Budioko (2016) “Salah satu perkembangan teknologi internet pada saat ini adalah Internet of thing. Ada beberapa protokol yang berkembang untuk implementasi Internet of things diantaranya adalah Message Queue Telemetry Transport (MQTT). Pada artikel ini dibahas implementasi protokol MQTT untuk sistem monitoring suhu jarak jauh. Implementasi sistem menggunakan sensor suhu LM35, Arduino UNO dan modul wifi Esp8266 ver 01. Prototype sistem berhasil direalisasikan baik pada Node Sensor maupun Node Monitor. Berdasarkan hasil pengujian, sistem dapat melakukan koneksi ke server MQTT lokal maupun server MQTT global, mampu mengirim data (publish) dan menerima data (subscribe)”.

Merujuk pada beberapa penelitian yang dikemukakan diatas, dengan itu dalam penelitian ini penulis merancang dan membangun sebuah sistem pengukuran dan pemantauan suhu udara secara *real time* berbasis IoT (*Internet of Things*) dimana data yang didapat dari alat tersebut akan secara otomatis dikirim ke *database* kaa *platform*. Dan kaa *platform* merupakan produk *open source* dimana seorang programmer dapat melakukan pengembangan sesuai dengan kebutuhan.

## **2.2 Landasan Teori**

Di dalam landasan teori terdapat beberapa teori yang bersangkutan dengan penelitian yang akan dilakukan, antara lain :

### **2.2.1 *Raspberry Pi***

Menurut Maik Schmidt (2012) “*Raspberry Pi* adalah sebuah komputer berukuran sebesar kartu kredit yang terhubung ke televisi dan sebuah keyboard. Komputer kecil ini bisa digunakan untuk proyek-proyek elektronik, dan hal lainnya yang bisa dilakukan oleh desktop komputer seperti sebagai mesin pengolah kata, games, dan perangkat ini juga mampu memainkan video beresolusi tinggi”.

### **2.2.2 *Waspnote***

Menurut *waspnote technical guide* (2018) “*Waspnote* didasarkan pada arsitektur modular. Idenya adalah mengintegrasikan modul yang dibutuhkan disetiap perangkat. Modul ini bisa diubah dan diperluas sesuai kebutuhan”.

### **2.2.3 *XBee PRO***

Menurut Abdul Kadir (2017) “*XBee* adalah keluarga yang dihasilkan oleh digi Corporation yang memungkinkan untuk mengirim data dari satu tempat ketempat yang lainnya secara nirkabel. Jarak yang ditangani sangat bervariasi dari puluhan meter hingga berkilo-kilo meter. Sebagai contoh, modul XBee 1mW Wire Antenna mampu melewati data hingga 100 meter dalam ruangan terbuka, sedangkan XBee Pro 60mW Wire Antenna mampu menjangkau hingga 1,6 kilometer”.

### **2.2.4 *Kaa Platform***

Menurut *overview kaaproject* (2018) “*kaa* adalah *platform middleware multi* guna untuk Internet dari Hal-hal yang memungkinkan penyelesaian lengkap solusi *end-to-end* IoT, aplikasi yang terhubung, dan produk cerdas. Platform Kaa

menyediakan *toolkit* terbuka dan kaya fitur untuk pengembangan produk IoT dan dengan demikian mengurangi biaya, risiko, dan waktu ke pasar secara dramatis”.

#### **2.2.5 Sensor *Thermal* (sensor suhu)**

Menurut D Sharon, dkk (1982) “Sensor thermal adalah sensor yang digunakan untuk mendeteksi gejala perubahan panas/temperature/suhu pada suatu dimensi benda atau dimensi ruang tertentu”.

#### **2.2.6 *Hypertext Markup Language* (HTML)**

Menurut Winarno dan Utomo (2010) “HTML singkatan dari Hypertext Markup Language dan berguna untuk menampilkan halaman web”.

#### **2.2.7 MySQL**

Menurut Anhar (2010) “MySQL (My Structure Query Language) adalah sebuah perangkat lunak sistem manajemen basis data SQL Database Management System atau DBMS dari sekian banyak DBMS seperti Oracle, MS SQL, Postagre SQL dan lainnya”.

#### **2.2.8 *Hypertext Preprocessor* (PHP)**

Pengertian PHP menurut Anhar (2010) “PHP adalah (*PHP Hypertext Preprocessor*) adalah bahasa pemrograman web berupa *script* yang dapat diintegrasikan dengan HTML”.

#### **2.2.9 Java**

Java adalah suatu teknologi di dunia software komputer, yang merupakan suatu bahasa pemrograman, dan sekaligus suatu platform. Sebagai bahasa pemrograman, java dikenal sebagai bahasa pemrograman tingkat tinggi. Java mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++. Java merupakan bahasa pemrograman berorientasi objek yang merupakan paradigma pemrograman masa depan. Sebagai bahasa pemrograman Java dirancang menjadi handal dan aman. Java juga dirancang agar dapat dijalankan di semua platform.



Dan juga dirancang untuk menghasilkan aplikasi – aplikasi dengan performansi yang terbaik, seperti aplikasi database Oracle 8i/9i yang core-nya dibangun menggunakan bahasa pemrograman Java(Suyanto, 2015).



**Gambar 2.1 Java**  
(sumber: *en.wikipedia.org*)

#### **2.2.10 Python**

Python merupakan bahasa pemrograman berorientasi objek dinamis yang dapat digunakan untuk berbagai macam pengembangan perangkat lunak (Nugroho dkk, 2013). Dimana bahasa ini dijalankan pada perangkat lunak EMC2-AXIS yang berada di sistem operasi Linux Ubuntu. Selain itu, bahasa ini mudah digunakan karena memiliki susunan tata bahasa yang jernih serta memiliki pustaka yang luas dan bahasa ini dapat diintegrasikan dengan bahasa lain seperti C/C++ dan Java(Gunawan dan Linggarjati, 2012).

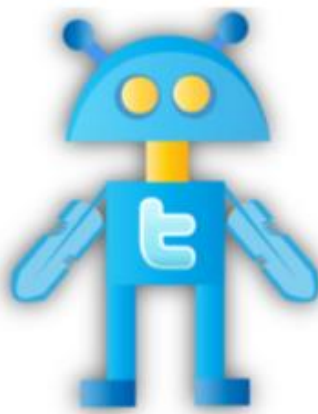


**Gambar 2.2 Python**  
(sumber: *www.raspberry.org*)

### 2.2.11 Twitterbot

Twitterbot adalah kepanjangan dari Twitter dan Robot. Twitter adalah sebuah situs web yang dimiliki dan dioperasikan oleh Twitter Inc., menawarkan jaringan sosial berupa microblogging sehingga memungkinkan pengguna untuk mengirim dan membaca pesan di Twitter. Twitter dengan pengguna lebih dari 500 juta dan 400 juta tweet perhari memungkinkan pengguna untuk berbagi pesan menggunakan tweet dan juga memungkinkan tweet tersebut otomatis mengirim ke postingan facebook. Pengguna Twitter diklasifikasikan menjadi beberapa kategori yaitu manusia (legitimate user), cyborg, dan bot. Bot merupakan kepanjangan dari Robot yang dikenal sebagai program otomatisasi. Selain untuk menghasilkan pesan dalam jumlah yang banyak beberapa bot diciptakan untuk menyebarkan pesan berbahaya (Aditya, 2016).

Twitterbot merupakan sebuah program bot yang digunakan untuk menghasilkan otomatis posting pada layanan microblogging Twitter, atau secara otomatis mengikuti pengguna Twitter. Tweets otomatis ini sering dipandang sebagai suatu bentuk hal yang merugikan atau menyenangkan. Beberapa pengguna Twitter bahkan sering menggunakan Twitterbot ini untuk membantu diri mereka sendiri dalam penjadwalan atau pengingat dan kemudian memposting tweet tersebut otomatis ke Facebook.



**Gambar 2.3 Twitterbots**  
(sumber: [botwiki.org/bots/Twitterbots](http://botwiki.org/bots/Twitterbots))

## **BAB III**

### **PERANCANGAN**

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

#### **3.1 Data dan Alat Penelitian**

Didalam penelitian terdapat data dan alat yang digunakan untuk membantu penelitian, adapun data dan alat tersebut antara lain :

##### **3.1.1 Bahan Penelitian**

Pengumpulan data dalam penelitian ini dilakukan untuk memperoleh informasi yang dibutuhkan dalam mencapai tujuan penelitian, yang mana tujuan penelitian yang diungkapkan dalam bentuk hipotesis ini merupakan jawaban sementara terhadap pertanyaan penelitian. Data yang dibutuhkan pada penelitian berupa data suhu udara kemudian hasil data tersebut diinputkan langsung ke mini PC.

##### **3.1.2 Alat Penelitian**

Alat yang digunakan pada proses penelitian ini meliputi perangkat keras (hardware) dan perangkat lunak (software).

###### **3.1.2.1 Perangkat Keras/*Hardware***

*Hardware* yang digunakan dalam penelitian ini antara lain :

###### **1. *Personal computer***

Sepesifikasi *personal computer* yang digunakan adalah *CPU* yaitu Intel® Core™ i5-760 CPU (2.80GHz), *Memory* 8GB dan *HDD* 1TB. Digunakan

untuk sebagai server utama, dimana pada *personal computer* tersebut akan diinstall kaa platform sebagai media penyimpanan data.

2. Sensor suhu LM35

Berfungsi untuk mengetahui suhu udara yang berada didekat sensor tersebut.

3. Komunikasi *wireless serial*

Berfungsi untuk berkomunikasi secara *wireless serial* antara mikrokontroler dengan mini PC. Komunikasi *wireless serial* disini menggunakan Xbee PRO

4. Mikrokontroler

Berfungsi sebagai pengontrol rangkaian elektronik sesuai dengan yang telah kita program sebelumnya dan biasanya mikrokontroler dapat menyimpan program didalamnya. Mikrokontroler yang digunakan adalah Waspote PRO.

5. Baterai *lithium-ion*

Berfungsi sebagai sumber daya untuk mikrokontroler. Disini menggunakan baterai dengan tegangan 3.7V serta kapasitas yang bisa ditampung adalah 6600mAh.

6. Mini PC

Berfungsi sebagai tempat melakukan konfigurasi komunikasi antara sistem sehingga saling berhubungan, baik dari data sensor dengan media penyimpanan yaitu database. Mini PC yang digunakan adalah Raspberry Pi.

7. *Power Bank*

Berfungsi untuk memberikan daya kepada mini PC, agar mini PC bisa dihidupkan.

### **3.1.2.2 Perangkat Lunak/Software**

*Software* yang digunakan dalam penelitian ini antara lain :

1. *Waspote PRO IDE*

Digunakan sebagai tempat melakukan program pada semua sensor yang digunakan untuk *end device* ataupun untuk sensor suhu nantinya.

2. X-CUT

Digunakan sebagai tempat melakukan program dalam melakukan pembacaan pada *port usb gateway*.

3. *Netbean 8.2*

Merupakan *IDE* yang digunakan untuk melakukan koding bahasa pemrograman *java* untuk membaca data yang didapat dari komunikasi *wireless serial* dan memasukkan data tersebut kedalam database server.

4. *Sistem Operasi Linux Ubuntu Kaa - Sanbox v0.10.0*

Berfungsi sebagai media penyimpanan data, dimana data yang telah di dapat dari sensor di simpan disini.

5. Raspbian

Merupakan sistem operasi yang digunakan pada mini PC Raspberry Pi yang digunakan untuk membangun konektivitas dari perancangan yang dilakukan. Merupakan versi dari OS Debian yang dikhususkan untuk Raspberry Pi

6. PHP (*Hypertext Preprocessor*)

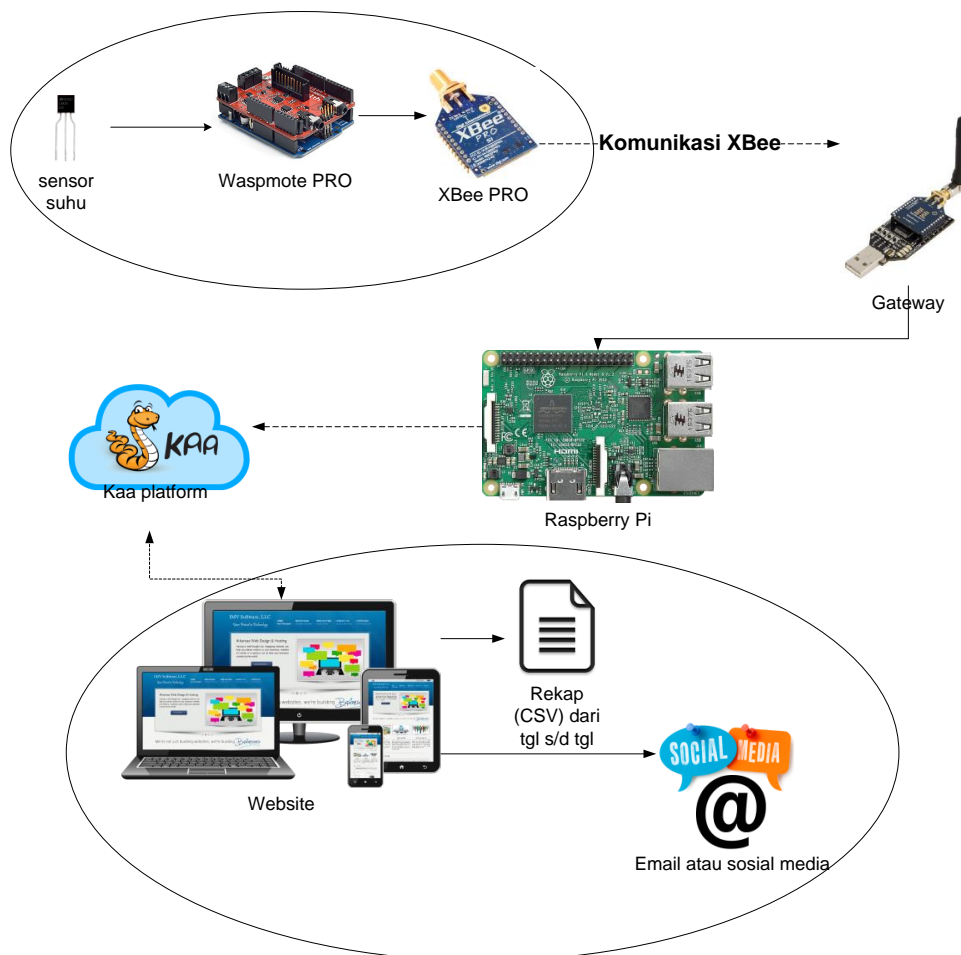
Merupakan bahasa pemrograman yang digunakan untuk menampilkan data yang ada didalam server kaa platform ke halama sebuah website, sehingga bisa dilihat oleh orang banyak.

## 3.2 Perancangan Sistem

Referensi yang digunakan sebagai landasan teori dalam pembuatan Sistem Pengukuran Dan Pemantauan Suhu Udara Secara *Real-Time* Berbasis IoT meliputi:

### 3.2.1 Perancangan Desain Sistem Secara Umum

Pada bagian ini akan dijabarkan perancangan sistem secara umum dan akan ditampilkan semua peralatan yang dibutuhkan untuk membuat sistem pengukuran dan pemantauan suhu udara secara *real time* sekaligus rancangan alur kerja dari alat-alat tersebut. Rancangan ini berguna untuk mempermudah dalam mengetahui alur kerja sistem secara garis besar. Adapun desain sistem secara umum yang membahas alur kerja sistem dapat dilihat pada gambar 3.1 berikut:

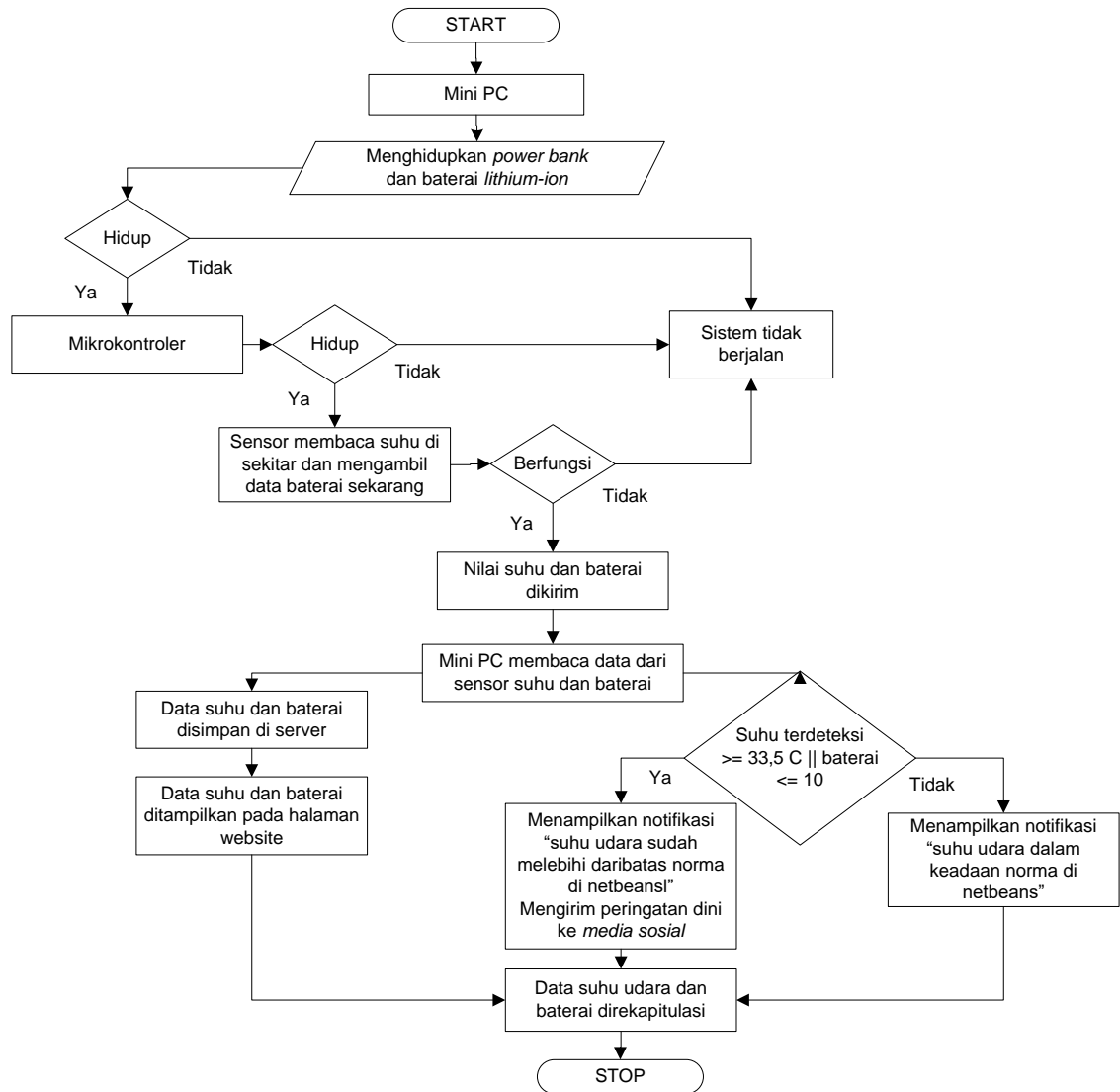


**Gambar 3.1** Gambaran Perancangan Sistem  
(sumber: Data Olahan)

Berdasarkan Gambar 3.1, penelitian ini akan merancang sistem monitoring suhu udara dengan menggunakan sensor suhu. Dimana sensor tersebut terhubung ke *waspmute*. *waspmute* menerima data dari sensor suhu, yang kemudian data tersebut di *sorting* dan kemudian mengirimkannya ke *Raspberry Pi* menggunakan *XBee PRO*. Data yang sampai di *rasberry pi* tadinya diparsing, data yang sudah diparsing kemudian dimasukkan ke *database server kaa platform*, yang nantinya data tersebut dapat dilihat dari berbagai jenis perangkat, melalui sebuah tampilan halaman *website*.

### 3.2.2 Rancangan Diagram Alir Sistem

Adapun rancangan diagram alir dari sistem yang akan dibuat dapat dilihat dari *flowchart* yang terdapat pada Gambar 3.2 dibawah ini.



**Gambar 3.2** *Flowchart* dari Sistem yang Dibuat  
(sumber: Data Olahan)

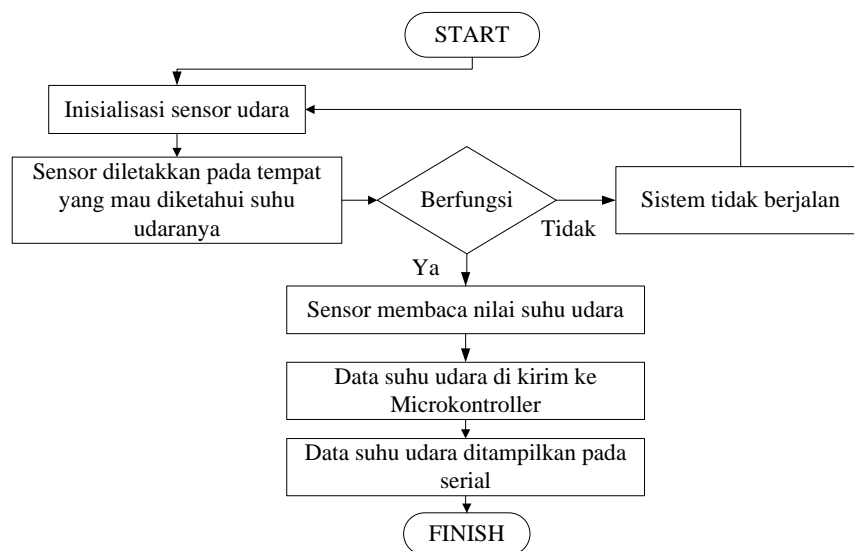
Gambar 3.2 menjabarkan dari dihidupkannya sumber daya (*power bank* dan baterai *lithium-ion*) hingga dapat menghidupkan mini PC dan mikrokontroler. Setelah keduanya hidup, maka sensor suhu akan membaca nilai suhu disekitar sensor suhu, setelah mendapatkan nilai atau data, kemudian nilai atau data



tersebut dikirimkan ke mikrokontroler dan langsung diteruskan ke mini PC. Dari mini PC data nilai suhu udara disimpan pada *database server* dan apabila suhu udara  $\geq 33,5$  C atau baterai  $\leq 10$  maka akan dikirim postingan ke twitter bahwa suhu udara sedang tidak normal atau baterai sudah mau habis. Data dari suhu udara tersebut dapat dilihat selengkapnya pada halaman website oleh pengguna. Pada halaman website tersebut dapat dibuat rekap mengenai data suhu udara dari tanggal awal sampai dengan tanggal akhir yang ingin dibuat rekapnya.

### 3.2.3 Rancangan Diagram Alir Sistem Sensor

Gambar 3.3 merupakan diagram alir yang menceritakan tentang proses kerja sensor yang diawali dari inisiasi sensor yang berfungsi untuk melakukan proses konfigurasi agar bisa membaca nilai suhu udara pada suatu tempat.



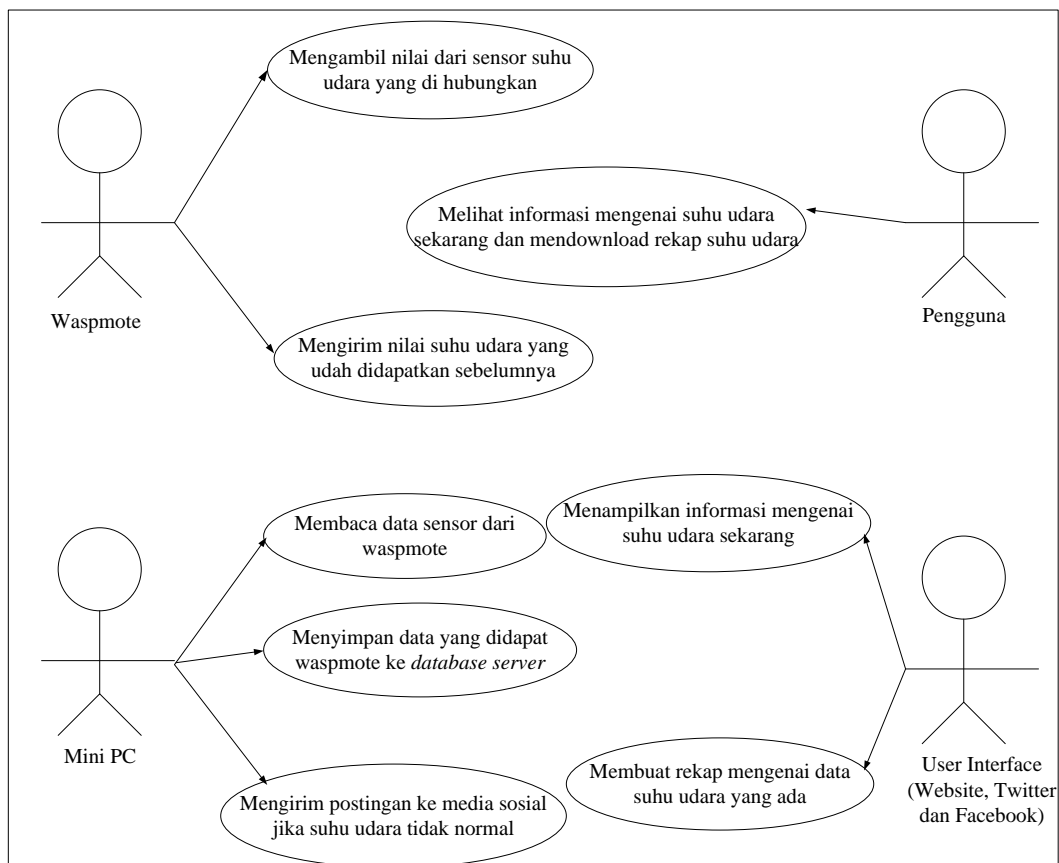
**Gambar 3.3** *Flowchart Sistem Sensor*  
(sumber: Data Olahan)

Inisiasi berfungsi untuk mengkonfigurasi sensor agar bisa berjalan seperti mana mestinya, dimana sensor tersebut dapat mengukur suhu udara dengan baik dan benar. Setelah inisiasi berhasil maka sensor suhu udara tersebut diletakkan pada tempat yang mau diketahui suhu udaranya. Kemudian sensor tersebut akan membaca nilai suhu udara di sekitar sensor tersebut, yang nantinya nilai tersebut dikirim ke mikrokontroler untuk kemudian data tersebut ditampilkan pada serial.

Namun, apabila sensor tidak dapat membaca suhu tubuh anak, maka diperlukan inisiasi ulang terhadap sensor suhu tersebut.

### 3.2.4 Rancangan Diagram Kasus Penggunaan Sistem

Diagram kasus penggunaan sistem menggambarkan fungsional sistem serta aktor yang terlibat. Gambar 3.4 menggambarkan bahwa sistem memiliki 4 aktor utama yaitu pengguna (*user*), Arduino, mini PC, dan *User Interface* (Aplikasi Android).

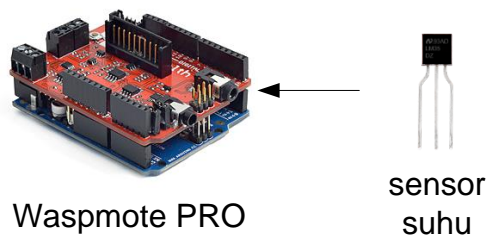


**Gambar 3.4** Diagram Kasus Penggunaan Sistem  
(*sumber: Data Olahan*)

### 3.2.5 Perancangan Hardware

Perancangan pada hardware ini terdapat pada perancangan sensor, perancangan *Gateway* dan perancangan *database*. Adapun penjelasannya sebagai berikut :

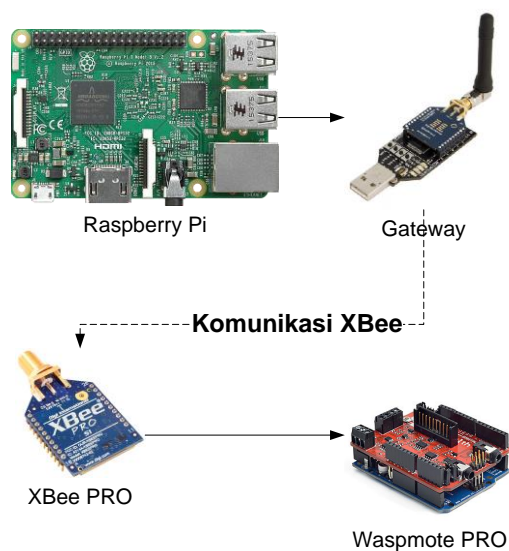
#### 1. Perancangan pada sensor



**Gambar 3.5** Perancangan Pada Sensor  
(*sumber: Data Olahan*)

Pada Gambar 3.5 dimana sensor berguna untuk mendeteksi kondisi suhu udara yang ada disekitar tempat sensor tersebut dan meneruskan data yang didapat ke perangkat *waspmute*.

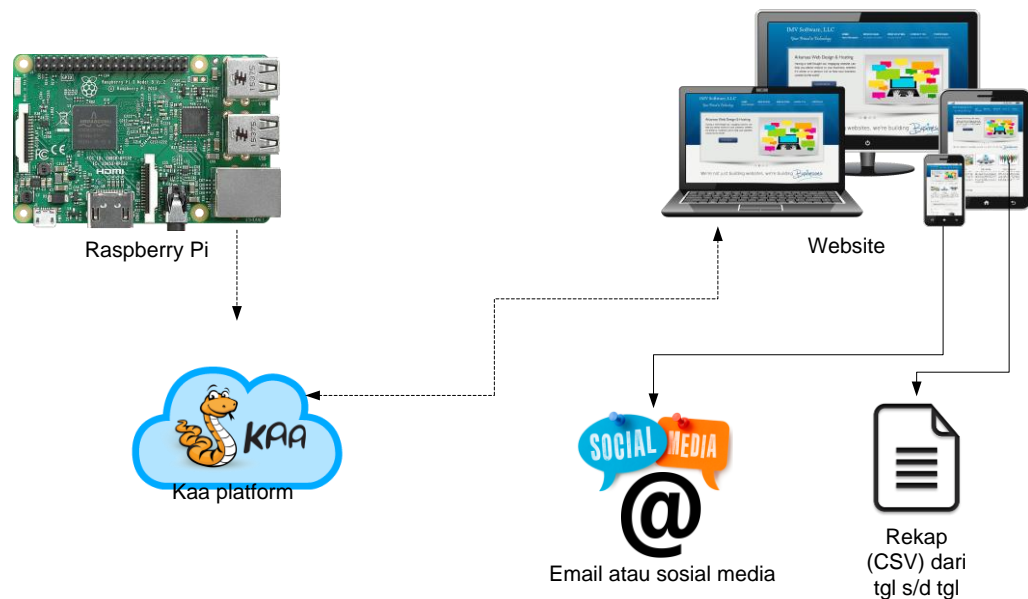
#### 2. Perancangan Gateway



**Gambar 3.6** Perancangan *Gateway*  
(*sumber: Data Olahan*)

Pada Gambar 3.6 dijelaskan bahwa *waspmote* dihubungkan ke mini PC menggunakan *xbee pro*. Pada mini PC akan di *install raspbian* adalah nama *operasi sistem* yang disediakan langsung dari mini PC tersebut. Kemudian dilanjutkan dengan menginstall *netbeans (java)* yang berfungsi untuk *open port* di mini PC sekaligus dengan *netbeans* akan dilakukan beberapa program supaya data yang didapat dari mikrokontroler bisa disimpan ke *database server*, dan program apabila suhu udara tidak normal akan dikirim postingan ke twitter, sehingga pengguna bisa langsung mengetahuinya sebagai peringatan dini.

### 3. Perancangan Database



**Gambar 3.7** Perancangan Database  
(sumber: Data Olahan)

Pada Gambar 3.7 dijelaskan bahwa mini PC mengirim data ke *database kaa platform*. Dari *database kaa platform* data tersebut diteruskan ke sebuah halaman *website* yang bertujuan agar data tersebut nantinya dapat dilihat dan diakses oleh pengguna. Dimana data yang ada dapat dilakukan rekap dan jika nilai suhu udara tersebut tidak normal atau nilai yang ada melebihi dari batas yang telah ditentukan, maka nantinya akan dikirim notifikasi ke media sosial.

### 3.2.6 Perancangan Basis Data

Perancangan *database* untuk sistem penyimpanan agar dapat ditampilkan pada sebuah halaman website yang telah dibuat sebelumnya. Dimana pada *database* menyimpan data-data mengenai nilai dari suhu udara pada tempat yang telah ditentukan sebelumnya, dimana data yang didapat seperti suhu udara, persentase baterai, dan serta waktu data tersebut didapat. Berikut Perancangan *database* yang dibuat sebagai berikut:

Nama *database*: suhu\_udara

Nama tabel: SensorData

**Tabel 3.1** Perancangan *Database*

No	Nama Field	Jenis	Fungsi
1	id_secret	String	Berfungsi untuk mendefinisikan id unik yang mengidentifikasi masingmasing perangkat waspmote.
2	id_wasp	String	Berfungsi untuk mendefinisikan string yang didefinisikan oleh pengguna yang dapat mengidentifikasi setiap Waspmote dalam penggunaan jaringan.
3	frame_number	String	Berfungsi untuk mendefinisikan jumlah urutan frame . Urutan nomor ini digunakan untuk mendeteksi kehilangan frame. counter ini adalah 8 - bit , sehingga ia

			dimulai dari 0 sampai 255. Namun , karena berupa ASCII frame, jumlah ini dikonversi ke string sehingga dapat dipahami . Setiap kali counter mencapai maksimum 255 , ini akan mereset ke 0..
4	bat	String	Berfungsi untuk mendefinisikan pada jumlah batrei yang tersisa
5	tca	String	Berfungsi untuk mendefinisikan pada tingkat temperatur pada lingkungan sekitar
6	timeStamp	long	Berfungsi untuk mendefinisikan waktu dimana data sensor tersebut di kumpulkan

### 3.2.7 Perancangan Antarmuka

Perancangan antarmuka yang digunakan adalah *website*. Ada beberapa perancangan antarmuka pada *website* sebagai *output* dari sistem pengukuran suhu udara. Berikut perancangan antarmuka *website* untuk sistem pemantauan dan pengukura suhu udara secara *real time*.

### 3.2.7.1 Perancangan Tampilan Dashboard

DASHBOARD	<div> <div>Jumlah Suhu</div> <div>Jumlah Batray</div> </div> <div>Area tabel</div>						
Real Time							
Suhu dan Batray							

**Gambar 3.8** Perancangan Tampilan Dashboard  
(*sumber: Data Olahan*)

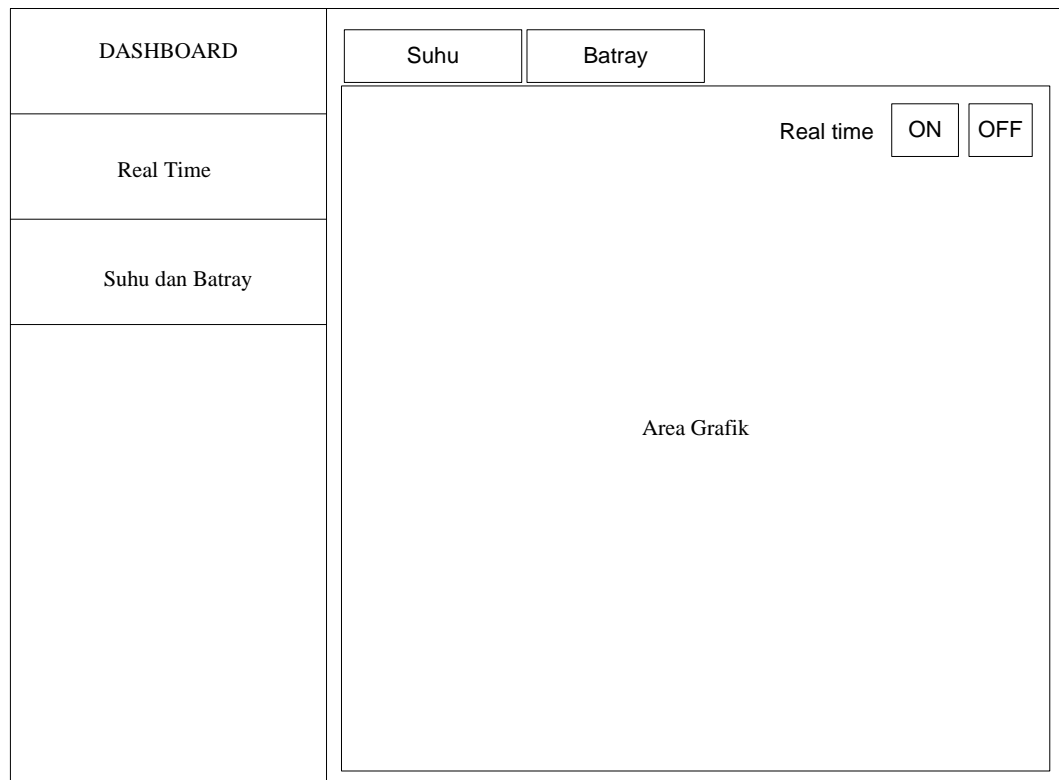
### 3.2.7.2 Perancangan Tampilan Suhu Udara dan Baterai

DASHBOARD	<div>Tanggal awal</div> <div>Tanggal akhir</div>						
Real Time	<div>Download</div>						
Suhu dan Batray	<div>Area Grafik</div>						
	Area tabel						

**Gambar 3.9** Perancangan Tampilan Suhu Udara dan Baterai  
(*sumber: Data Olahan*)



### 3.2.7.3 Perancangan Tampilan Suhu Udara dan Baterai Secara *Real-Time*



**Gambar 3.10** Perancangan Tampilan Suhu Udara dan Baterai *real-time*  
(*sumber: Data Olahan*)

## **BAB IV**

### **HASIL DAN PENGUJIAN**

#### **4.1 Hasil**

Penelitian ini menghasilkan alat pengukuran suhu udara dimana dengan menggunakan sistem ini pengguna dapat mengetahui informasi mengenai seputar suhu udara, pada tempat yang telah di pasang alat tersebut. Selain itu alat ini akan menampilkan informasi yang telah di dapat melalui sebuah halaman website, dan pada halaman website pengguna dapat mengunduh rekap mengenai suhu udara berdasarkan tanggal, yang nantinya jenis file rekapnya adalah dalam format CSV. Pada alat ini juga dapat memposting status suhu udara ke Twitter, dimana jika suhu udara melewati batas yang telah ditentukan atau baterai pada mikrokontroler tinggal sedikit, yang bertujuan sebagai peringatan dini. Informasi yang di tampilkan di website berupa grafik tentang data suhu udara dan persentase baterai yang tersisa.

#### **4.2 Pengujian**

Pada tahap ini dilakukan pengujian bertujuan untuk mengimplementasikan rancangan yang telah dibuat dengan menjalankan dan melakukan uji coba pada rancang bangun alat pengukuran suhu udara ini yang menghasilkan informasi berupa grafik di website yang nantinya bisa dilihat secara *real-time* dan informasi di twitter apabila suhu udara melebihi dari yang telah ditentukan, serta dapat mengunduh rekap berdasarkan tanggal sesuai dengan data yang telah didapat sebelumnya dari sensor suhu, file rekapnya menggunakan format CSV.

#### 4.2.1 Lingkungan Pengujian

Lingkungan uji coba dilakukan langsung pada tempat yang ingin diketahui suhu udaranya, dalam pengujian ini melibatkan 2 tempat lobi yang telah ditentukan yaitu di politeknik negeri bengkalis. Uji coba dilakukan dengan menggunakan komponen berupa 1 buah sensor suhu udara yang disiposikan pada tempat lobi yang telah ditentukan, sebuah mini PC Raspberry Pi, sebuah mikrokontroler Wasp mote IDE, dan sebuah batrai dan *power bank* sebagai sumber daya.

#### 4.2.2 Persiapan Pengujian

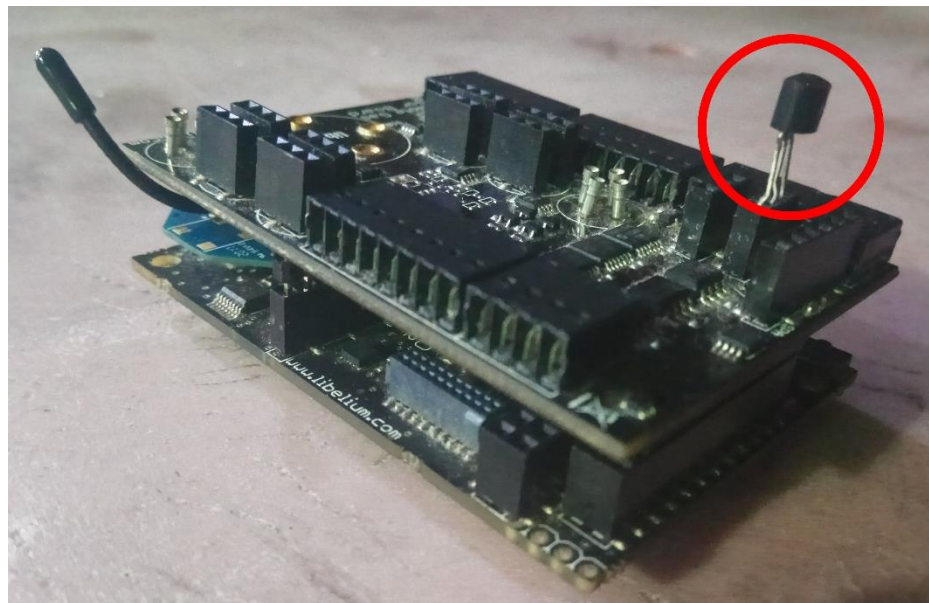
Persiapan untuk uji coba dibagi menjadi dua, yaitu persiapan pada perangkat keras dan pada perangkat lunak.

##### 4.2.2.1 Persiapan Perangkat Keras

Persiapan perangkat keras meliputi mini PC, mikrokontroler, dan sensor suhu udara. Berikut Penjelasan persiapan yang akan dilakukan:

##### 1. Persiapan Sensor

Pada persiapan perangkat sensor, komponen yang dibutuhkan adalah mikrokontroler yang dihubungkan dengan sensor. Dapat dilihat pada Gambar 4.1 berikut:



**Gambar 4.1** Persiapan Sensor  
(sumber: Data Olahan)

## 2. Persiapan Mini PC

Pada persiapan perangkat mini PC, komponen yang dibutuhkan adalah mikrokontroler yang dihubungkan menggunakan *xbee pro* sebagai berkomunikasi secara *wireless serial* antara mikrokontroler dengan mini PC. Dapat dilihat pada Gambar 4.2 berikut.



**Gambar 4.2** Persiapan Mini PC  
(sumber: Data Olahan)

### 4.2.2.2 Persiapan Perangkat Lunak

Persiapan perangkat lunak yaitu melewati beberapa tahapan dalam membuat pengkodean (*coding*) untuk sensor, melakukan *install library* untuk membaca *port* USB mikrokontroler yang berfungsi dalam proses *rx-tx* (*receive-transmit*) data dari mikrokontroler ke mini PC, membuat *coding* pemrograman java pada NetBeans IDE untuk menyimpan data ke *database server*.

#### 1. Coding Program Sensor

Coding program sensor merupakan proses pengkodean yang berfungsi dalam menjalankan sensor suhu yang berfungsi untuk membaca nilai suhu udara. Kode menggunakan bahasa C dan pada sensor suhu menggunakan *library waspSensorGas\_v20*.

Coding dapat dilihat pada gambar berikut:

```
#include <WaspXBee802.h>
#include <WaspFrame.h>
#include <WaspSensorGas_v20.h>
#define GAIN 1//GAIN of the sensor stage

float temperatureVal; //variable to store the read
Temperature

//Variabel for sending data
char* MAC_ADDRESS="0013A200416C36FE";// Destination
MAC address
packetXBee* packet; //Pointer to an XBee packet
structure

void setup()
{
    USB.ON();// Open USB port
    xbee802.ON();// Turn ON XBEE

    //Only printed on USB, not SD
    USB.println(F("Starting Sensing Environmental"));
    USB.println(F("Wait a minute....."));
    USB.println(F("-----
---"));

    //begin sensor configuration
    SensorGasv20.ON();// Turn on the sensor board
    RTC.ON();// Turn on the RTC

    delay(5000);// sensor warm up time
    //end sensor configuration
}

//end setup

void loop()
{
    //Read the sensor Temperature
    temperatureVal =
    SensorGasv20.readValue(SENS_TEMPERATURE);

    // begin XBEE Communication
    // 1. Create ASCII frame
    frame.createFrame(ASCII, "sensor01");
    frame.addSensor(SENSOR_TCA, temperatureVal);
    frame.addSensor(SENSOR_BAT, (uint8_t)
    PWR.getBatteryLevel());
    frame.addSensor(SENSOR_TIME, RTC.hour, RTC.minute,
    RTC.second);
    frame.showFrame();

    // 2. SEND Packet
    // set parameters to packet:
```

```

    packet=(packetXBee*) calloc(1,sizeof(packetXBee));
    // Memory allocation
    packet->mode=UNICAST; // Choose transmission mode:
    UNICAST or BROADCAST

    // set destination XBee parameters to packet
    xbee802.setDestinationParams( packet, MAC_ADDRESS,
    frame.buffer, frame.length);
    xbee802.sendXBee(packet); // Send XBee packet

    // Check TX flag
    if( xbee802.error_TX == 0 )
    {
        USB.println(F("send ok"));
        Utils.blinkGreenLED();
    }
    else
    {
        USB.println(F("send error"));
        Utils.blinkRedLED();
    }

    // Free variables
    free(packet);
    packet=NULL;
    // end XBEE Communication

    delay(5000); // Delay for receiving data from
    environtment
}

```

**Gambar 4.3** Coding Sensor Suhu Udara  
(sumber: Data Olahan)

## 2. Melakukan *Install Library*

Tahap awal yang penting dalam pembacaan *port* mikrokontroler pada mini PC adalah diharuskan untuk melakukan instalasi *library*. Pembacaan *port* pada waspmote merupakan langkah awal yang penting untuk melakukan pengambilan data nilai suhu udara dari mikrokontroler ke *server* pada mini PC. Pada mini PC, pembacaan *port* memerlukan sebuah *library* yang harus diinstal terlebih dahulu sebelum melanjutkan ke proses selanjutnya. Adapun *library* yang harus diinstal adalah *library openjdk-8-jre* seperti pada Gambar 4.4 berikut:

```

pi@raspberrypi:~ $ sudo apt-get install openjdk-8-jre
Reading package lists... Done
Building dependency tree
Reading state information... Done
openjdk-8-jre is already the newest version (8u171-b11-1~deb9u1).
openjdk-8-jre set to manually installed.
The following packages were automatically installed and are no longer required:
  libexiv2-14 wolframscript
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 19 not upgraded.

```

**Gambar 4.4** Instalasi *Library* Pembacaan *Port*  
(sumber: Data Olahan)

### 3. Pengujian Program

Pengkodean program menggunakan Aplikasi NetBeans IDE dan menggunakan bahasa Java, pengkodean baru bisa dilakukan tanpa hambatan apabila telah melakukan instalasi *library openjdk-8-jre*. Apabila library telah diinstal, maka selanjutnya dapat melakukan pengkodean program Java pada NetBeans IDE yang berfungsi dalam membaca data sensor berupa nilai suhu dari mikrokontroler (proses *rx-tx*). Proses *rx-tx* (*Receive-Transmit*) adalah kegiatan membuka *port* dari mini PC dan mikrokontroler agar dapat melakukan pengiriman data nilai suhu dari mikrokontroler ke mini PC. Data yang dikirim tersebut dilakukan penangkapan (*catch*) dan pemisahan (*parsing*) agar dapat membedakan antara nilai suhu dari sensor satu dengan sensor lainnya. Pada program java ini juga dilakukan perintah untuk menampilkan nilai suhu udara dan berapa persentase baterai, yang tersedia dari sensor yang dipasang pada mikrokontroler untuk bagian suhu. Nilai suhu dan baterai tersebut nantinya yang akan dikirim ke *database*. Setelah dilakukan proses *parsing*, maka data tersebut disimpan ke *database*. Adapun *coding* untuk proses pembacaan *port* dan proses *rx-tx* pada mikrokontroler tersebut dapat dilihat pada Gambar 4.5 dibawah ini:

```

/**
 * @param args the command line arguments
 */
SerialPort serialPort;
    /** The port we're normally going to use. */
    private static final String PORT NAMES[] = {

```

```

        "/dev/ttyACM0", //for Ubuntu
        "/dev/tty.usbserial-A9007UX1", // Mac OS X
        "/dev/ttyUSB0", // Linux
        "COM3", // Windows
    };
    /**
     * A BufferedReader which will be fed by a
    InputStreamReader
     * converting the bytes into characters
     * making the displayed results codepage independent
     */
    private BufferedReader input;
    /** The output stream to the port */
    private OutputStream output;
    /** Milliseconds to block while waiting for port
    open */
    private static final int TIME_OUT = 2000;
    /** Default bits per second for COM port. */
    private static final int DATA_RATE = 115200;
    static String
    sampah,sampah1,sampah2,sampah3,sampah4,sampah5,id_was
    pmote,sensor,urutan,suhu,time,batray;

    public void initialize() {
        CommPortIdentifier portId = null;
        Enumeration portEnum =
        CommPortIdentifier.getPortIdentifiers();

        //First, Find an instance of serial port as set in
        PORT_NAMES.
        while (portEnum.hasMoreElements()) {
            CommPortIdentifier currPortId =
            (CommPortIdentifier) portEnum.nextElement();
            for (String portName : PORT_NAMES) {
                if
            (currPortId.getName().equals(portName)) {
                portId = currPortId;
                break;
            }
        }
        if (portId == null) {
            System.out.println("Could not find COM
port.");
            return;
        }

        try {
            // open serial port, and use class name for
            the appName.
            serialPort = (SerialPort)
            portId.open(this.getClass().getName(),
                TIME_OUT);

            // set port parameters
            serialPort.setSerialPortParams(DATA_RATE,

```



```

        SerialPort.DATABITS_8,
        SerialPort.STOPBITS_1,
        SerialPort.PARITY_NONE);

    // open the streams
    input = new BufferedReader(new
InputStreamReader(serialPort.getInputStream()));
    output = serialPort.getOutputStream();

    // add event listeners
    serialPort.addEventListener(this);
    serialPort.notifyOnDataAvailable(true);
} catch (Exception e) {
    System.err.println(e.toString());
}
}

/**
 * This should be called when you stop using the
port.
 * This will prevent port locking on platforms like
Linux.
 */
public synchronized void close() {
    if (serialPort != null) {
        serialPort.removeEventListener();
        serialPort.close();
    }
}
}

```

**Gambar 4.5** Source Code Baca Port dan Koneksi Mikrokontroler  
(sumber: Data Olahan)

Sementara untuk proses pemisahan (*parsing*) data yang telah didapat seperti id waspmote, frame number, battery, sensor suhu dapat menggunakan *parsing*. Dimana *parsing* berguna untuk memisah nilai id waspmote, frame number, battery, sensor suhu dimana nilai tersebut masih terhitung sebagai satu kesatuan ketika dikirim dari mikrokontroler. Dalam hal ini, pemisahan menggunakan symbol '#:'. Proses *parsing* dapat dilihat pada Gambar 4.6 dibawah ini:

```

public synchronized void
serialEvent(SerialPortEvent oEvent) {
    if (oEvent.getEventType() ==
SerialPortEvent.DATA_AVAILABLE) {
        try {
            String nilai_output =
input.readLine();
            String[] bilah =
nilai_output.split("#:");
            sampah = bilah[0];

```

```

        id_waspmote    = bilah[1];
        sensor         = bilah[2];
        urutan         = bilah[3];
        sampah1        = bilah[4];
        suhu           = bilah[5];
        sampah2        = bilah[6];
        batray          = bilah[7];
        sampah3        = bilah[8];
        time            = bilah[9];
        sampah4        = bilah[10];

        System.out.println("id waspmote   : "+
id_waspmote);
        System.out.println("sensor      : "+
sensor);
        System.out.println("urutan      : "+
urutan);
        System.out.println("suhu       : "+
suhu);
        //System.out.println("time      : "+
time);
        System.out.println("batray     : "+
batray);

        } catch (Exception e) {
            //System.err.println(e.toString());
        }
    }

    // Ignore all the other eventTypes, but you
    should consider the other ones.
}

```

**Gambar 4.6** Source Code Parsing data  
(sumber: Data Olahan)

Ketika telah dilakukan pemisahan (*parsing*) data tersebut, maka data suhu udara siap untuk dikirim ke *database server*. Untuk melakukan pengiriman data suhu udara dari NetBeans ke *database server* yang telah dipersiapkan, *source code* dapat dilihat pada Gambar 4.7 dibawah ini:

```

public static void kirimData() throws Exception {

    String url =
"http://localhost/koneksi/input_data.php?id_waspmote=
"+id_waspmote+"&sensor="+sensor+"&urutan="+urutan+"&s
uhu="+suhu+"&batray="+batray;
    URL obj = new URL(url);
    HttpURLConnection con = (HttpURLConnection)
obj.openConnection();
}

```

```

        // optional default is GET
        con.setRequestMethod("GET");

        //add request header
        //con.setRequestProperty("User-Agent",
        USER_AGENT);

        int responseCode = con.getResponseCode();

        BufferedReader in = new BufferedReader(new
        InputStreamReader(con.getInputStream()));

        String inputLine;
        StringBuffer response = new StringBuffer();

        while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
        }
        in.close();

        //print result
        System.out.println(response.toString());
    }

    public static void kirimDataakeServer() throws
    Exception {

        String url = "http://smart-
        suhu.hughesspec.com/input_data.php?id waspmote="+id_w
        aspmote+"&sensor="+sensor+"&urutan="+urutan+"&suhu="+
        suhu+"&batray="+batray;
        URL obj = new URL(url);
        HttpURLConnection con = (HttpURLConnection)
        obj.openConnection();

        // optional default is GET
        con.setRequestMethod("GET");

        //add request header
        //con.setRequestProperty("User-Agent",
        USER_AGENT);

        int responseCode = con.getResponseCode();
        //System.out.println("\nSending 'GET' request
        to URL : " + url);
        //System.out.println("Response Code : " +
        responseCode);

        BufferedReader in = new BufferedReader(new
        InputStreamReader(con.getInputStream()));

        String inputLine;
        StringBuffer response = new StringBuffer();

        while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
        }
    }

```

```

    }
    in.close();

    //print result
    System.out.println(response.toString());
}

public static void main(String[] args) {
    // TODO code application logic here
    BacaArduino main = new BacaArduino();
    main.initialize();

    //memulai tugas/file java ini, akan di
    execute setiap X second
    main.start();
    System.out.println("Mulai");
}

//persiapan untuk kirim data setiap X second
long delay = 10 * 1000; // delay in milliseconds,
10 adalah 10 detik
LoopTask task = new LoopTask();
Timer timer = new Timer("kirim datanya");

public void start() {
    timer.cancel();
    timer = new Timer("kirim datanya");
    Date executionDate = new Date(); // no params
= now
    timer.scheduleAtFixedRate(task,
    executionDate, delay);
}

private String executeCommand(String command) {

    StringBuffer output = new StringBuffer();

    Process p;
    try {
        p =
Runtime.getRuntime().exec(command);
        p.waitFor();
        BufferedReader reader =
            new BufferedReader(new
InputStreamReader(p.getInputStream()));

        String line = "";

        while ((line = reader.readLine()) !=
null) {
            output.append(line + "\n");
        }
    }
}

```

```

    } catch (Exception e) {
        e.printStackTrace();
    }

    return output.toString();
}

```

**Gambar 4.7** Source Code Pengiriman Data ke Database  
(sumber: Data Olahan)

Ketika mengirim data ke *database*, maka harus diatur jeda waktu pengiriman data (*delay*). Dalam perancangan ini, pengiriman data tersebut diberikan jeda waktu selama 10 detik. *Source code* yang berfungsi untuk pemberian perintah masa jeda waktu pengiriman (*delay*) kedalam *database*, dan fungsi pengulangan dalam pengiriman data dapat dilihat pada Gambar 4.8 dibawah ini:

```

private class LoopTask extends TimerTask {
    @Override
    public void run() {
        System.out.println("-----");
        System.out.println("data akan dikirim
setiap 10 seconds.");
        try{
            kirimData();
            kirimDataakeServer();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

**Gambar 4.8** Source Code Jeda Waktu dan Looping Pengiriman Data  
(sumber: Data Olahan)

Setelah menambahkan *source code* tersebut pada NetBeans, maka dibuat dua *file php* yang berfungsi untuk melakukan koneksi ke *database* dan menyimpan data suhu ke *database*. *File* pertama diberi nama *connect.php* yang berfungsi dalam membuat koneksi dari *file java* pada NetBeans pada *database server*. *File* kedua diberi nama *input\_data.php* yang berfungsi sebagai perintah *input* data sensor suhu

dimana pada *file* ini data sensor suhu yang telah terbaca ditangkap pada *source code java*.

Kedua *file* tersebut memiliki *source code* sebagai berikut:

a. Koneksi *Database* (*connect.php*)

```
<?php

$db = new
mysqli("localhost","yhughes_hiddeni","hiddeni2123","
yhughes_smart_suhu");

if (get_magic_quotes_gpc()) {
    $process = array(&$_GET, &$_POST, &$_COOKIE,
&$_REQUEST);
    while (list($key, $val) = each($process)) {
        foreach ($val as $k => $v) {
            unset($process[$key][$k]);
            if (is_array($v)) {
                $process[$key][stripslashes($k)] =
$v;
                $process[] =
&$process[$key][stripslashes($k)];
            } else {
                $process[$key][stripslashes($k)] =
stripslashes($v);
            }
        }
    }
    unset($process);
}

if($db->connect_errno){
    echo $db->connect_error;
    die('Maaf, ada kesalahan teknis');
}
```

Gambar 4.9 Source Code Koneksi ke Database  
(sumber: Data Olahan)

b. Input Data Sensor dari Mini PC Database (*input\_data.php*)

```
<?php

date_default_timezone_set('Asia/Jakarta');

// array for JSON response
$response = array();

// check for required fields
if ($_GET != null) {
```

```

$id_waspmote = trim($_GET['id_waspmote']);
$sensor = trim($_GET['sensor']);
$urutan = trim($_GET['urutan']);
$suhu = trim($_GET['suhu']);
$batray = trim($_GET['batray']);
$time = trim($_GET['time']);
require 'connect.php';

if($insert = $db->query("INSERT INTO
tbl_suhu (id_secret, id_waspmote, sensor, urutan,
suhu, batray, time) VALUES ('', '$id_waspmote',
'$sensor', '$urutan', '$suhu', '$batray',
CURRENT_TIMESTAMP)")){

    // successfully inserted into database
    $response = "Data berhasil di input";
    echo $response;
} else {
    // failed to insert row
    $response = "Oops! Error input data";
    echo $response;
}
} else {
    // required field is missing
    $response = "Ada field yang kosong";
    echo $response;
}
?>

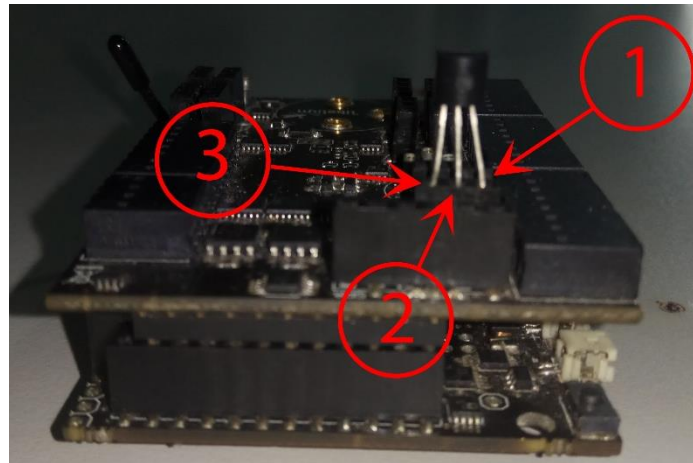
```

**Gambar 4.10** Source Code Input Data Sensor dari Mini PC ke Database  
(sumber: Data Olahan)

#### 4.2.3 Pengujian Sensor

Pengujian pada perangkat sensor melibatkan beberapa komponen. Pengujian ini menggunakan sebuah mikrokontroler Waspmote PRO yang dilengkapi dengan sebuah sensor suhu udara yang telah dihubungkan ke mikrokontroler. Mikrokontroler berfungsi sebagai pengatur sensor suhu untuk mengambil data berupa nilai suhu dengan memasukkan perintah berupa *source code*, dan juga berfungsi sebagai tempat memproses data-data yang masuk. Dalam penelitian ini, sensor suhu yang digunakan memiliki 3 buah pin, dimana pada pin – pin tersebut memiliki kegunaan tersendiri, seperti pada bagian pin pertama memiliki kegunaan sebagai tempat tegangan masuk, yaitu kisaran 4V sampai dengan 20V yang di perlukan untuk mengoperasikan sensor tersebut, agar bisa bekerja dengan baik, pin yang kedua memiliki kegunaan sebagai output atau hasil keluaran

dari sensor suhu yang berupa arus listrik, dan kemudian akan di conversikan oleh mikrokontroler kedalam bentuk angka atau arus digital, sedangkan kegunaan pin yang ketiga adalah GND atau biasa di sebut *ground* sebagai arus negatif. Dibawah ini adalah hasil dari pengujian sensor suhu yang akan digambarkan pada Gambar 4.11 dan Gambar 4.12.



**Gambar 4.11** Koneksi Dari Sensor ke Mikrokontroler  
(sumber: Data Olahan)

```

send error
=====
Current ASCII Frame:
Length: 56
Frame Type: 128
frame (HEX): 3C3D3E8003233338323534323631332373656E736F7230312332235443413A33302E3030234241543A35362354494D453A302D32D353123
frame (STR): <=>EQ#382542613#sensor01#2#TCA:30.00#BAT:56#TIME:0-2-51#
=====
send error
D#
Starting Sensing Environmental
Wait a minute.....
-----
=====
Current ASCII Frame:
Length: 56
Frame Type: 128
frame (HEX): 3C3D3E8003233338323534323631332373656E736F7230312330235443413A33302E3332234241543A35362354494D453A302D332D333523
frame (STR): <=>EQ#382542613#sensor01#0#TCA:30.32#BAT:56#TIME:0-3-35#
=====
send error
=====
Current ASCII Frame:
Length: 56
Frame Type: 128
frame (HEX): 3C3D3E8003233338323534323631332373656E736F7230312331235443413A33302E3332234241543A35362354494D453A302D332D333523
frame (STR): <=>EQ#382542613#sensor01#1#TCA:30.32#BAT:56#TIME:0-3-35#
=====
send error
=====
Current ASCII Frame:
Length: 56
Frame Type: 128
frame (HEX): 3C3D3E8003233338323534323631332373656E736F7230312332235443413A33302E3030234241543A35362354494D453A302D332D333523
frame (STR): <=>EQ#382542613#sensor01#2#TCA:30.00#BAT:53#TIME:0-3-35#
=====
send error
=====

```

**Gambar 4.12** Tampilan Nilai Suhu Pada Serial Wasp mote PRO  
(sumber: Data Olahan)



#### 4.2.4 Pengujian Pada Tempat Lobi

Setelah persiapan pengujian selesai, maka dilakukan pengujian langsung pada tempat lobi yang suhu udaranya mau diketahui. Pengujian menggunakan seluruh komponen perangkat lunak maupun perangkat keras yang dibutuhkan dalam penelitian ini. Adapun tujuan dari pengujian ini adalah untuk melihat fungsionalitas pemantauan suhu udara tersebut, dan ketika pengujian dapat dilihat baik tidaknya fungsi peralatan tersebut.

Ketika pengujian, dapat dilihat bahwa peralatan tersebut telah berjalan dengan baik sesuai dengan perancangan yang telah dilakukan. Sensor suhu dapat membaca nilai suhu udara pada tempat tersebut. Adapun gambaran pengujian dapat dilihat pada Gambar 4.13.



**Gambar 4.13** Pemantau Suhu Pada Lobi Menggunakan Sensor Suhu  
(sumber: Data Olahan)

#### 4.2.5 Hasil *Monitoring* Suhu Udara

Hasil pemantauan suhu udara yang dilakukan pada tempat lobi yang terdapat di politeknik negeri bengkalis, dengan melibatkan 2 tempat sebagai sampel diantaranya adalah tempat lobi yang terdapat pada gedung utama, dan yang satu laginya adalah tempat lobi yang terdapat pada gedung C,

pemantauan tersebut dilakukan pada tanggal 12 Agustus 2018. Pemantauan suhu udara dilakukan dengan melakukan pemantauan langsung dengan menggunakan sensor suhu. Masing-masing pemantauan dilakukan dengan durasi 5 menit. Hasil pemantauan dilakukan dengan menggunakan sensor suhu udara untuk memantau suhu udara pada tempat lobi yang telah ditentukan. Sensor suhu yang diletakkan pada tempat tertentu pada lobi tersebut, menampilkan suhu rata-rata yang dihitung dengan cara menjumlahkan nilai dari sensor dan hasilnya dibagi seberapa banyak nilai sensor, seperti pada rumus dibawah ini:

$$\frac{(\text{jumlah dari nilai suhu})}{(\text{banyak nilai suhu yang ada})}$$

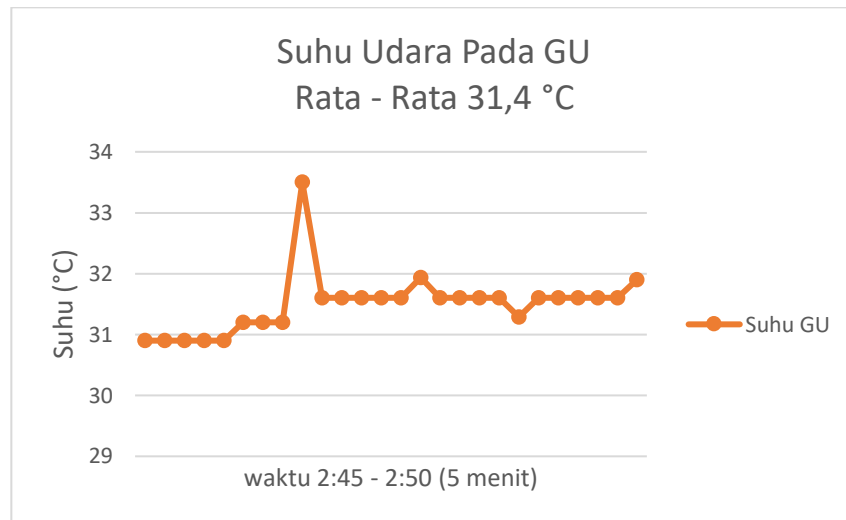
Nilai rata-rata dari hasil penghitungan tersebut bisa dijadikan sebagai nilai dari suhu udara pada tempat tersebut, pemantauan suhu udara dilakukan dengan menggunakan sensor suhu udara. Adapun nilai suhu udara pada tempat lobi yang diperiksa dengan menggunakan sensor suhu udara adalah:

**Tabel 4.1** Nilai Suhu Pada Lobi Menggunakan Sensor Suhu

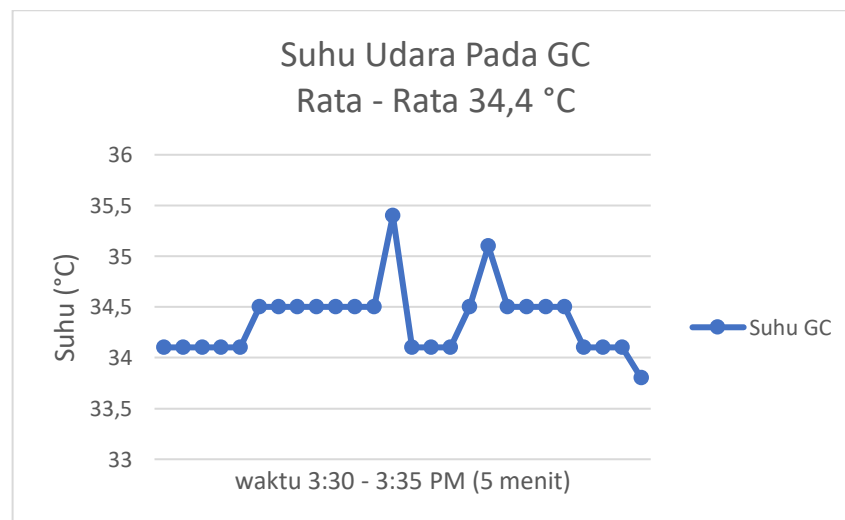
No.	Nama	Suhu Udara Dengan Sensor (°C)	Waktu Pengukuran
1	Gedung U	31,43	2:45 – 2:50 PM (5 menit)
2	Gedug C	34,41	3:30 – 3:35 PM (5 menit)

(sumber: Data Olahan)

Pada tabel diatas, dapat dilihat bahwa perbandingan yang sangat jauh antara suhu udara di lobi gedung utama sama di lobi gedung C. Perbedaan tersebut juga dapat dilihat dari grafik dibawah ini:



**Gambar 4.14** Grafik Pemeriksaan Suhu Udara di GU  
(sumber: Data Olahan)



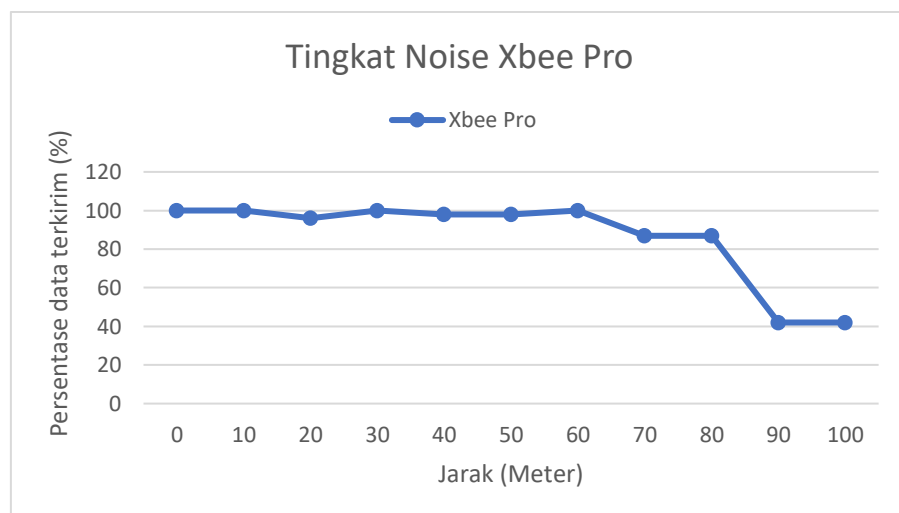
**Gambar 4.15** Grafik Pemeriksaan Suhu Udara di GC  
(sumber: Data Olahan)

#### 4.2.6 Hasil Pengujian Tingkat *Noise* Pada Xbee Pro

Pengujian tingkat *noise* pada xbee pro dilakukan pada area tanah lapang tanpa ada hambatan, yang dilakukan di politeknik negeri bengkalis pada tanggal 11 agustus 2018. Yang pada pengujian tersebut langsung dilakukan menggunakan 2 buah xbee, dimana xbee pertama bertugas sebagai *router* dan xbee yang kedua bertugas sebagai *coordinator*. Dimana xbee *router* mengirim data ke xbee *coordinator*, dan dari data yang terkirim tersebut kita bisa mengetahui pada jarak, apakah jarak antara xbee *router* ke *coordinator* semakin jauh akan berpengaruh terhadap tingkat *noise* data yang dikirim, hal tersebut dapat diketahui dengan menggunakan rumus dibawah ini:

$$\frac{\text{jumlah data terkirim}}{\text{jumlah data seharusnya}} \times 100$$

Dari penghitungan tersebut, maka ditemukan tingkat persentase *noise* dari pengiriman data melalui Xbee. Tingkat *noise* Xbee Pro dapat dilihat pada gambar dibawah ini:



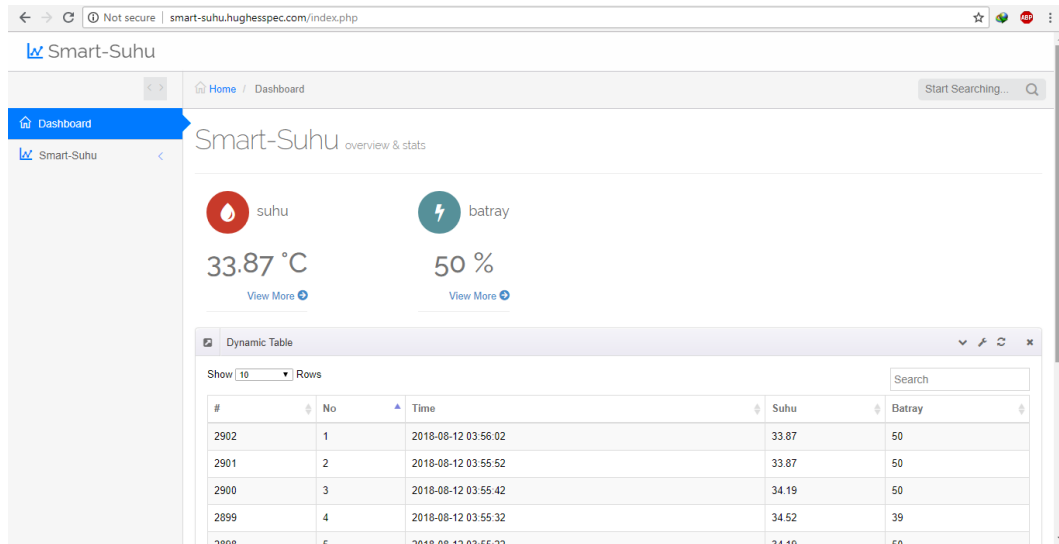
**Gambar 4.16** Grafik Tingkat *Noise* Pada Xbee Pro  
(sumber: Data Olahan)

#### 4.2.7 Tampilan *website* Pemantauan Suhu Udara

Halaman *website* digunakan untuk memonitoring suhu udara. Berikut adalah tampilan beberapa halaman yang terdapat pada *website*:

##### 1. Tampilan *Dashboard*

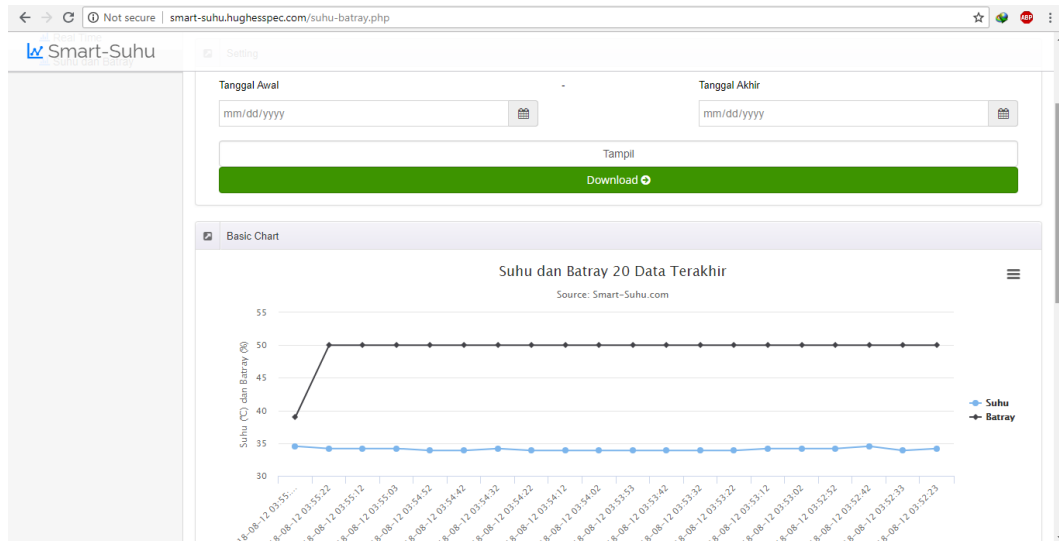
Tampilan *dashboard* dapat dilihat pada Gambar 4.17:



**Gambar 4.17** Tampilan *Dashboard*  
(sumber: Data Olahan)

## 2. Tampilan Grafik dan Rekap

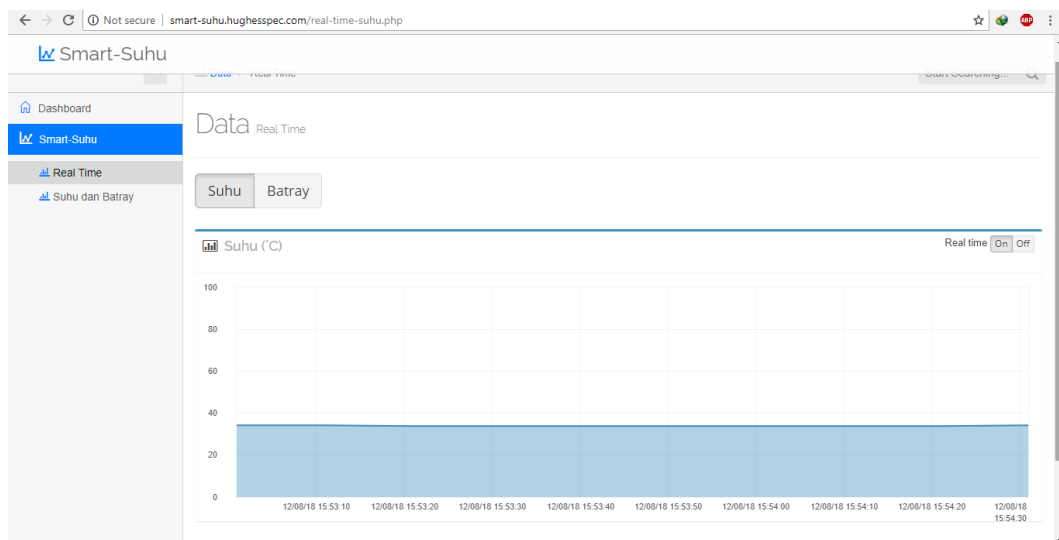
Tampilan grafik dan rekap dapat dilihat pada Gambar 4.18:



**Gambar 4.18** Tampilan Grafik dan Rekap  
(sumber: Data Olahan)

## 3. Tampilan Grafik *Real-Time*

Tampilan dari grafik *real-time* dapat dilihat pada Gambar 4.19:



**Gambar 4.19** Tampilan Grafik *Real-Time*  
(sumber: Data Olahan)

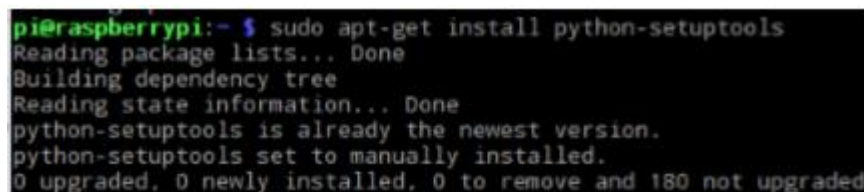
#### 4.2.7 Pengujian Media Sosial

Setelah beberapa pengujian sebelumnya sudah selesai, ini merupakan tahap pengujian yang terakhir dimana kondisi suhu udara dan setiap menerima inputan tersebut apabila fungsi if nya terpenuhi maka akan langsung mengirim update status ke Media Sosial secara otomatis berupa waktu dan suhu udara tersebut dengan membaca data dari Database. Update status ke Media Sosial secara otomatis tersebut menggunakan Twitterbot, dimana pembuatan Twitterbot ini dilakukan di Mini PC dengan menggunakan bahasa pemrograman Python. Sebelum membuat program di Python ada beberapa langkah yang harus dilewati dan mendapatkan token. Langkah-langkahnya adalah sebagai berikut :

a. Menginstall paket.

Sebelum menginstall paket-paket yang dibutuhkan harus update dan upgrade terlebih dahulu sehingga semua paket dalam Mini PC up to date. Kemudian jalankan perintah berikut pada terminal :

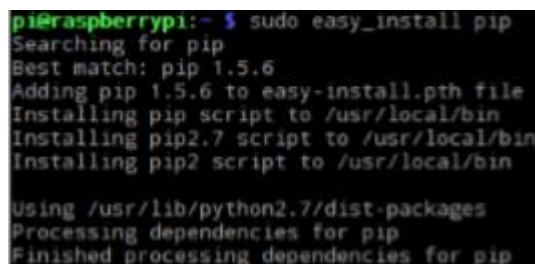
Sudo apt-get install Python-setuptools



```
pi@raspberrypi:~ $ sudo apt-get install python-setuptools
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-setuptools is already the newest version.
python-setuptools set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 180 not upgraded
```

**Gambar 4.20** sudo apt-get install Python-setuptools  
(sumber: Data Olahan)

Sudo easy\_install pip



```
pi@raspberrypi:~ $ sudo easy_install pip
Searching for pip
Best match: pip 1.5.6
Adding pip 1.5.6 to easy-install.pth file
Installing pip script to /usr/local/bin
Installing pip2.7 script to /usr/local/bin
Installing pip2 script to /usr/local/bin

Using /usr/lib/python2.7/dist-packages
Processing dependencies for pip
Finished processing dependencies for pip
```

**Gambar 4.21** sudo easy\_install pip  
(sumber: Data Olahan)

Sudo pip install twython

```
pi@raspberrypi:~$ sudo pip install twython
Requirement already satisfied (use --upgrade to upgrade):
twython in /usr/lib/python2.7/dist-packages
Cleaning up...
```

**Gambar 4.22** Sudo pip install twython  
(sumber: Data Olahan)

b. Mendapatkan Akses Token

Sebelum mendapatkan akses token dari Twitterbot yang akan dibuat, harus hosting website terlebih dahulu. Dimana pada TA ini alamat url yang sudah dihostingkan yaitu : <http://smart-suhu.hughesspec.com>. Setelah itu login dengan akun Twitter yang akan di buat twitterbot lewat <http://apps.twitter.com/app/new>.

c. Membuat Koding Program di Python

Sebelum membuat koding program ini, harus menginstall library terlebih dahulu untuk membaca data dari database ke Python yaitu Python-mysqldb yang dapat dilihat pada gambar berikut ini :

```
pi@raspberrypi:~$ sudo apt-get install python-mysqldb
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-mysqldb is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgra
ded.
```

**Gambar 4.23** sudo apt-get install python-mysqldb  
(sumber: Data Olahan)

Kemudian buat koding program twitterbot dengan menggunakan bahasa Python untuk mengirim update status ke twitter dan status ini akan secara otomatis terkirim. Program ini dibuat sekaligus membaca data dari database dan langsung mengirim update status ke Twitter dengan memasukan akses token yang sudah didapatkan sebelum dari akun Twitter



yang akan menerima tweet otomatis. File Python ini dibuat dengan nama MyTweeter.py dan disimpan dalam folder “/home/pi/ MyTweeter/” dan koding programnya dapat dilihat pada gambar 4.24 dibawah ini :

```
#!/usr/bin/env python
import sys
import os
import random
import MySQLdb
from twython import Twython

db =
MySQLdb.connect("localhost","root","hidenil23","db_suhu")

cursor = db.cursor()

CONSUMER_KEY = 'M2oZDbVpyEjtlpfCbD2EAaYKW'
CONSUMER_SECRET =
'mNSr2hvy1xzhPORKsMOcKpHnVuOZbGRcTGAtZW8eHKWIaQlOAW'
ACCESS_KEY = '1003492161545879552-
LDXeRKvswm2baclkG8weaHubRR91c4'
ACCESS_SECRET =
'zBcShKoXLLnrz09CcdELVO4ZqS0fEWGaps9kZo7dzLWJJ'
sql = "SELECT time,suhu,batray FROM tbl_suhu"

try :
    cursor.execute(sql)
    results = cursor.fetchall()
    for row in results:
        time = row [0]
        suhu = row [1]
        batray = row [2]

        tweet = "Pada tanggal %s \n PERINGATAN!!! \n  Suhu
: %s \n  Batray : %s" % \
        (time,suhu,batray)
        api =
Twython(CONSUMER_KEY,CONSUMER_SECRET,ACCESS_KEY,ACCESS_SECR
ET)
        api.update_status(status=tweet)

except :
    print "Error: unable to fetch data"

db.close()
```

**Gambar 4.24** Source Code MyTweeter.py  
(sumber: Data Olan)

Kemudian tambahkan coding pada program java yang telah dibuat sebelumnya untuk melakukan eksekusi command

“python/home/pi/MyTweeter/MyTweeter.py” dan diletakkan pada serialEvent atau tempat melakukan pemarsingan data, dimana data yang didapat berasal dari *port* serial atau tempat data tersebut masuk, dan pada *source* coding sebelumnya akan dilakukan perulangan karena inputan akan melakukan perulangan begitu juga mengeksekusi file MyTweeter.py ini setiap menerima inputan data sensor, dan apabila fungsi if terpenuhi maka akan mengirim status ke Media Sosial. Adapun penambahan koding programnya dapat dilihat pada Gambar 4.25 berikut ini:

```
if (Float.valueOf(suhu) >= 33.50 || Float.valueOf(batray) <= 10.00)
{
    System.out.println("suhu udara sudah melebihi dari batas normal");
    //post ke tw
    String command = "python /home/pi/MyTweeter/MyTweeter.py";
    String output = executeCommand(command);
} else {
    System.err.println("suhu udara masi dalam keadaan normal");
}
```

**Gambar 4.25** Source Code MyTweeter.py Pada Java  
(sumber: Data Olahan)

Setelah penambahan koding diatas selesai maka pengujian mengirim update status ke Media Sosial sudah dapat dilakukan, untuk melihat hasilnya dapat dilihat pada gambar berikut.

#### d. Status Media Sosial

##### Twitter



**Gambar 4.26** Status Di Twitter Sesuai Dengan Waktu Di Database  
(*sumber: Data Olahan*)

### 4.3 Pembahasan

Tahap ini merupakan Pembahasan dari hasil sistem yang telah dibangun dan diuji coba, serta kendala-kendala dan selama perancangan. Pada bagian ini juga membahas kelebihan dan kekurangan dari perangkat yang dibuat ini yang didapatkan dari dimulai dari perancangan hingga pengujian sistem pengukuran dan pemantauan suhu udara.

#### 4.3.1 Hasil Keseluruhan Sistem

Dari pengujian yang telah dilakukan, sistem pemantaun dan pengukuran suhu udara, yang dilakukan dengan sensor suhu udara ini dapat digunakan untuk *monitoring* dalam jarak jauh. Petugas/pengguna yang melakukan pemantauan dan pengukuran suhu udara, dapat meletakkan alat pemantauannya pada tempat lobi yang ingin diketahui suhu udaranya, dan nantinya dapat melihat nilai suhu udara tersebut pada halaman *website* secara *real-time*. Selama *monitoring* dilakukan, apabila suhu udara melewati dari batas yang telah di tentukan, maka akan muncul notifikasi ke media sosial, sebagai peringatan dini yang dikirimkan secara otomatis. Notifikasi tersebut berisi informasi mengenai tanggal, jam, menit, dan

berapa suhu udara yang melewati batas pada tempat tersebut. Suhu udara yang melewati batas memiliki rentang 33,5°C-35,5°C, pada suhu ini menyatakan bahwa kondisi lingkungan mulai cukup sulit diterima. Apabila suhu masi dibawah/normal maka tidak akan menampilkan notifikasi ke media sosial. Nilai suhu udara yang telah didapat sebelumnya juga dapat diunduh pada halaman *website* dalam bentuk file keluaran CSV, yang dapat di jadikan sebagai rekap nantinya.

#### 4.3.2 Kendala

Selama pembuatan sistem ini memiliki kendala ketika melakukan pengukuran dan pemantauan suhu udara pada tempat lobi. Kendala berupa ketergantungan sistem terhadap koneksi internet, sehingga apabila tidak terdapat koneksi internet maka nilai suhu udara tidak akan tampil pada halaman *website*.

#### 4.3.3 Kelebihan

Dari sistem yang telah dibuat, sistem pemantauan dan pengukuran suhu udara ini, dapat memberikan informasi nilai suhu udara secara *real-time* pada tempat lobi, dengan hanya melihat halaman *website* yang telah di buat sebelumnya. Petugas/pengguna juga dapat mengunduh nilai suhu udara yang telah didapat sebelumnya, dengan file keluaran berupa CSV yang dapat di gunakan sebagai rekap nantinya.

#### 4.3.4 Kekurangan

Kekurangan pada sistem ini adalah harus terkoneksi ke internet agar mendapatkan informasi nilai suhu udara pada tempat lobi. Apabila mini PC tidak terkoneksi dengan internet, maka mini PC tidak dapat mengirim data suhu udara ke *database server* dan pada halaman *website* tidak akan menampilkan nilai suhu udara pada tempat lobi.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Kesimpulan yang diperoleh dari pengerjaan Tugas Akhir ini adalah:

1. Sistem pengukuran dan pemantauan suhu udara secara *real-time* ini dapat memberikan informasi mengenai suhu udara pada tempat lobi secara *real-time*, dengan menggunakan sensor suhu yang telah di letakkan pada tempat lobi tersebut. Dan hasilnya dapat dilihat langsung dari halaman *website*.
2. Sistem pengukuran dan pemantauan suhu udara secara *real-time* sebagai wadah untuk peralatan berupa sensor suhu dan peralatan pendukung seperti Waspote PRO dan Raspberry Pi.
3. Agar suhu udara dapat terus tampil secara *real-time* pada halaman *website*, maka Raspberry Pi harus terkoneksi dengan internet.
4. Alat ini hidup dan berfungsi dengan menggunakan baterai dan *power bank* sebagai *power supply*.
5. Sistem ini digunakan untuk melihat nilai suhu udara juga dilengkapi dengan fitur unduh nilai suhu sebelumnya dengan file keluaran berupa CSV agar dapat digunakan sebagai rekap nantinya.
6. Sistem ini dapat mengirim notifikasi ke media sosial, sebagai peringatan dini jika suhu udara melebihi dari batas yang telah di tentukan.

#### **5.2 Saran**

Pada pengerjaan alat pengukur suhu udara ini tetap terdapat kekurangan, oleh karena itu diharapkan perlu adanya pengembangan kedepannya agar pengerjaan alat ini menjadi lebih baik dan tepat guna. Saran untuk pengerjaan kedepannya adalah dengan meningkatkan jumlah sensor yang digunakan sehingga pemantauan suhu udara pada lobi akan menjadi sangat efektif.

## DAFTAR PUSTAKA

- Anhar. (2010) Panduan Menguasai PHP & MySQL secara Otodidak, Jakarta: Mediakita.
- Budioko, T. (2016) Sistem Monitoring Suhu Jarak Jauh Berbasis Internet Of Things Menggunakan Protokol MQTT, STMIK Akakom Yogyakarta.
- Indriani, A., Johan, Witanto, Y., Hendra. (2014) Pemanfaatan Sensor Suhu LM 35 Berbasis Microcontroller ATmega 8535 pada Sistem Pengontrolan Temperatur Air Laut Skala Kecil, Fakultas Teknik Universitas Bengkulu, Jurnal Rekayasa Mesin, 5(2), 183-192.
- Kaaproject. (2018) *Kaa IoT Development Platform overview*, (Online), (<https://www.kaaproject.org/overview/>), diakses tanggal 22 Januari 2018.
- Kadir, A. (2017) Pemrograman *Arduino* dan *Processing*, Elex Media Komputindo.
- Kaur, G., Joshi, S., & Singh, G. (2014) Food Sustainability Using Wireless Sensors Networks: Waspote and Meshlium, International Journal of Computer Science and Information Technologies.
- Schmidt, M. (2012) Rangkaian Raspberry Pi, (Online), (<http://dwiagusti.ilearning.me/kkp/bab-2/>) diakses tanggal 22 Januari 2018.
- Sharon, D., wiliam, D.C. (1982) Jenis Sensor Dan Transduser, (Online), (<http://elektronika-dasar.web.id/jenis-sensor-dan-transduser/>), diakses tanggal 23 Januari 2018.
- Tim penyusun. (2017) Buku Panduan Proposal TA Mahasiswa Politeknik: Politeknik Negeri Bengkalis.

Waspote technical guide. (2018) documentation, (Online),  
(<http://www.libelium.com/development/waspote/documentation/waspote-technical-guide/>), diakses tanggal 25 Januari 2018.

Winarno, Utomo. (2010) Web Browser Pengertian web browser. Yogyakarta :  
Penerbit Elex Media Komputindo.

# **RANCANG BANGUN SISTEM PENGUKURAN DAN PEMANTAUAN SUHU UDARA SECARA REAL TIME BERBASIS IoT MENGGUNAKAN KAA PLATFORM**

Nama Mahasiswa : Al Mutaqhin Gustian Arhef

NIM : 6103151005

Dosen Pembimbing I : Nurul Fahmi, M.T

Dosen Pembimbing II : Eko Prayitno, M. Kom

## **Abstrak**

Lobi merupakan tempat yang penting, karena di lobi pengunjung maupun orang-orang lainnya melakukan aktifitas, oleh karena itu suhu udara lobi perlu diperhatikan, bahkan perlu di-*monitoring* supaya kenyamanan pengunjung dapat terjaga, sehingga pengunjung betah dan nyaman ketika melakukan aktifitas selama di lobi. Tapi masalahnya jika hanya mengandalkan alat pengukur temperatur biasa dan melakukan pemantauan secara manual itu sangat kurang efektif, oleh karena itu untuk menyelesaikan permasalahan yang terlampir diatas maka akan diusulkan suatu judul yang berjudul “RANCANG BANGUN SISTEM PENGUKURAN DAN PEMANTAUAN SUHU UDARA SECARA *REAL TIME* BERBASIS IoT MENGGUNAKAN *KAA PLATFORM*” pada judul diatas nantinya akan dibuat suatu alat yang dapat mengukur suhu udara secara *real time*, dengan bantuan sensor suhu dimana data dikirim ke *raspberry pi* yang kemudian dikirim lagi ke *kaa platform*, yang kemudian data tersebut dapat dilihat melalui tampilan *website*. Dan pada halama *website* tersebut dapat membuat file *output* dengan format CSV, yang dapat digunakan sebagai rekap nantinya. Apabila suhu udara terlalu tinggi atau melewati dari batas yang telah ditentukan, maka akan dikirimkan notifikasi ke twitter sebagai peringatan dini.

**Kata Kunci:** Suhu udara, *Website*, *Real time*, *Kaa platform*.



## **KATA PENGANTAR**

Puji dan syukur penulis haturkan atas kehadiran Tuhan Yang Maha Esa yang telah memberikan nikmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan Tugas Akhir (TA) yang berjudul “Rancang Bangun Sistem Pengukuran Dan Pemantauan Suhu Udara Secara Real Time Berbasis IoT Menggunakan Kaa Platform”. Shalawat serta salam semoga senantiasa tercurahkan buat Nabi junjungan alam Nabi Muhammad S.A.W beserta para keluarga, sahabat dan pengikutnya.

Dalam penulisan dan penyusunan laporan Tugas Akhir (TA) ini tidak terlepas dari bantuan, bimbingan serta dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Ir. Muhamad Milchan, M.T sebagai Direktur Utama Politeknik Negeri Bengkalis.
2. Bapak Danuri, S.T., M. Cs., selaku Ketua Jurusan Teknik Informatika Politeknik Negeri Bengkalis.
3. Ibu Sri Mawarni, M. Si., selaku Ketua Program Studi Diploma III Teknik Informatika Politeknik Negeri Bengkalis.
4. Bapak Nurul Fahmi, S.ST, M.T, sebagai dosen pembimbing I Tugas Akhir Politeknik Negeri Bengkalis.
5. Bapak Eko Prayitno, S. Kom., M. Kom, sebagai dosen pembimbing II Tugas Akhir Politeknik Negeri Bengkalis.
6. Seluruh Bapak/Ibu Dosen dan Laboran Jurusan Teknik Informatika Politeknik Negeri Bengkalis.

Penulis menyadari bahwa dalam pengerjaan Laporan Tugas Akhir ini masih banyak terdapat kekurangan. Untuk itu, kritik dan saran yang membangun sangat diharapkan demi kesempurnaan laporan ini. Semoga Laporan Tugas Akhir ini dapat memberikan manfaat bagi para pembaca dan semua pihak yang membutuhkan.

Akhir kata, penulis berharap Laporan Tugas Akhir yang dibuat dapat bermanfaat bagi siapapun yang membutuhkan referensi maupun ide untuk melakukan pengembangan dan penelitian. Terima kasih.

Bengkalis, 13 Agustus 2018

Penulis

# **DESIGN OF AIR TEMPERATURE MEASUREMENT AND MONITORING SYSTEM USING KAA PLATFORM IoT BASED**

Student Name : Al Mutaqhin Gustian Arhef

NIM : 6103151005

1<sup>st</sup> Supervisor : Nurul Fahmi, M.T

2<sup>nd</sup> Supervisor : Eko Prayitno, M. Kom

## **Abstract**

Lobby is an important place, because in the lobby visitors and other people do activities. Therefore the lobby air temperature needs to be considered, even need to be monitored so that the visitors comfort can be maintained, so that visitors feel comfortable and comfortable while doing activities in the lobby. But the problem is if only relying on ordinary temperature gauges and manually monitoring it is very ineffective, therefore to resolve that problems it will be proposed a research "REAL TIME AIR TEMPERATURE MEASUREMENT AND MONITORING SYSTEM USING KAA PLATFORM IoT BASED" in this research will be made a tool that can measure air temperature in real time, with the help of a temperature sensor where data is sent to raspberry pi which is sent back to the platform, which then can be seen through the website display. And on the website page, you can create an output file with CSV format, which can be used as a recap later. If the air temperature is too high or passes from a predetermined limit, a notification will be sent to twitter as an early warning.

**Keywords:** Air temperature, Website, Real time, Kaa platform.