

## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

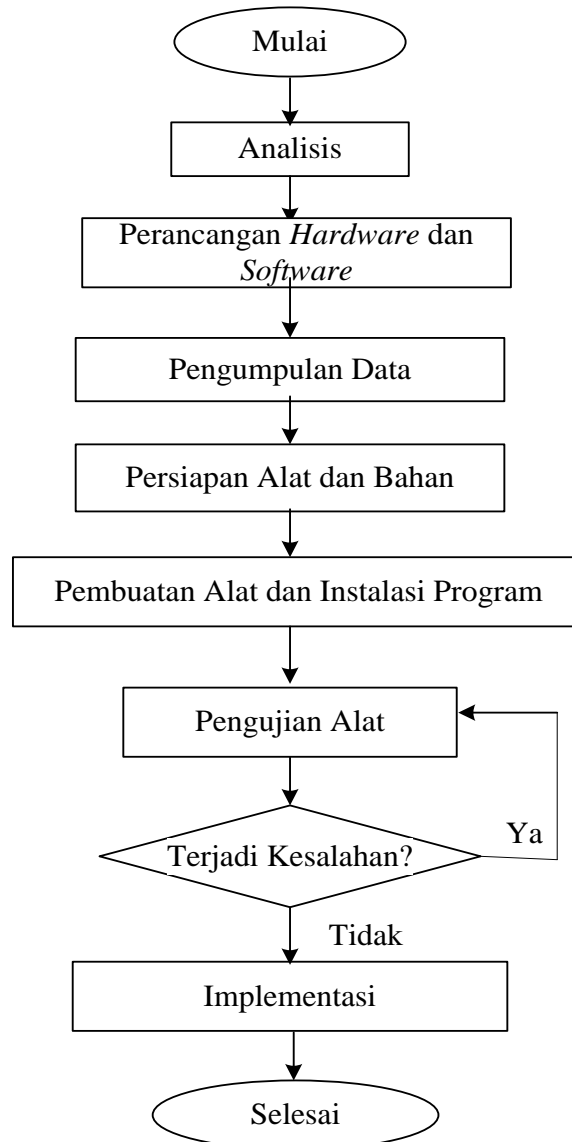
#### 3.1 Analisis

Pada perancangan sistem kerja diperlukan analisis perancangan pembuatan sistem akuisisi data tekanan udara, suhu dan kelembaban. Analisis dalam perancangan sistem ini meliputi kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*software*). Perancangan perangkat keras yang digunakan untuk penampil dan perekam kondisi tekanan udara, suhu dan kelembaban yang akan dilakukan oleh sensor BMP085 untuk mengukur tekanan udara, sensor DHT22 untuk mengukur suhu dan kelembaban dan mikropengendali Arduino Nano yang merupakan papan mikropengendali ATmega328 merupakan sebuah *chip* yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program didalamnya. Untuk kebutuhan *software* berupa perancangan program sesuai dengan perancangan alat seperti aplikasi Arduino I.D.E dan *Processing*.

#### 3.2 Perancangan Sistem

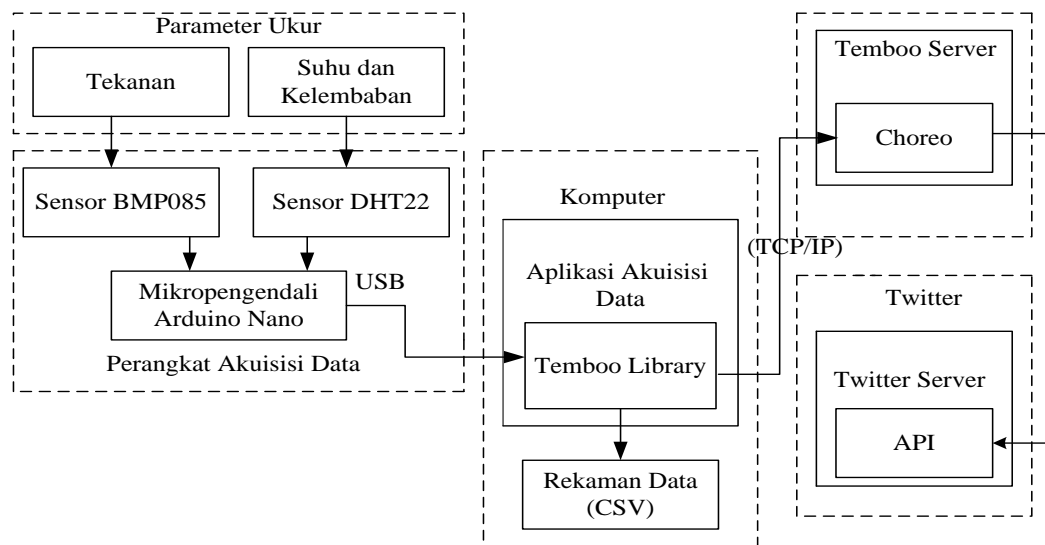
Dalam perancangan sistem aplikasi penampil dan perekam data tekanan udara, suhu dan kelembaban pada PC yang akan ditampilkan di *Twitter* dengan menggunakan *flowchart* yang merupakan metode yang dikerjakan oleh perseorangan dan individu dengan melalui proses secara bertahap dapat dilihat pada (Gambar 3.1). Keuntungan menggunakan metode ini yaitu merupakan alat (*tool*) dasar dan mudah dipergunakan serta sangat bermanfaat bagi suatu pihak perusahaan Manufaktur dalam mengidentifikasi terutama untuk menjelaskan setiap langkah dalam menjalankan proses operasionalnya, sebagai dokumentasi prosedur kerja dalam ISO, sebagai peta kerja untuk mencegah terjadi kehilangan arah, serta mempermudah pengambilan keputusan.

Berikut ini *flowchart* sistem kerja dari sistem akuisisi data tekanan udara, suhu dan kelembaban, seperti (Gambar 3.1).

Gambar 3.1 *Flowchart* Pengerjaan Tugas Akhir

Sistem kerja dapat dilihat parameter apa saja yang akan digunakan seperti komponen yang akan digunakan untuk dapat menunjang kebutuhan yang diperlukan dan merancang aplikasi melalui *software* dalam proses pengerjaan alat. Tahap selanjutnya, dilakukan analisis alat yang dirancang seperti komponen BMP085, DHT22 dan Arduino Nano. Berikutnya proses perancangan *hardware* dan *software* yang dibutuhkan seperti *hardware* adalah merancang komponen DHT22, BMP085 dan Arduino Nano. Selanjutnya membuat *sketch* pada masing-masing *software* seperti untuk Arduino I.D.E, *Processing*, *Temboo* dan *Twitter*. Mengumpulkan data untuk masing-masing *hardware* dan *software* diperoleh dari *datasheet* masing-masing komponen yang digunakan. Dilanjutkan dengan persiapan alat dan bahan seperti

sensor DHT22 dan BMP085 untuk masukan dan *software* sebagai pemrosesnya seperti *Processing*, *Temboo* dan satu media sosial yang digunakan sebagai keluaran seperti *Twitter*. Selanjutnya dilakukan pengujian alat melalui *software* yang digunakan, jika alat berjalan sesuai perencanaan awal maka akan dilanjutkan pengimplementasian pada suatu tempat untuk pengukuran tekanan udara, suhu dan kelembaban. Jika alat tidak berjalan sesuai perencanaan awal, maka dilakukan pengulangan pada tahap pengujian sampai mendapatkan data tekanan udara, suhu dan kelembaban yang akan ditampilkan di *Twitter*.



Gambar 3.2 Rancangan Sistem Akuisisi Data Tekanan Udara, Suhu dan Kelembaban Dengan Penayangan Hasil Pengukuran Melalui *Twitter*

Gambar di atas merupakan blok diagram perancangan sistem akuisisi data tekanan udara, suhu dan kelembaban dengan penayangan hasil pengukuran melalui *Twitter* seperti (Gambar 3.2).

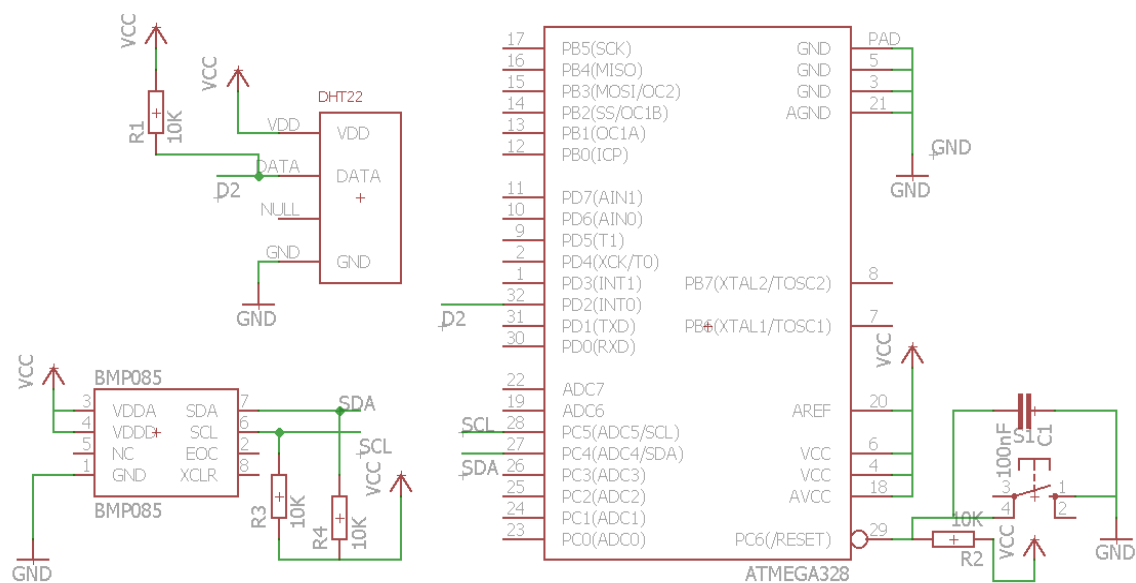
Blok diagram sistem Tugas Akhir ini mempunyai 5 bagian utama yaitu parameter ukur, perangkat akuisisi data, komputer, *Twitter* dan *Temboo* seperti (Gambar 3.2). Parameter ukur terdiri dari sensor BMP085 berfungsi mengukur tekanan udara dan sensor DHT22 untuk mengukur suhu dan kelembaban udara, kemudian hasil pengukuran akan dibaca dan diubah oleh mikropengendali berupa sinyal analog yang telah diubah menjadi sinyal digital oleh ADC (*Analog Digital Converter*) pada mikropengendali *Arduino Nano*. Kemudian aplikasi akuisisi data dalam komputer akan menerima sinyal keluaran dari mikropengendali *Arduino Nano V.3* melalui komunikasi data serial *USART (Universal Synchronous/Asynchronous Receiver/Transmitter)*. Pada aplikasi akuisisi data berperan untuk menampilkan dan

akan menyimpan data dalam format CSV (*Comma-Separated Value*). Data yang disimpan akan dikirim ke server Temboo menggunakan *library* Temboo melalui protokol TCP/IP. Selanjutnya data dikirimkan ke API melalui Choreo pada Temboo server dan ditampilkan dalam bentuk status teks pada *Twitter* dalam jangka waktu tertentu.

### 3.2.1 Perancangan Sistem Perangkat Keras (*Hardware*)

Pada perancangan perangkat keras terdapat pada perangkat akuisisi data yang terdiri dari Arduino Nano, sensor BMP025 untuk tekanan udara dan sensor DHT22 untuk pengukuran suhu dan kelembaban. Penggunaan Arduino Nano yang semakin memadai terutama pada papan pengembangan mikropengendali yang kecil semakin mendukung penggunaan *breadboard*, selanjutnya terdapat mikropengendali ATmega328 sebagai pemroses utamanya dan *chip* FTDI FT232L sebagai komunikasi serial.

Pada (Gambar 3.3) di bawah merupakan rancangan *hardware* melalui aplikasi Eagle 7.4.0 yang digunakan, meliputi komponen sensor DHT22, sensor BMP085 dan Arduino Nano.

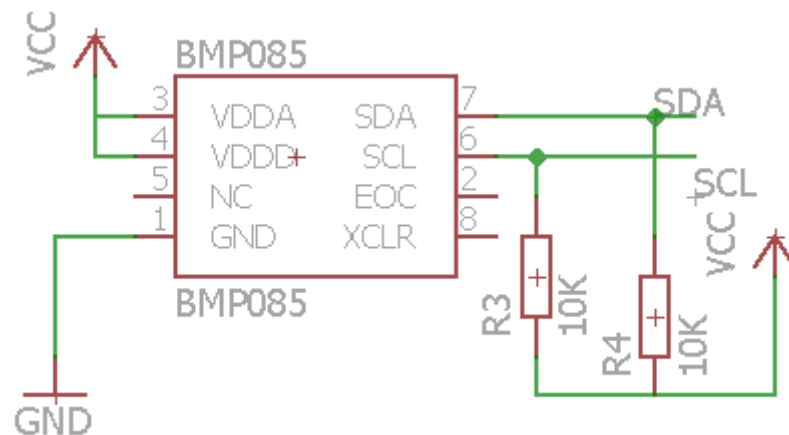


Gambar 3.3 Rancangan *Hardware* di Eagle

#### 3.2.1.1 Rangkaian Sensor BMP085

Sensor BMP085 digunakan untuk pengukuran tekanan udara dan juga mampu mengukur ketinggian diatas permukaan laut, penggunaan sensor ini dihubungkan ke Arduino Nano melalui pin VCC pada 3.3 V, pin *Ground* (GND) ke pin GND Arduino Nano. Pin SCL pada sensor BMP085

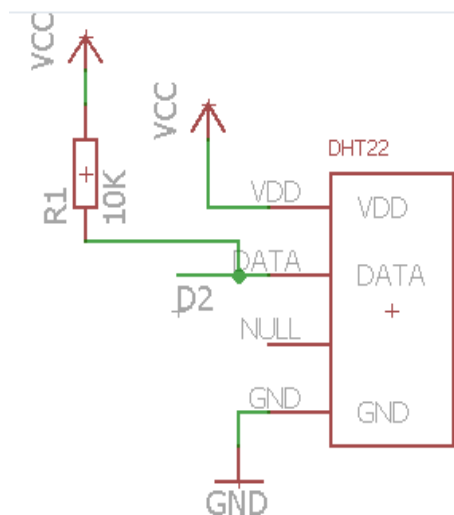
ke pin A5 pada Arduino Nano dan pin sensor SDA ke pin A4 Arduino Nano seperti (Gambar 3.4)



Gambar 3.4 Rangkaian Sensor BMP085 di Eagle

### 3.2.1.2 Rangkaian Sensor DHT22

Dan untuk sensor DHT22 merupakan multifungsi karena dapat digunakan untuk pengukuran suhu dan kelembaban udara, dengan sinyal digital yang dihasilkan dari sensor ini mudah dibaca oleh mikropengendali manapun. Untuk sensor ini pada pin 1 dihubungkan ke pin VCC pada Arduino, pin 2 dihubungkan dengan pin data pada sensor, pin 3 dihubungkan ke pin NC pada Arduino Nano dan pin 4 pada sensor dihubungkan ke pin *Ground* (GND) pada Arduino Nano seperti (Gambar 3.5)



Gambar 3.5 Rangkaian Sensor DHT22 di Eagle

Dalam perancangan rangkaian Tugas Akhir ini menggunakan *management* sederhana sehingga beresiko kegagalan proyek yang lebih rendah secara keseluruhan. Dan ketika dari awal semua kebutuhan sistem dapat

dipersiapkan secara utuh, benar dan dilakukan dengan tahap secara berurutan, maka sistem dapat berjalan dengan baik tanpa masalah.

### 3.2.1.3 Perancangan Komunikasi Serial

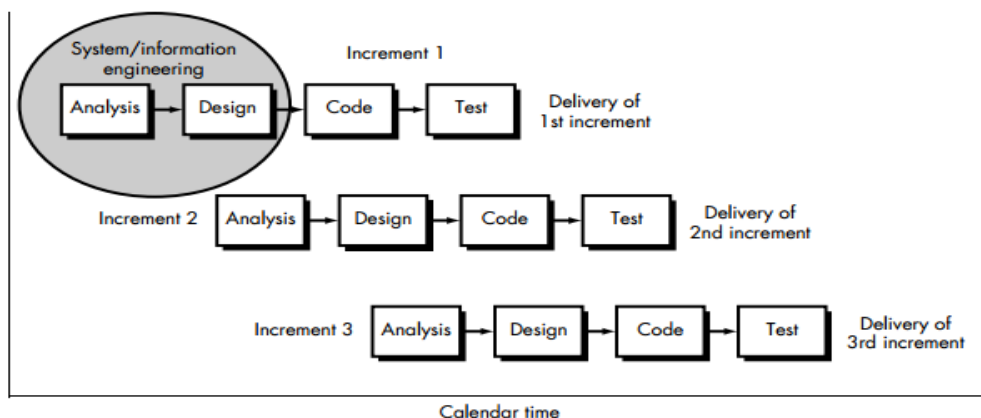
Komunikasi antara mikropengendali dan komputer memanfaatkan komunikasi serial berupa USB. Sementara komunikasi data menggunakan USART dengan konfigurasi 8 bit data, 1 bit stop dan *bit rate* 9600 bps. Menghubungkan perangkat rangkaian pendeteksi tekanan udara, suhu dan kelembaban dengan komputer, kemudian diproses melalui perangkat pemrograman *Processing*.

Data tekanan udara, suhu dan kelembaban yang ditampilkan pada serial monitor di *software* Arduino I.D.E contohnya seperti #2700 ; 2700 ; 100304. Untuk lebih jelasnya pola data yang ditampilkan adalah #data kelembaban ; suhu ; tekanan udara.

### 3.2.2 Perancangan Sistem Perangkat Lunak (*Software*)

Pada perancangan sistem perangkat lunak menggunakan model *Incremental* yang merupakan model pengembangan sistem pada *software engineering* berdasarkan *requirement software* yang dibagi menjadi beberapa bagian sehingga model pengembangannya secara bertahap.

Pada tahapan-tahapan *Incremental* di bawah ini dilakukan secara berurutan.



Gambar 3.6 Model *Incremental*

#### *Incremental 1*

Pada fase ini merupakan analisa dari kebutuhan sistem. Pada tahap analisa terhadap kebutuhan sistem yang dilakukan dengan melakukan pengumpulan data serta informasi yang dibutuhkan maka akan terbentuk

sebuah perencanaan aplikasi. Sehingga akan tercipta sebuah tahap-tahapan yang dapat dilakukan untuk membuat sebuah aplikasi penampil dan perekam data suhu dan kelembaban ruangan menggunakan perangkat pemrograman *Processing*.

#### 1. Kebutuhan sistem

Kebutuhan sistem yang dibutuhkan untuk dapat menjalankan aplikasi terdiri dari *hardware*, diantaranya :

- Sensor BMP085 dan Sensor DHT22
- Mikropengendali Arduino Nano V.3
- *Laptop* sebagai media untuk merancang dan membuat aplikasi yang sudah terinstal aplikasi *Processing* versi 2.2.1 Win (*Standard*). Adapun spesifikasi minimal *laptop* yang harus digunakan untuk menjalankan aplikasi Eclipse Indigo adalah sebagai berikut :
  - a) Windows Xp, Windows 7 (32/64 bit), Linux.
  - b) Ruang kosong pada *harddisk* sebesar 315 MB.

Pada Metode *Incremental I* parameter tekanan udara, suhu dan kelembaban yang diukur oleh sensor BMP085 untuk pengukuran suhu dan sensor DHT22 untuk sensor suhu dan kelembaban. Selanjutnya hasil keluaran dari sensor berupa sinyal analog yang akan diubah oleh ADC pada mikropengendali Arduino Nano. Sinyal keluaran dari ADC berupa bilangan bulat yang akan diubah menjadi *string* dan dikirimkan pada aplikasi akuisisi data pada PC. Sebelum mencapai tahap tes yang merupakan pengujian pada model ini, selanjutnya melangkah pada tahap *Incremental II*.

#### 2. *Incremental II*

Pada metode ini terdapat analisa kebutuhan user yang mana melakukan pengumpulan data dan informasi yang dibutuhkan dari perangkat mikropengendali Arduino Nano sehingga akan tercipta sebuah tahap-tahapan yang dapat dilakukan untuk membuat aplikasi penampil dan perekam data tekanan udara, suhu dan kelembaban menggunakan perangkat yang dibutuhkan, seperti :

##### 1. *Software*

- Arduino I.D.E

- *Processing* versi 2.2.1 Win (*Standard*)

## 2. *Hardware*

- Laptop sebagai media untuk merancang dan membuat aplikasi dimana *laptop* tersebut harus sudah terinstal *software* *Arduino I.D.E* dan aplikasi *Processing* versi 2.2.1 Win (*Standard*). Adapun spesifikasi minimal laptop yang harus digunakan untuk menjalankan aplikasi *Eclipse Indigo* adalah sebagai berikut :
  - a. Windows Xp, Windows 7 (32/64 bit), Linux.
  - b. Ruang kosong pada *harddisk* sebesar 315 MB.

Pada metode *Incremental II* aplikasi akuisisi data dibangun pada laptop menggunakan bahasa pemrograman *Processing* yang akan menerima data dari mikropengendali *Arduino Nano* melalui kabel USB dan akan menampilkan dan menyimpan data dalam bentuk format CSV. Sebelum mencapai tahap tes pada metode ini selanjutnya melangkah pada metode *Incremental III*.

## 3. *Incremental III*

Pada metode ini dibutuhkan adanya analisa untuk *user* dengan melakukan pengumpulan data dan informasi yang akan dibutuhkan sehingga mampu menghasilkan data yang dapat ditampilkan melalui *Twitter*.

### a. *Kebutuhan user*

*Kebutuhan user* merupakan pengguna aplikasi penampil dan perekam data tekanan udara, suhu dan kelembaban menggunakan aplikasi *Temboo* dan *Twitter* sebagai *keluaran*.

### b. *Kebutuhan sistem*

*Kebutuhan sistem* yang dibutuhkan berupa :

- *Temboo* merupakan sebuah layanan *cloud* menyediakan modul-modul yang berfungsi untuk menjembatani bahasa pemrograman terhadap API yang akan diakses.
- *Twitter* sebagai salah satu situs media sosial menyediakan layanan web dalam bentuk status teks 140 karakter.

Selanjutnya pada metode *design* sistem ini yang dilakukan melalui data yang dikirimkan ke server *Temboo* menggunakan pustaka *library*. *Temboo* melalui *protocol* TCP/IP. Data pada *Temboo* selanjutnya dihubungkan ke



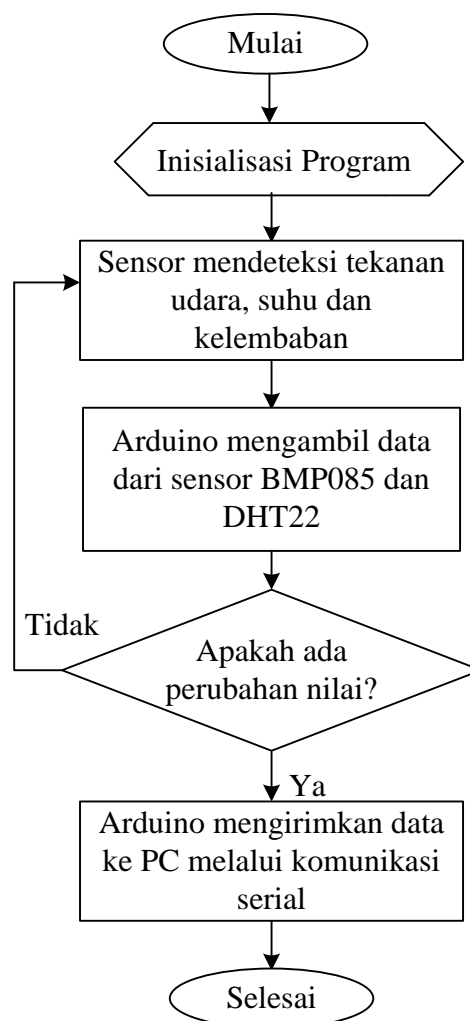
*Twitter* API melalui Choreo pada aplikasi Temboo. Dan selanjutnya *Twitter* akan menampilkan dalam bentuk teks sebagai keluarannya.

Pada tahap pembuatan program ini dilakukan translasi dari *design* sistem yang telah dirancang dan dilengkapi dengan informasi dan data yang telah tersedia menjadi kode-kode komputer.

Tahapan tes ini dilakukan setelah aplikasi selesai. Tahap ini dilakukan pengujian sistem apakah sistem telah memenuhi persyaratan yang diharapkan, dan pengujian dilakukan disetiap unit perangkat lunak yang dilakukan ditingkat kelas dan unit pengujian minimal. Pengujian yang dilakukan untuk mengetahui nilai pengukuran tekanan udara, suhu dan kelembaban, tampilan dan fungsi dari aplikasi yang telah dibuat.

### 3.2.2.1 PERANCANGAN PROGRAM ARDUINO

Berikut merupakan *flowchart* sistem sensor DHT22 dan BMP085 di dalam mikropengendali Arduino Nano.



Gambar 3.7 *Flowchart* Kerja Program Arduino

*Flowchart* di atas terdapat unit pengirim melalui komunikasi USART yang terhubung dengan komponen sensor BMP085, sensor DHT22 dan Arduino Nano. Pada awal program dilakukan proses inisialisasi program seluruh bagian dari sistem. Kemudian sensor akan mendeteksi adanya tekanan udara, suhu dan kelembaban melalui sensor BMP085 dan DHT22. Pada unit pengirim data yang diperoleh dari masukan sensor BMP085 dan sensor DHT22 diproses oleh mikropengendali Arduino Nano. Pada unit penerima data yang diterima diproses oleh mikropengendali dan disimpan oleh komputer. Bila tidak ada perubahan nilai pengukuran atau pengukuran tidak sesuai dengan program maka akan kembali mendeteksi pengukuran sensor BMP085 dan DHT22. Kemudian data tekanan udara, suhu dan kelembaban ditampilkan melalui serial monitor apabila terjadi perubahan nilai pengukuran.

Dari aplikasi Arduino dilakukan proses pembacaan data untuk sensor BMP085 sebagai pengukur tekanan udara dan sensor DHT22 sebagai pengukur suhu dan kelembaban, sehingga nilai pengukuran tekanan udara, suhu dan kelembaban dapat diukur dan ditampilkan melalui *software* Arduino melalui komunikasi serial.

Berikut ini merupakan *listing* program di Arduino I.D.E melalui USB yang terhubung dengan perangkat sensor BMP085, DHT22 dan Arduino Nano. Dimulai dengan program yang digunakan untuk memanggil *library* untuk sensor DHT22 dan BMP085 yaitu:

```
#include <DHT.h> #include <Wire.h>
#include<Adafruit_BMP085.h> #include "DHT.h"
```

Selanjutnya untuk inisialisasi program, untuk konfigurasi *baudrate* serial 9600 dengan kecepatan mengirim data 9600 persekon, untuk mengirim data ASCII + CR, LF (kode enter), melakukan konfigurasi awal atau *setup* yang dijalankan sekali saja setelah Arduino mulai berjalan atau setelah tombol reset ditekan dan untuk mengeset kecepatan transmisi data menggunakan program :

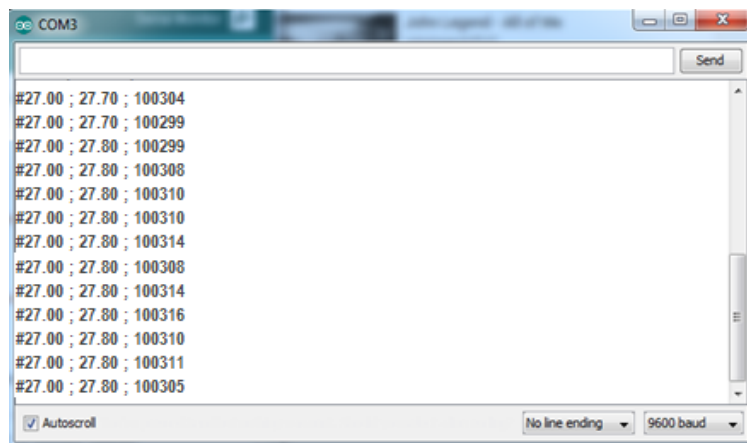
```
void setup() {
  Serial.begin(9600);
  if (!bmp.begin()) {
```

```
Serial.println("Could not find a valid BMP085  
sensor, check wiring!");  
while (1) {}  
Serial.begin(9600);  
Serial.println("Kelembaban;Suhu;Tekanan");  
dht.begin();
```

Berikut program digunakan untuk untuk mengirim data ASCII + CR, LF (kode enter) dan mengeksekusi program untuk keluaran tekanan udara, suhu dan kelembaban yang diukur akan dijalankan berulang-ulang untuk data tekanan udara, suhu dan kelembaban :

```
void loop() {  
Serial.print(dht.readTemperature());  
Serial.print(" ; ");  
Serial.print(bmp.readTemperature());  
Serial.print(" ; ");  
Serial.println(bmp.readPressure());
```

Dan program pemberian jeda selama 0,5 detik adalah `Delay(500);`



Gambar 3.8 Hasil Data Yang Direkam di Arduino

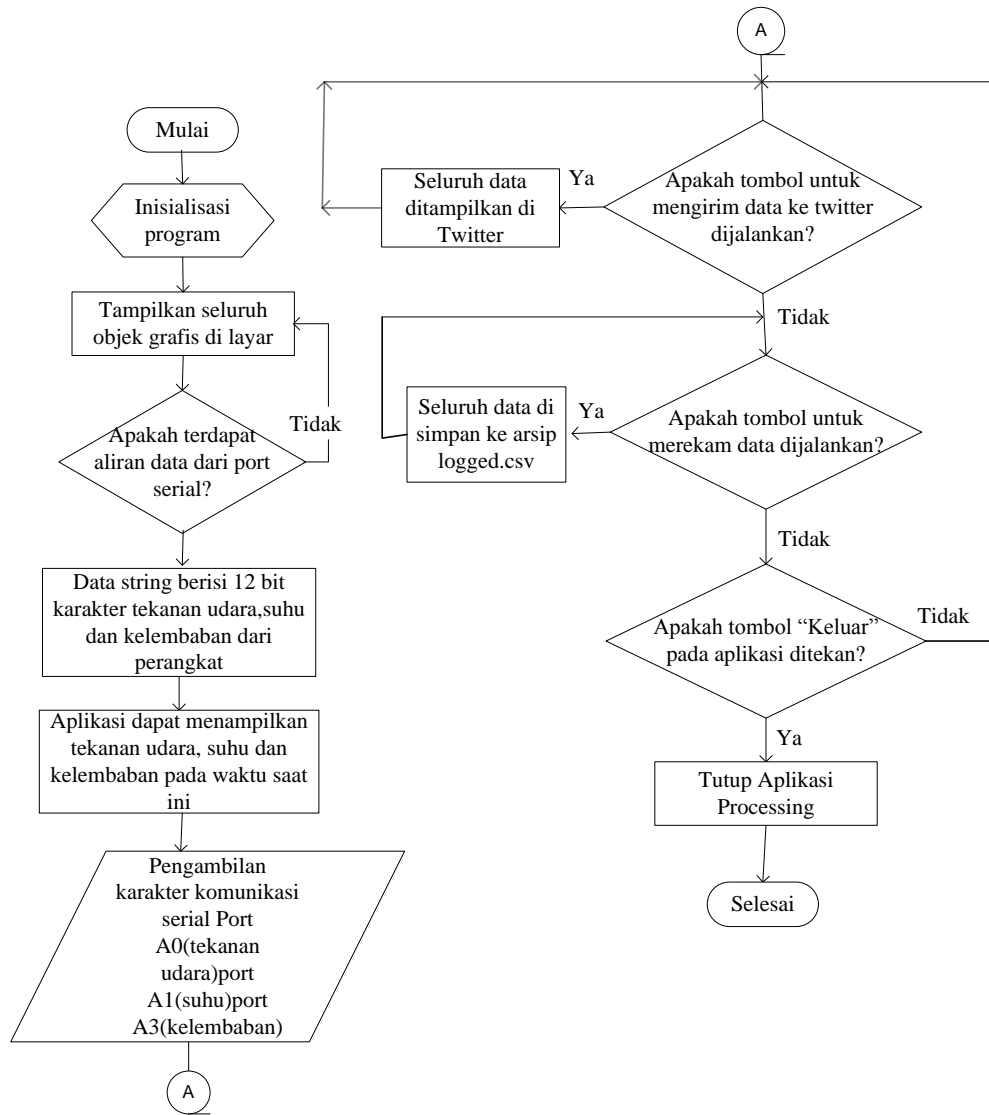
Sebagai contoh keluaran format perekaman data yaitu *file* yang berekstensi CSV didalamnya terdiri dari parameter yang direkam didalam aplikasi antar muka pengguna tersebut adalah sebagaimana ditampilkan dalam (Gambar 3.8). Parameter yang direkam dipisahkan dengan simbol titik koma (;) yang terdiri dari *Pressure*, *Temperature*, *Humidity* secara terus menerus tiap detik.

### 3.2.2.2 PERANCANGAN APLIKASI PROCESSING.

Perancangan tampilan program untuk data tekanan udara, suhu dan kelembaban suatu daerah yang berbasis GUI menggunakan perangkat pemrograman *Processing* versi 2.2.1 Win (*Standard*) dengan menggunakan komunikasi serial USART menggunakan mini USB data tekanan udara, suhu dan kelembaban dikirim ke PC dan diproses dengan menggunakan aplikasi *Processing* versi 2.2.1 Win (*Standard*). Alasan menggunakan *Processing* versi 2.2.1 Win (*Standard*) karena *file* yang terinstal lebih kecil. Sehingga lebih ringan ketika dijalankan. Pada versi ini termasuk salah satu kategori *stable releases*, artinya adalah pada versi ini akan lebih stabil. Sudah mencukupi untuk membuat aplikasi yang berinteraksi dengan *hardware* sensor BMP085 dan sensor DHT22. Selain itu tidak ditemukan permasalahan untuk mengekspor aplikasi ke Windows, Linux dan Mac.

Dimulai dengan proses inialisasi program dengan memanggil modul komunikasi serial, pengaturan *delay* dan objek komunikasi serial. Selanjutnya penentuan besar ukuran objek grafis di layar untuk program perancangan GUI adalah [560, 530] *pixel* dan melakukan pengecekan *port serial* yang aktif pada komputer. Ketika *port* serial sudah sesuai dengan komputer pengguna dari sensor akan mengirim karakter data yang berupa data tekanan udara, suhu dan kelembaban.

Dari *Processing* dilakukan proses pembacaan data dan pemisahan karakter atau pengambilan karakter untuk tekanan udara, suhu dan kelembaban. Sehingga nilai tekanan udara, suhu dan kelembaban yang ada pada perangkat sensor dapat ditampilkan melalui proses komunikasi serial. Selanjutnya data akan dikirim ke *Twitter* dengan menyimpan data diarsip *logged.csv*, apabila data tidak mengirim ke *Twitter* maka akan diulang di pengambilan karakter data. Untuk proses perancangan aplikasi keseluruhan ditunjukkan pada *flowchart* di bawah pada (Gambar 3.9) *flowchart* perancangan komunikasi serial.



Gambar 3.9 Flowchart Perancangan Aplikasi Keseluruhan

Pada perancangan sistem aplikasi penampil data tekanan udara, suhu dan kelembaban pada PC dirancang melalui aplikasi *Processing* untuk mengetahui alur kerja keseluruhan sampai menampilkan di *Twitter* dari algoritma untuk aplikasi *Processing* (Gambar 3.9) dimulai dari inisialisasi seluruh bagian program. Objek grafis akan ditampilkan di layar sesuai dengan program yang telah dijalankan. Selanjutnya apabila terdapat aliran data dari *port* serial maka telah dapat menampilkan tekanan udara, suhu dan kelembaban pada waktu saat ini atau sesuai dengan waktu yang ditampilkan di laptop dan jika tidak, maka akan dikembalikan ke perintah menampilkan seluruh objek grafis. Selanjutnya aplikasi akan dapat menampilkan data tekanan udara, suhu dan kelembaban. Untuk tombol yang berfungsi untuk

mengirim data di *Twitter* dijalankan dan data akan disimpan ke arsip *logged.csv*, jika tidak maka akan kembali hanya pada penampilan data tekanan udara, suhu dan kelembaban dan dilanjutkan dengan tombol keluar yang berarti arsip *logged.csv* keluar dari aplikasi *Processing*.

Berikut ini terdapat *listing* program dalam *Processing* untuk komunikasi serial agar dapat terhubung dengan perangkat sensor BMP085 dan sensor DHT22. Sehingga data tekanan udara, suhu dan kelembaban dapat ditampilkan pada aplikasi antar muka. Dimulai dengan proses inisialisasi program dengan mengimpor *library* atau berkas pustaka yang diperlukan dari Java melalui perintah dibawah ini, yang berfungsi untuk memanggil modul komunikasi serial dan Java.

```
Import Processing.serial.*;
```

Kedua baris program yang berkaitan untuk pengaturan *delay* untuk komunikasi serial. Program Variabel *lastTime* untuk menyimpan catatan waktu sebelumnya, Variable *DelayTime* untuk menentukan panjang waktu *delay* sebesar 1000 ms atau 1 detik.

```
Int lastTime;  
Int delayTime=1000;
```

Selanjutnya kedua definisi di bawah ini berkaitan dengan objek komunikasi serial. Data merupakan variabel yang bertipe objek serial pada *serial data*. Sedangkan *dataString* merupakan variabel yang bertipe *string* dengan fungsi untuk menyimpan data yang diterima dari komunikasi.

```
Serial data;  
String dataString;
```

Untuk variabel yang berfungsi untuk menyimpan hasil pemilihan (*parsing*) *string* data yang diperoleh dari Arduino. Variabel *A0* berfungsi untuk menyimpan hasil pembacaan dari *port* analog *A0*. Begitu juga dengan variable *A1* dan *A2* melakukan penyimpanan hasil pembacaan dari *port* analog *A1* dan *A2*. Kedua variable tersebut diberi nilai awal “—“ saat inisialisasi.

```
String A0="—";  
String A1="—";  
String A2="—";
```

`Void setup ()` *listing* program di bawah ini berhubungan dengan komunikasi serial yaitu `println(Serial.list())` berfungsi untuk menampilkan seluruh *port* serial yang aktif dalam komputer. Perintah tersebut

hanya melakukan pengecekan *port* "COM3" ini diketahui setelah melihat tampilan dari basis program sebelumnya yang melakukan pengecekan *port* serial yang digunakan.

```
Void setup()  
{  
Size(500,400);  
Smooth();  
tulisan1=loadFont("Andalus-48.vlw");
```

#### **3.2.2.2.1 Cara Kerja Aplikasi di Processing**

Aplikasi *Processing* akan berjalan untuk menampilkan data tekanan udara, suhu dan kelembaban dibagian kondisi terukur ketika keluaran data dari sensor DHT22 dan BMP085 sudah terbaca di *software* Arduino I.D.E. Perekaman data akan dilakukan oleh ikon "REKAM DATA", saat melakukan perekaman data dalam pengujian proses penyimpanan data akan menghasilkan *file logged.csv*. Untuk pengiriman data tekanan udara, suhu dan kelembaban ke *Twitter* maka hanya dengan menekan ikon yang bernama "KIRIM TWITTER" dibagian operasi maka data tekanan udara, suhu dan kelembaban akan dikirimkan ke *Twitter* berdasarkan format penampilan status yang telah diatur. Untuk pemunculan data hari/tanggal dan waktu disesuaikan dengan kondisi yang ada di PC.

#### **3.2.2.2.2 Perancangan Tampilan Data Processing**

Untuk perancangan program aplikasi tampilan program di *Processing* terdiri dari dua bagian yaitu kondisi terukur dan operasi. Bagian kondisi terukur akan ditampilkan data tekanan udara, suhu dan kelembaban yang berasal dari sensor BMP085 dan sensor DHT22. Rangkaian dari BMP085 dan DHT22 dihubungkan ke aplikasi melalui komunikasi serial USART dengan *mini* USB. Selain data tekanan udara, suhu dan kelembaban juga menampilkan data waktu untuk hari, tanggal dan tahun pada saat itu. Data tersebut dari komputer (PC), sehingga pada aplikasi akan menampilkan sesuai dengan format yang ada pada komputer begitu juga dengan tampilan waktu untuk jam, menit dan detik yang berasal dari sistem komputer.

Tampilan program pada bagian perekam data oleh perangkat program *Processing*, untuk data tekanan udara, suhu dan kelembaban akan disimpan dalam komputer sebagai bentuk perekam data. Terdapat tombol keluar untuk

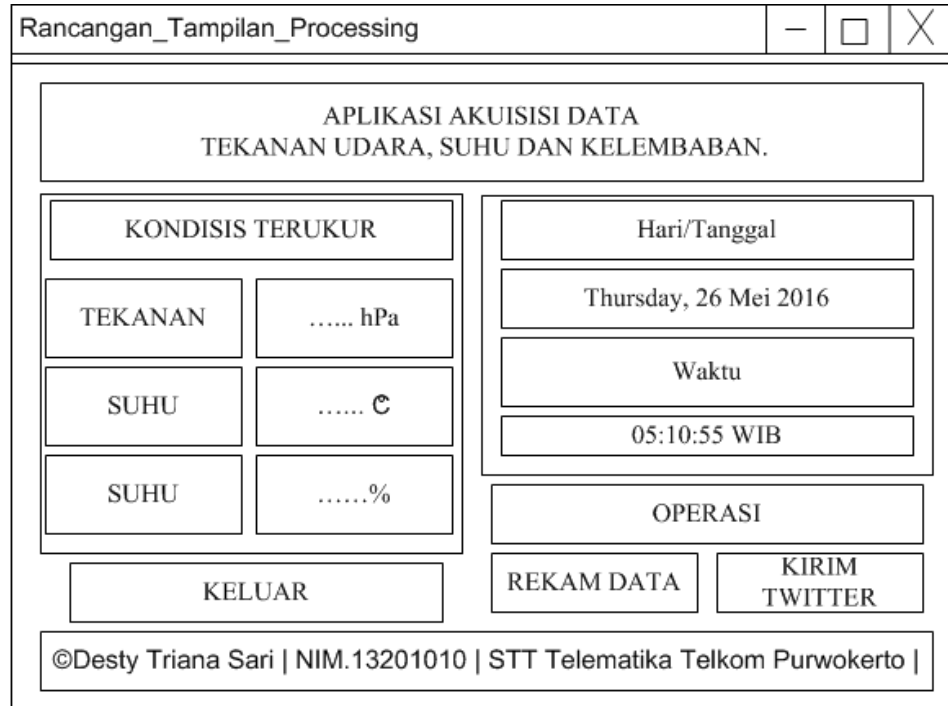
keluar dari program aplikasi, juga berfungsi untuk menyimpan hasil perekaman data dan kelembaban dalam komputer.

Perancangan aplikasi antarmuka diawali dengan membuat objek-objek grafis, dapat dilihat pada (Tabel 3.1). Sebagai contoh untuk tombol “EXIT” menggunakan *listing* program sebagai berikut :

```
cmdButton(295, 430, 235, 55);  
//exit  
fill(128);  
strokeWeight(2);  
stroke(#A20A3A);  
rect(5, 290, 245, 80);  
//tulisan exit  
textFont(tulisan1);  
fill(255);  
textSize(20);  
text("EXIT", 110, 335);
```

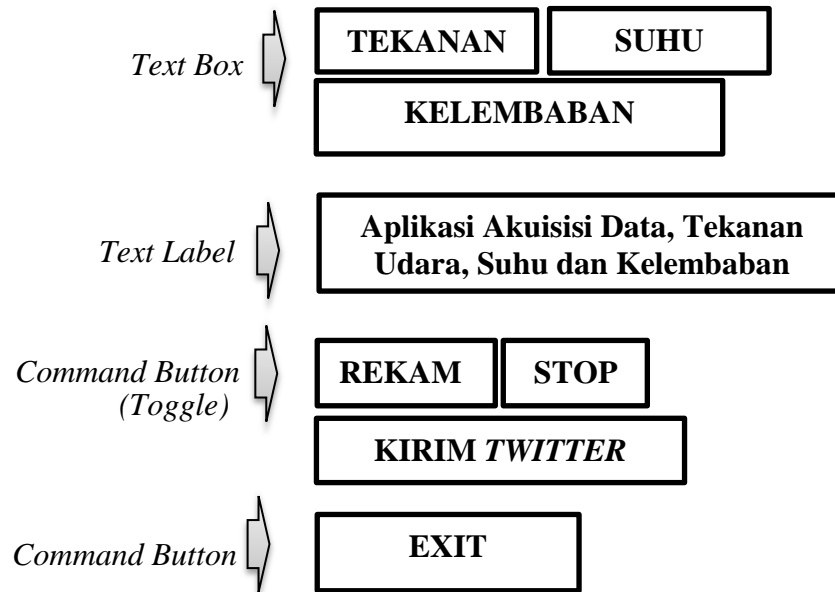
Pada (Gambar 3.10) adalah rancangan tampilan dari aplikasi *Processing*, judul dari tampilan aplikasi adalah “Aplikasi Akuisisi Data Tekanan Udara, Suhu dan Kelembaban”. Parameter untuk kondisi terukur ada tiga yaitu besar tekanan udara, suhu dan kelembaban disertakan satuan untuk masing-masing parameter ukurnya. Untuk operasi pengukuran akan ditampilkan pengiriman data tekanan udara, suhu dan kelembaban ke *Twitter*. Dan ditampilkan data hari, tanggal dan waktu jalannya operasi penampilan data. Untuk tombol *exit* merupakan perancangan *commandButton*. Dan dimunculkan juga *list* keterangan profil pembuat dari aplikasi *Processing* yang ditampilkan secara berjalan. Seperti ©Desty Triana Sari | NIM. 13201010 | STT Telematika Telkom Purwokerto | Telp. 089639516829 | E-mail 13201010@st3telkom.ac.id.





Gambar 3.10 Rancangan Tampilan Aplikasi Melalui *Processing*

Berikut merupakan bentuk tampilan dari objek-objek grafis dalam rancangan aplikasi penampil data tekanan udara, suhu dan kelembaban menggunakan perangkat pemrograman *Processing*. Dirancang dengan berbasis GUI, objek-objek tersebut ditampilkan pada (Gambar 3.11)



Gambar 3.11 Objek-objek Dalam Perancangan Aplikasi

Tabel 3.1 Objek-objek Pada Aplikasi Antarmuka Pengguna.

No	Objek	Penjelasan Objek
1	<i>Text Box</i>	Objek grafis terdiri dari kotak dan tulisan dibentuk dengan <i>rectangle</i> dan <i>cmdLabel</i> pada <i>Processing</i> . Seperti pada objek kondisi terukur, perekam data, suhu, dan kelembaban dalam sketsa perancangan aplikasi.
2	<i>Text Label</i>	<i>Text Label</i> berfungsi untuk memberikan tulisan berupa <i>text</i> pada aplikasi yang dirancang. Pembentukan objek ini pada <i>Processing</i> adalah label.
3	<i>Command Button (Toggle)</i>	<i>Command Button (Toggle)</i> adalah objek grafis untuk tombol perintah yang melibatkan intraksi pengguna. Contohnya pada satu tombol objek yang terdiri dari dua perintah rekam dan hentikan pada sketsa perancangan <i>Processing</i> . Pembentuk objek tersebut terdiri dari <i>cmdButton</i> dan <i>cmdLabel</i> dengan pemproses aksi menggunakan <i>mousePressed</i> .
4	<i>Command Button</i>	<i>Command Button</i> merupakan tombol perintah pada aplikasi, pembentuk objek berupa <i>cmdButton</i> dan label. Untuk memproses aksi menggunakan <i>mousePressed</i> pada <i>Processing</i> . Contoh pada sketsa untuk tombol perintah keluar yang berfungsi untuk menyimpan data dan menutup program.

Pada gambar sketsa (Gambar 3.10) adalah rancangan bentuk keluaran yang diharapkan dapat menampilkan dan merekam hasil data tekanan udara, suhu dan kelembaban melalui perangkat pemrograman *Processing*. Rangkaian sensor BMP085 dan DHT22 dihubungkan ke aplikasi melalui komunikasi serial USART menggunakan mini USB. Selain data tekanan udara, suhu dan kelembaban, juga menampilkan data waktu untuk hari, tanggal dan tahun pada saat itu. Data tersebut berasal dari sistem komputer, sehingga pada aplikasi akan menampilkan sesuai dengan format yang ada pada komputer begitu juga

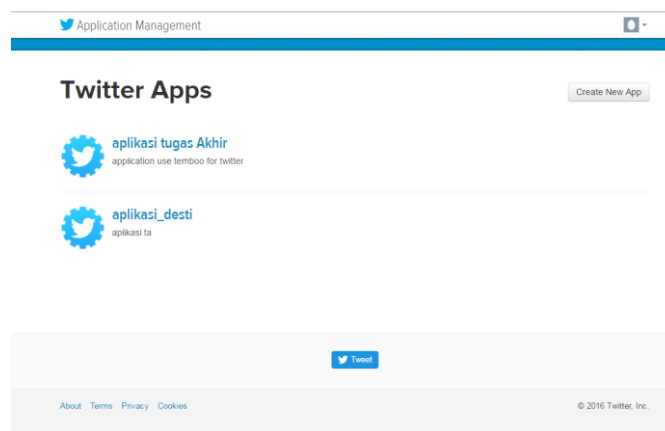
dengan tampilan data waktu untuk jam, menit dan detik yang berasal dari sistem komputer.

Tampilan program pada bagian perekam data oleh perangkat pemrograman *Processing*, untuk data tekanan udara, suhu dan kelembaban akan disimpan dalam komputer. Tombol keluar digunakan untuk keluar dari program aplikasi, juga berfungsi untuk menyimpan hasil perekam data tekanan udara, suhu dan kelembaban didalam komputer.

### 3.2.2.2.3 Perancangan Tampilan Data di Twitter

Pada perancangan untuk bisa menampilkan data di *Twitter* ada 3 hal yang perlu dilakukan yaitu mendapatkan API Key dari *Twitter*, melakukan pengaturan di Temboo dan pengaturan tampilan di *Twitter*.

Untuk API Key (*Application Programming Interface Key*) merupakan kode digit yang dikeluarkan oleh setiap situs seperti *Twitter* yang digunakan pada Tugas Akhir ini. Hal yang pertama dilakukan harus sudah mempunyai akun *Twitter* dan masuk ke <https://apps.Twitter.com> kemudian akan mendapatkan halaman awal seperti (Gambar 3.12).



Gambar 3.12 Tampilan *Application Management* di *Twitter*

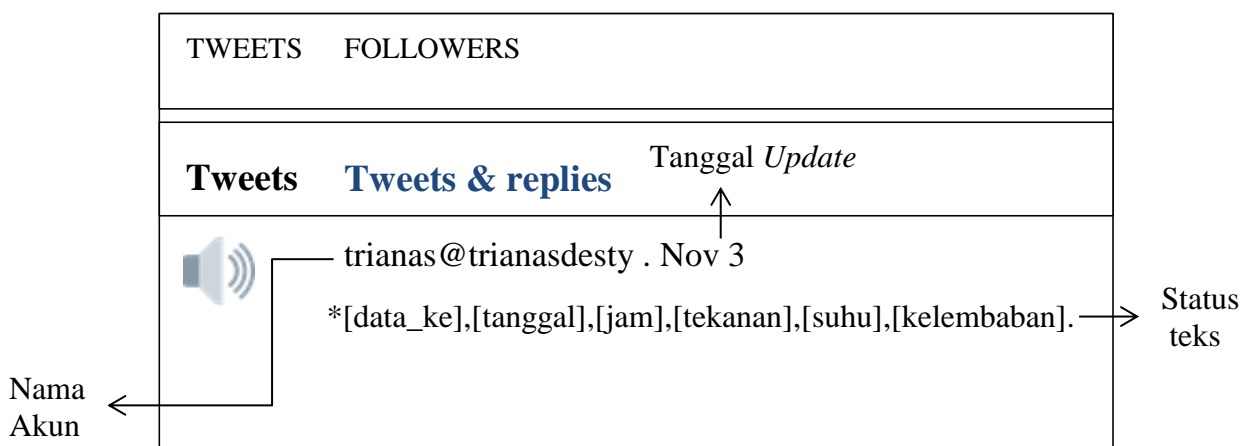
Apabila sudah mendapatkan halaman di atas, kemudian klik **Create New App**, kemudian akan diperintahkan mengisi Nama, Deskripsi, *Website* dan *Callback URL* pada halaman *form* entri. Selanjutnya, apabila ingin membaca *developer agreement* yang berisi perjanjian pengembang, kemudian ceklist **Yes, I agree** kemudian klik **Create Your Twitter Application**. Selanjutnya akan tampilkan halaman yang berisi *consumer key* (API Key), *Consumer Secret* (API Secret), *Access Level*, *Owner* dan *Owner ID*. Dan

untuk mendapatkan *access token* bias klik pada **Create my access token** dan akan langsung didapat *access token* yang diinginkan.

Selanjutnya melakukan pengaturan di Temboo, pertama masuk ke <https://Temboo.com> yang akan ditampilkan halaman Temboo dan pilih *icon Twitter* di Choreos, selanjutnya pilih *icon tweets* dan pilih *StatusesUpdate*, selanjutnya akan dimunculkan tampilan yang harus mengisi untuk *AccessToken*, *AccessTokenSecret*, *ConsumerKey*, *ConsumerSecret* dan *StatusUpdate* sesuai dengan tampilan di API Key di *Twitter* sebelumnya.

Di bawah ini merupakan rancangan tampilan hasil pengiriman data di *Twitter*, dimulai dengan data tekanan udara, suhu dan kelembababan direkam oleh aplikasi *Processing* menggunakan ekstensi CSV (*Comma Separated Values*). Pada pengiriman data ke *Twitter* akan digunakan akun trianas (@trianasdesty). Tampilan data dari hasil pengiriman ke *Twitter* untuk nilai parameter yang direkam akan dipisah oleh tanda koma. Berikut susunan dari tampilan data yang akan ditampilkan melalui *Twitter* :

\*[data\_ke],[tanggal],[jam],[tekanan],[suhu],[kelembaban].



Gambar 3.13 Rancangan Hasil Pengiriman Data di *Twitter*

Skrip aplikasi pada *Processing* di Tugas Akhir ini memiliki nama Rancangan\_Tampilan\_Processing.pde yang terdapat di *file* bernama Rancangan\_Tampilan\_Processing.zip. Pada skrip di bawah ini terdapat hal yang harus diatur terlebih dahulu sebelum dijalankan. Yaitu mengisikan *port* serial yang terhubung ke mikropengendali sesuai dengan yang terdeteksi di *Device Manager*. Pada Tugas Akhir ini, *port* serial yang digunakan adalah COM3 dan atur bitrate ke 4800 bps.

```
//Inisialisasi port serial
```

```
//println(Serial.list());  
dataPort = new Serial(this, "COM3", 4800);
```

Selanjutnya menuliskan *[Account]*, *[Aplication]* dan *[Key]* sesuai dengan data yang diperoleh dari Temboo dengan melihatnya dibagian pengaturan akses di Tembonya. `TembooSession` session=new `TembooSession("[namauser]", "[myFirstAppmy]", "[qcirCaNky0YPov0VeoVdYAdQyUFE8uaj]");`

Untuk pemrograman selanjutnya mengisikan *[Cosumer Key]*, *[Access Token Secret]* sesuai informasi yang didapatkan dari *Twitter* atau dapat dilihat dibagian API Key untuk *Twitter*.

```
statusesUpdateChoreo.setConsumerKey("[Consumer  
Key]");  
statusesUpdateChoreo.setAccessToken("[Access  
Token]");  
statusesUpdateChoreo.setConsumerSecret("[Consumer  
Secret]");  
statusesUpdateChoreo.setAccessTokenSecret("[Access  
Token Secret]");
```