

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

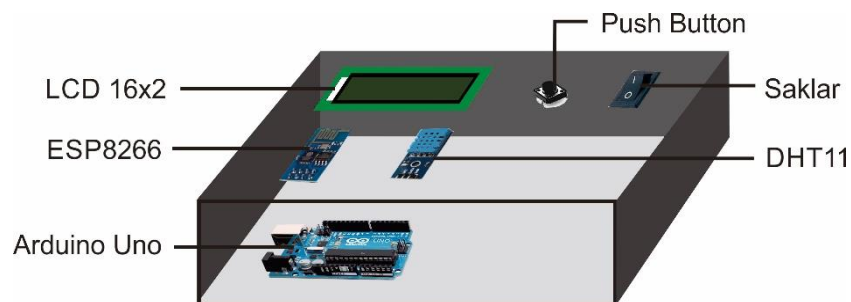
Pada bab 5 ini akan menjelaskan tentang perancangan sistem dan implementasi sistem secara terperinci baik dari sisi perancangan dan implementasi perangkat keras maupun dari sisi perangkat lunak.

### 5.1 Perancangan Sistem

Dalam perancangan sistem ini akan menjabarkan keseluruhan perancangan sistem mulai dari perancangan *Prototype* alat, perancangan perangkat keras dan perancangan perangkat lunak.

#### 5.1.1 Perancangan *Prototype* Alat Prediksi Hama Wereng

Perancangan *Prototype* dalam Sistem Prediksi Hama Wereng membahas beberapa perangkat atau komponen yang saling terhubung satu sama lain sehingga membentuk satu kesatuan sistem. Dalam perancangannya, desain *Prototype* dirancang menggunakan software CorelDraw X6 untuk menampilkan perancangan yang ada. *Prototype* berbentuk kotak berwarna hitam dengan ukuran panjang 18 cm, lebar 11 cm dan tinggi 6 cm. Bentuk *Prototype* alat ditunjukkan pada Gambar 5.1 berikut.

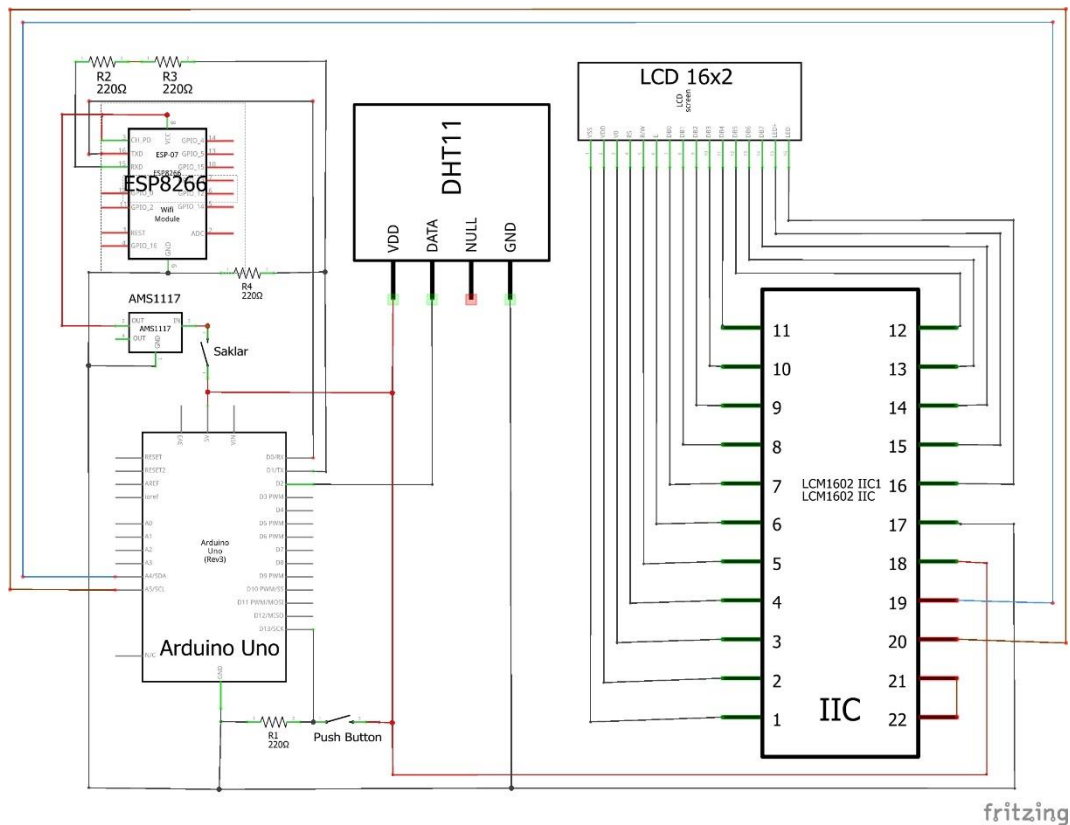


**Gambar 5.1 Desain *Prototype* Alat**

Pada bagian dalam kotak terdapat Mikrokontroler Arduino Uno sebagai unit pemroses yang terhubung ke semua perangkat lainnya. Sensor DHT11 diletakkan didalam namun menghadap dan terlihat dari luar, hal ini bertujuan agar nilai suhu dan kelembaban lebih terfokus di bagian luar kotak atau disekeliling alat. Modul WIFI ESP8266 terletak di sebelah sensor DHT11 dengan sisi antena menghadap ke luar kotak, hal ini bertujuan agar modul ESP8266 dapat lebih baik dalam menangkap sinyal dari koneksi internet yang ada. Informasi nilai data dan hasil klasifikasi sistem akan ditampilkan di LCD 16x2 yang ditempat menghadap ke arah pengguna sistem. *Push button* digunakan untuk memberikan *trigger* kepada sistem agar memilih nilai data yang terakhir dan mengolahnya sehingga akan menghasilkan klasifikasi prediksi hama wereng. Terdapat juga saklar yang digunakan untuk memutus tegangan ke modul ESP8266 saat melakukan *upload* program ke sistem.

### 5.1.2 Perancangan Perangkat Keras

Perancangan perangkat keras memuat semua detail hubungan antara satu perangkat dengan perangkat lain sesuai dengan kebutuhan yang diharapkan oleh sistem. Dalam hal ini akan dijabarkan hubungan skematik antara tiap pin – pin disetiap perangkat. Dimana Mikrokontroler Arduino Uno sebagai perangkat utama yang tersambung dengan beberapa perangkat lainnya seperti sensor DHT11, modul ESP8266, LCD16x2 melalui IIC (I2C), *Push button*, saklar, IC AMS1117 dan resistor. Secara keseluruhan kematik perancangan perangkat keras dapat dilihat pada Gambar 5.2 dibawah ini.



**Gambar 5.2 Diagram Skematik Sistem**

Pada diagram skematik diatas terdapat *push button* yang terhubung ke pin digital 13 dan terhubung ke GND melalui resistor 220 ohm, sedangkan disisi yang lain terhubung dengan VCC 5V dari Arduino Uno. Pada modul ESP8266, sumber tegangan VCC dan pin CH\_PD diambil dari *output* IC AMS1117 sebagai penurun tegangan dari 5V Arduino UNO ke 3.3V, hal ini sudah dijelaskan pada kebutuhan perangkat keras IC AMS1117. Pin GND ESP8266 dihubungkan ke GND AMS1117 dan Arduino Uno, untuk pin RX ESP8266 dihubungkan ke pin TX Arduino Uno dengan menambahkan secara seri 2 resistor dan 1 resistorukuran 220 ohm ke GND . Secara keseluruhan rangkaian antara pin modul ESP8266, IC AMS1117 dan Arduino Uno dijelaskan pada Tabel 5.1 berikut.

**Tabel 5.1 Koneksi pin ESP8266, IC AMS1117 dan Arduino Uno**

Pin ESP8255	Pin IC AMS1117	Pin Arduino Uno
-	IN 5V	5V
GND	GND	GND
VCC, CH_PD	OUT 3.3V	-
TX	-	RX
RX	-	TX

Pada sensor DHT11 terdapat 3 pin yang terhubung dengan Arduino Uno, yaitu pin VCC, GND dan pin data yang terhubung ke pin digital 2 Arduino Uno. Berikut ini adalah tabel keterangan koneksi antara DHT11 dan Arduino Uno.

**Tabel 5.2 Koneksi pin DHT11 dengan Arduino Uno**

Pin DHT11	Pin Arduino Uno
VCC	5V
GND	GND
DATA	D2

Rangkaian untuk menampilkan hasil sistem menggunakan LCD16x2 membutuhkan tambahan modul IIC (I2C). Hal ini bertujuan untuk mengurangi jumlah pin LCD16x2 yang terkoneksi langsung ke Arduino Uno. Dengan menggunakan modul IIC (I2C) sistem dapat mengurangi penggunaan pin yang semula bisa menggunakan 11 pin menjadi 4 pin saja. Selain mengurangi jumlah penggunaan pin, modul IIC (I2C) juga sekaligus sudah menggantikan penambahan potensiometer yang biasanya digunakan untuk mengatur gelap atau terangnya tampilan pada layar LCD 16x2 dengan adanya trimpot yang sudah ada pada modul IIC (I2C). Pada Tabel 5.3 dibawah menunjukkan detail koneksi antara LCD16x2 ke modul IIC (I2C) dan antara modul IIC (I2C) ke pin Arduino Uno.

**Tabel 5.3 koneksi antara LCD 16x2 dan Modul IIC (I2C)**

LCD 16x2	IIC (I2C)
VSS	1
VDD	2
VO	3
RS	4
R/W	5
E	6
DB0	7
DB1	8
DB2	9
DB3	10
DB4	11
DB5	12
DB6	13

DB7	14
LED+	15
LED-	16

Pada hal ini, penggunaan modul IIC (I2C) diibaratkan jembatan antara LCD 16x2 dengan Arduino Uno. Penggunaan modul IIC (I2C) dengan Arduino Uno hanya menggunakan 4 pin diantara keduanya,,dari sisi modul IIC (I2C) terdapat pin SCL, SDA, VCC dan GND. Pin SCL (*Serial Clock*) dan pin SDA (*Serial Data*) merupakan pin yang membawa informasi antara IIC (I2C) dengan pengontrolnya yaitu Arduino Uno. Tabel 5.4 menunjukkan sambungan antara pin modul IIC (I2C) dengan pin Arduino Uno secara terperinci.

**Tabel 5.4 Koneksi pin IIC (I2C) dengan pin Arduino Uno**

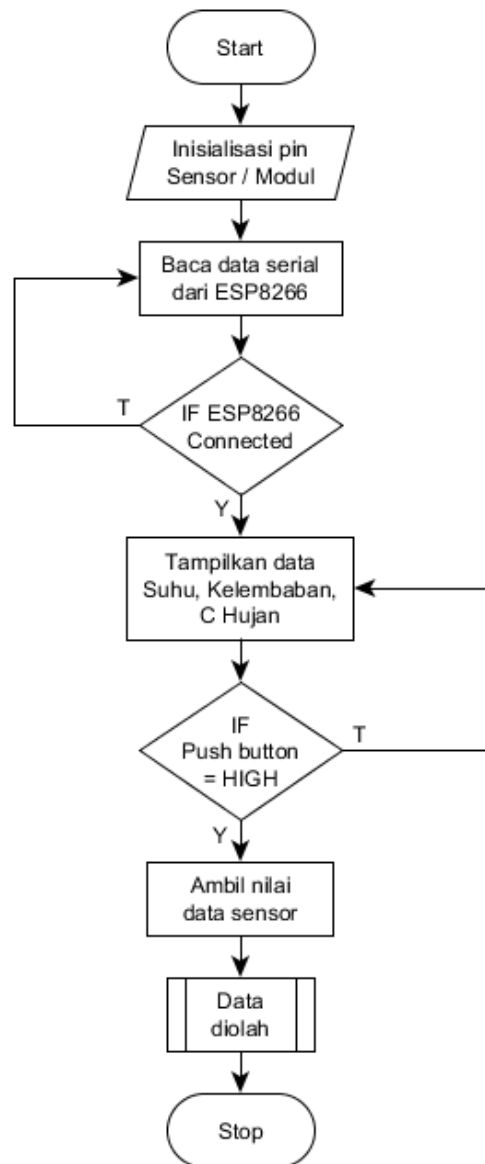
Pin IIC (I2C)	Pin Arduino Uno
VCC	VCC
GND	GND
SCL	SCL
SDA	SDA

### 5.1.3 Perancangan Perangkat Lunak

Pada perancangan perangkat lunak akan dibagi menjadi 2, yaitu membahas mengenai perancangan perangkat lunak pada mikrokontroler saat pengambilan data pada sensor DHT11 dan modul ESP8266 serta perancangan perangkat lunak pada saat perhitungan klasifikasi prediksi hama wereng menggunakan metode *Naive Bayes* .

#### 5.1.3.1 Perancangan Pengambilan Data Uji

Dalam proses pengambilan data sensor DHT11 dan ESP8266 atau yang disebut data uji, terdapat beberapa tahap yang perlu dilakukan secara bertahap agar proses pengambilan data dapat berjalan sesuai kebutuhan sistem. Secara keseluruhan, proses pengambilan data uji akan dijabarkan pada Gambar 5.3 tentang *flowchart* atau digram alir dari pengambilan data.



**Gambar 5.3 Diagram alir perancangan perangkat lunak pengambilan data uji**

Perancangan perangkat lunak pada pengambilan data uji bertujuan untuk menguraikan secara detail proses yang terjadi dari awal hingga data siap diolah untuk diklasifikasi dengan *Naive Bayes*. Pertama, sistem akan menginisialisasi penggunaan pin DHT11 dan ESP8266 untuk membedakan *input* dan *output* yang digunakan. Kemudian sistem menunggu ESP8266 terkoneksi dengan internet dengan cara membaca hasil data serial yang dikirimkan oleh ESP8266 ke Arduino Uno. Jika ESP8266 telah mengirimkan data nilai curah hujan yang diakses secara online, maka sistem akan menampilkan secara keseluruhan data nilai suhu, kelembaban dan curah hujan ke LCD 16x2.

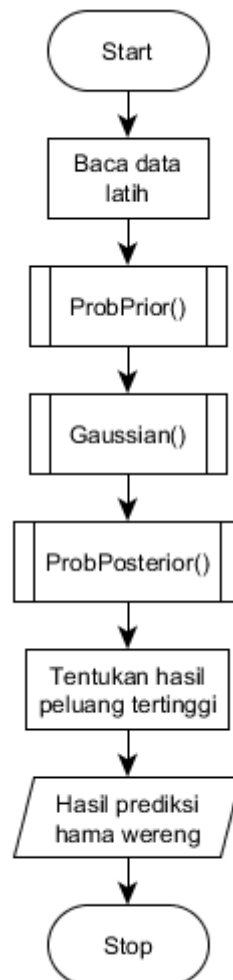
Proses pengambilan data curah hujan yang dilakukan oleh ESP8266 dari situs [www.accuweather.com](http://www.accuweather.com) membutuhkan platform *thingspeak* sebagai jembatan yang memproses data dan menghasilkan alamat / URL yang dapat di akses secara

langsung oleh ESP8266. Struktur data dari data didapatkan ESP8266 setelah melakukan *request* ke thingspeak terdiri dari beberapa atribut seperti status *request*, informasi data, atribut protokol HTTP, informasi permintaan data, *date*, informasi server dan nilai data utama yang diinginkan.

Data yang sudah diambil oleh DHT11 dan ESP8266 akan ditampilkan secara terus menerus sampai menunggu *trigger* dari penekanan *push button*. Jika *push button* sudah ditekan, sistem akan memilih nilai data terakhir yang akan diolah untuk diklasifikasikan menggunakan metode *Naive Bayes* guna memprediksi ada atau tidaknya hama wereng.

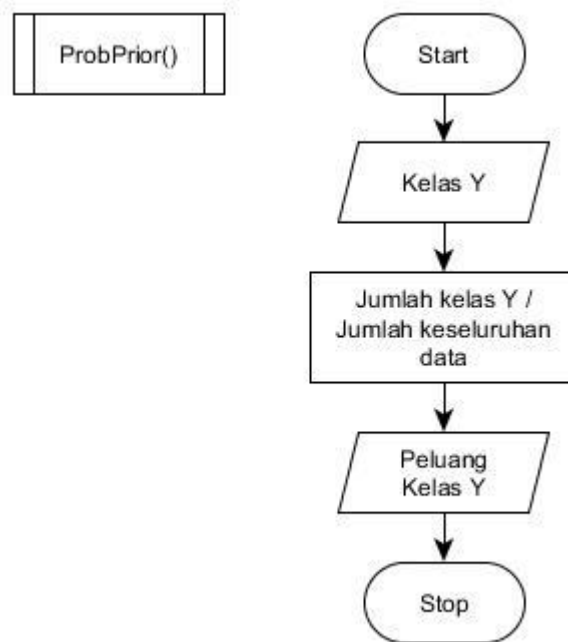
### 5.1.3.2 Perancangan Klasifikasi *Naive Bayes*

Dalam proses perancangan klasifikasi *Naive Bayes* terdapat beberapa tahapan yang harus dilakukan. Proses klasifikasi bisa dilakukan jika proses pengambilan data sudah berjalan dan sudah terdapat *trigger* dari penekanan *push button* oleh pengguna. Berikut adalah diagram alir dari klasifikasi *Naive Bayes*.



Gambar 5.4 Diagram alir perancangan perangkat lunak *Naive Bayes*

Hasil pembacaan data oleh sensor merupakan nilai yang akan dijadikan sebagai fitur untuk mempengaruhi penentuan klasifikasi prediksi hama wereng dengan perhitungan *Naive Bayes*. Selain itu, hasil klasifikasi juga dipengaruhi oleh nilai dari data latih. Proses klasifikasi dimulai dengan menentukan hasil dari fungsi *ProbPrior()*, kemudian menentukan hasil dari fungsi *Gaussian()*, lalu menentukan hasil dari fungsi *ProbPosterior()*, setelah itu menentukan hasil nilai peluang tertinggi yang didapatkan setiap kelas. Hasil dari peluang akan dihitung tingkat akurasi, data uji dengan nilai error terkecil akan dijadikan sebagai data latih pada perhitungan berikutnya. Untuk penjelasan dari fungsi-fungsi tersebut akan dijelaskan pada beberapa diagram alir berikut ini.



**Gambar 5.5 Diagram alir fungsi ProbPrior()**

Dalam mengklasifikasikan prediksi hama wereng menggunakan *Naive Bayes*, tahap pertama yang dilakukan yaitu dengan menghitung nilai *prior* dari setiap kelas prediksi. *Prior* merupakan nilai terjadinya suatu kelas dengan cara membagi banyaknya data dalam suatu kelas dengan jumlah keseluruhan data yang ada pada data latih. Pada sistem ini terdapat 2 kelas prediksi, yaitu ada wereng dan tidak ada wereng.

Pada pengambilan data latih didapatkan data dari hasil rekapitulasi keadaan organisme pengganggu tumbuhan (OPT) tanaman pangan Dinas Tanaman Pangan, Holtikultura dan Perkebunan Kabupaten Malang. Diambil data serangan hama wereng (WBC) di lokasi yang dipilih untuk dilakukan klasifikasi prediksi hama wereng. Dalam penentuan keadaan ada hama wereng diambil dari parameter sisa serangan dan tambah serangan pada data hasil rekapitulasi. Data latih terdiri dari kumpulan keadaan serangan yang terjadi di tahun 2017, dengan disertai data suhu, kelembaban dan curah hujan yang terjadi pada waktu yang bersamaan disaat itu.

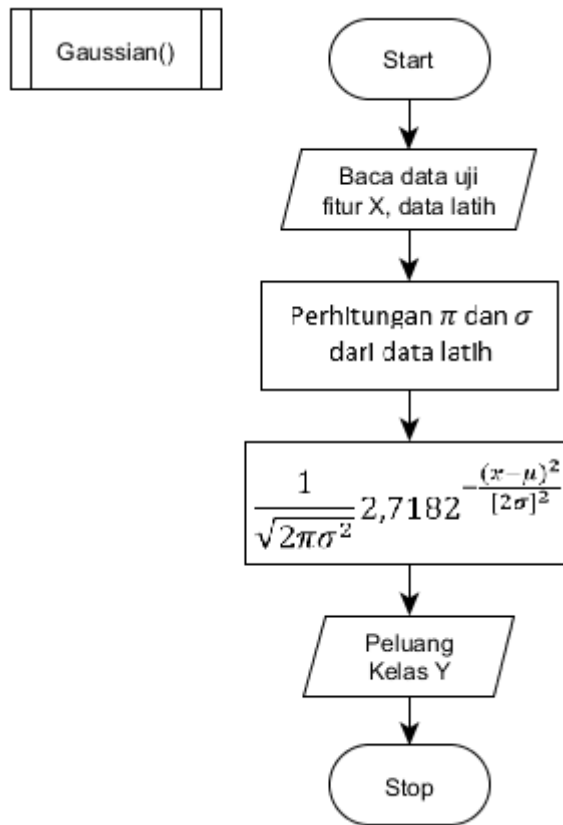
Terdapat 19 keadaan serangan hama wereng yang dijadikan sebagai data latih, dimana selain dari hasil data rekapitulasi terdapat juga 2 keadaan serangan yang diambil dari hasil evaluasi perhitungan error klasifikasi seperti yang dijelaskan di Gambar 5.5 sebelumnya. Secara keseluruhan data latih akan ditampilkan pada Tabel 5.5.

**Tabel 5.5 Data Latih serangan hama wereng tahun 2017**

No.	Waktu	Keadaan Serangan	Suhu (°C)	Kelembaban (%)	C Hujan (mm)
1	Januari P2	Ada	25,2438	87,6875	13,8666
2	Februari P1	Ada	25,6067	84,8	16,0909
3	Februari P2	Ada	26,1846	85,5385	20
4	Maret P1	Ada	26,275	82,8333	9,7777
5	April P1	Ada	26,3769	83,1538	9,125
6	April P2	Ada	26,4467	83,1333	13,8
7	Mei P1	Ada	26,1	81,2667	27,6
8	Mei P2	Ada	25,7714	80,5714	16,6667
9	Juni P1	Ada	25,6733	82,5333	8
10	Januari P1	Tidak Ada	26,2333	84,2	31,1
11	Juni P2	Tidak Ada	25,34667	80,4667	18
12	Juli P1	Tidak Ada	24,6267	82,5333	2,5
13	Juli P2	Tidak Ada	24,3	81,645	6
14	Agustus P1	Tidak Ada	24,36667	81,53333	1
15	Agustus P2	Tidak Ada	24,50625	79,3125	1
16	September P1	Tidak Ada	25,30667	77,6	1
17	September P2	Tidak Ada	25,89333	78	1
18	Hasil Evaluasi 1	Tidak Ada	25	80	5
19	Hasil Evaluasi 2	Ada	26	80	15

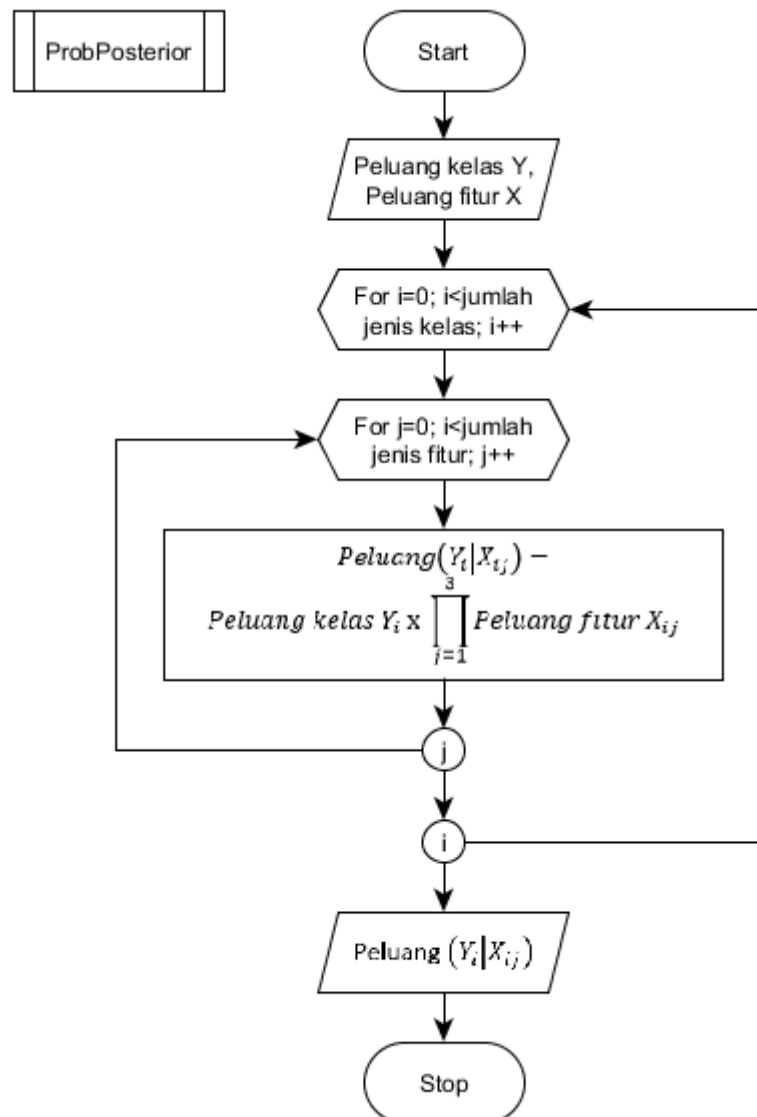
Tabel 5.5 diatas merupakan data latih dimana setiap keadaan serangan hama wereng disertai data suhu dan kelembaban yang didapatkan dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG). Untuk data curah hujan didapatkan dari situs penyedia data cuaca [www.accuweather.com](http://www.accuweather.com). Penentuan nilai data suhu, kelembaban dan curah hujan diambil dari rata – rata nilai yang terjadi setiap periode keadaan serangan. Periode keadaan serangan yaitu awal bulan hingga tanggal 15 (P1) dan tanggal 16 hingga akhir bulan (P2).





**Gambar 5.6 Diagram alir fungsi *Gaussian()***

Pada tahap kedua, sistem akan menentukan nilai peluang dari setiap fitur yang ada. Terdapat 3 fitur yang digunakan dalam sistem ini yaitu fitur pembacaan suhu, kelembaban yang dilakukan oleh sensor DHT11 dan fitur curah hujan yang diambil secara *online* menggunakan modul ESP8266. Kemudian perlu dilakukan perhitungan nilai *mean* dan standar deviasi masing - masing fitur yang ada pada data latih dengan menggunakan **Persamaan(2.4)** dan **Persamaan(2.5)**. Hasil perhitungan dari *mean* dan standar deviasi akan dijadikan nilai yang mewakili data latih pada sistem. Nilai tersebut akan dimasukkan pada pemrograman mikrokontroler untuk memudahkan sistem mengakses nilai pada data laith yang ada. Kemudian dilanjutkan dengan melakukan perhitungan *Gaussian* menggunakan **Persamaan(2.3)**. Setelah itu sistem akan mendapatkan peluang dari kelas ada wereng dan tidak ada wereng.



**Gambar 5.7 Diagram alir fungsi ProbPosterior()**

Setelah perhitungan *Gaussian*, tahap selanjutnya yaitu menentukan nilai dari posterior, nilai peluang posterior adalah peluang yang menentukan besarnya peluang setiap kelas pada sistem. Peluang tersebut akan ada ketika setiap fitur telah emndapatkan masukkan nilai dari sensor. Proses penentuan besarnya peluang tiap kelas dilakukan dengan melakukan perkalian antara hasil dari fungsi ProbPrior() dengan fungsi *Gaussian*().

Pada tahap akhir, proses klasifikasi dengan metode *Naive Bayes* akan menentukan nilai peluang yang tertinggi dari setiap posterior dengan membandingkan nilai posteroir dengan nilai peluang posterior lainnya. Kelas dengan nilai peluang posterior tertinggi merupakan hasil akhir dari klasifikasi kelas ada wereng dan tidak ada wereng.

Dalam perhitungan manual untuk proses klasifikasi prediksi keadaan hama wereng dapat dilakukan percobaan perhitungan, sebagai contoh terdapat fitur data uji dengan nilai suhu = 26, kelembaban = 81, curah hujan = 20 dengan berdasarkan data latih pada Tabel 5.5 :

1. Menghitung peluang *prior* pada setiap keadaan serangan hama wereng

$$P_{Ada\ wereng} = \frac{Jumlah\ keadaan\ ada\ wereng}{Jumlah\ seluruh\ data} = \frac{10}{19} = 0,526315$$

$$P_{Tidak\ ada\ wereng} = \frac{Jumlah\ keadaan\ tidak\ ada\ wereng}{Jumlah\ seluruh\ data} = \frac{9}{19} = 0,473684$$

2. Menghitung nilai *mean* dan standar deviasi tiap fitur data latih pada masing – masing keadaan serangan

- *Mean*

$$\begin{aligned} \bar{X}_{Suhu(Ada\ wereng)} &= \frac{\sum_{i=1}^n Suhu_i}{Jumlah\ jenis\ ada\ wereng} \\ &= \frac{25,243 + 25,606 + 26,184 + 26,275 + \dots + 26}{10} = 25,9678 \end{aligned}$$

$$\begin{aligned} \bar{X}_{Kelembaban(Ada\ wereng)} &= \frac{\sum_{i=1}^n Kelembaban}{Jumlah\ jenis\ ada\ wereng} \\ &= \frac{87,687 + 84,8 + 85,538 + 82,833 + \dots + 80}{10} = 83,1518 \end{aligned}$$

$$\begin{aligned} \bar{X}_{C\ Hujan(Ada\ wereng)} &= \frac{\sum_{i=1}^n C\ Hujan_i}{Jumlah\ jenis\ ada\ wereng} \\ &= \frac{13,866 + 16,09 + 20 + 9,777 + \dots + 15}{10} = 14,9927 \end{aligned}$$

Dilakukan juga perhitungan yang sama untuk mencari nilai *mean* tiap fitur data latih dari jenis keadaan tidak ada wereng. Hasil perhitungan *mean* secara keseluruhan jenis keadaan wereng ditunjukkan pada Tabel 5.6.

**Tabel 5.6 Mean tiap fitur dari jenis keadaan wereng**

	Suhu	Kelembaban	C Hujan
Ada wereng	25,96784	83,15178	14,99269
Tidak ada wereng	25,06439	80,58787	7,4

- Standar deviasi

$$\begin{aligned}\sigma_{Suhu(Ada\ wereng)} &= \sqrt{\frac{\sum_{i=1}^n (Suhu_i - \bar{x}_{Suhu(Ada\ wereng)})^2}{Jumlah\ jenis\ ada\ wereng - 1}} \\ &= \sqrt{\frac{(25,245 - 25,967)^2 + (25,606 - 25,967)^2 + \dots + (26 - 25,967)^2}{10 - 1}} \\ &= 0,385352391\end{aligned}$$

$$\begin{aligned}\sigma_{Kelem(Ada\ wereng)} &= \sqrt{\frac{\sum_{i=1}^n (Kelembaban_i - \bar{x}_{Kelembaban(Ada\ wereng)})^2}{Jumlah\ jenis\ ada\ wereng - 1}} \\ &= \sqrt{\frac{(87,687 - 83,151)^2 + (84,8 - 83,151)^2 + \dots + (80 - 83,151)^2}{10 - 1}} \\ &= 2,347947443\end{aligned}$$

$$\begin{aligned}\sigma_{C\ Hujan(Ada\ wereng)} &= \sqrt{\frac{\sum_{i=1}^n (C\ Hujan_i - \bar{x}_{C\ Hujan(Ada\ wereng)})^2}{Jumlah\ jenis\ ada\ wereng - 1}} \\ &= \sqrt{\frac{(13,866 - 25,967)^2 + (16,09 - 25,967)^2 + \dots + (15 - 25,967)^2}{10 - 1}} \\ &= 5,785422823\end{aligned}$$

Dilakukan juga perhitungan yang sama untuk mencari nilai standar deviasi tiap fitur data latih dari jenis keadaan tidak ada wereng. Hasil perhitungan standar deviasi secara keseluruhan jenis keadaan wereng ditunjukkan pada Tabel 5.7.

**Tabel 5.7 Standar deviasi tiap fitur dari jenis keadaan wereng**

	Suhu	Kelembaban	C Hujan
<b>Ada wereng</b>	0,38535	2,34794	5,78542
<b>Tidak ada wereng</b>	0,68649	2,13468	10,42604

3. Menghitung nilai *gaussian* tiap fitur yang diujikan (suhu = 26, kelembaban = 81, curah hujan = 20)

$$\begin{aligned}P(Suhu = 26|Ada) &= \frac{1}{\sqrt{2\mu\sigma_{Suhu(Ada)}^2}} e^{-\frac{(26 - \mu_{Suhu(Ada)})^2}{2\sigma_{Suhu(Ada)}^2}} \\ &= \frac{1}{\sqrt{2 \times 3,14 \times 0,38535^2}} e^{-\frac{(26 - 25,967)^2}{2 \times 0,38535^2}} \\ &= 1,03192890\end{aligned}$$

$$\begin{aligned}
P(\text{Suhu} = 26 | \text{Tdk ada}) &= \frac{1}{\sqrt{2\mu\sigma_{\text{Suhu(Tdk ada)}}^2}} e^{-\frac{(26 - \mu_{\text{Suhu(Tdk ada)}})^2}{2\sigma_{\text{Suhu(Tdk ada)}}^2}} \\
&= \frac{1}{\sqrt{2 \times 3,14 \times 0,68649^2}} e^{-\frac{(26 - 25,06439)^2}{2 \times 0,68649^2}} \\
&= 0,22964546
\end{aligned}$$

Dilakukan juga perhitungan yang sama untuk mencari nilai *gaussian* tiap fitur dari jenis keadaan tidak ada wereng. Hasil perhitungan *gaussian* secara keseluruhan jenis keadaan wereng ditunjukkan pada Tabel 5.8.

**Tabel 5.8 Nilai *gaussian* tiap fitur dari jenis keadaan wereng**

	Suhu	Kelembaban	C Hujan
Ada wereng	1,03192890	0,11167405	0,04742644
Tidak ada wereng	0,22964546	0,18348196	0,01843983

4. Menghitung nilai peluang posterior tiap kelas atau keadaan serangan

$$\begin{aligned}
&P(\text{Ada} | \text{Suhu}_{=26}, \text{Kelem}_{=80}, \text{C Hujan}_{=20}) \\
&= P_{\text{Ada}} \times P(\text{Suhu} = 26 | \text{Ada}) \times P(\text{Kelem} = 81 | \text{Ada}) \times P(\text{C Hujan} = 20 | \text{Ada}) \\
&= 0,526315 \times 1,03192890 \times 0,11167405 \times 0,04742644 \\
&= 0,002876526
\end{aligned}$$

$$\begin{aligned}
&P(\text{Tdk ada} | \text{Suhu}_{=26}, \text{Kelem}_{=80}, \text{C Hujan}_{=20}) \\
&= P_{\text{Tdk ada}} \times P(\text{Suhu} = 26 | \text{Tdk ada}) \times P(\text{Kelem} = 81 | \text{Tdk ada}) \times \\
&\quad P(\text{C Hujan} = 20 | \text{Tdk ada}) \\
&= 0,473684 \times 0,22964546 \times 0,18348196 \times 0,01843983 \\
&= 0,000368042
\end{aligned}$$

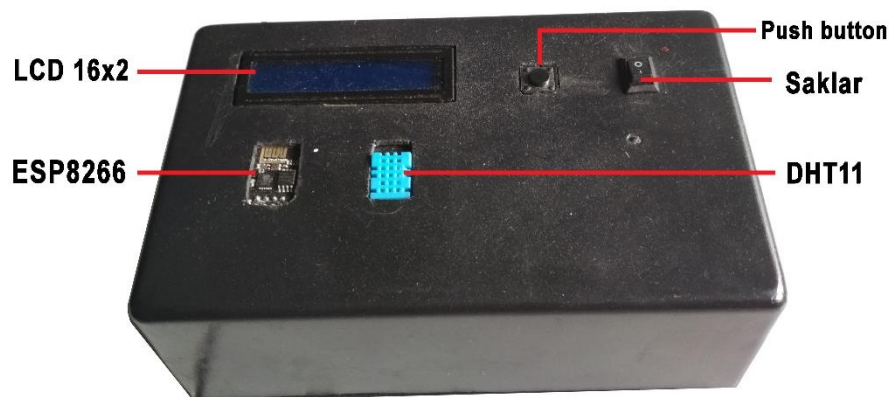
Dari hasil perhitungan peluang posterior, didapatkan hasil perkalian antara peluang *prior* dan nilai *gaussian* tiap fitur yang diujikan. Nilai peluang posterior tertinggi menunjukkan bahwa peluang ada wereng lebih tinggi dari pada peluang tidak ada wereng pada data uji Suhu = 26, Kelembaban = 81, Curah hujan = 20. Dengan demikian hasil perhitungan termasuk pada jenis keadaan **ada wereng**.

## 5.2 Implementasi Sistem

Pada tahap implementasi sistem, semua perancangan yang telah dilakukan pada tahap sebelumnya akan direalisasikan pada sistem secara keseluruhan sesuai dengan kebutuhan sistem. Beberapa hal yang akan dijelaskan dalam subbab ini diantaranya terkait implementasi *Prototype*, implementasi pada sisi perangkat keras dan implementasi pada sisi perangkat lunak.

### 5.2.1 Implementasi *Prototype* Alat Prediksi Hama Wereng

Pada tahap implementasi *Prototype* alat prediksi hama wereng ini mengacu pada perancangan yang ada pada subbab 5.1.1, sehingga pada realisasinya berbentuk kotak berwarna hitam dengan ukuran 18 x 11 x 6 cm<sup>3</sup>. Kotak tersebut terbuat dari bahan plastik dengan ketebalan sebesar 2 mm. Pada sebagian sisi atas dan samping kotak akan dilubangi untuk dijadikan tempat komponen atau modul yang dibutuhkan dalam implementasi sistem. *Prototype* akan ditunjukkan pada Gambar 5.8 berikut.

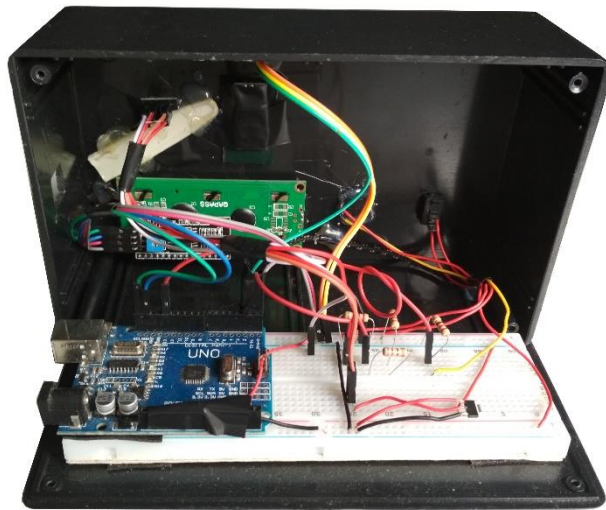


**Gambar 5.8 Implementasi *Prototype* Alat Prediksi Hama Wereng**

Dilihat dari bagian atas alat, terlihat LCD 16x2 untuk menampilkan data uji dan hasil prediksi, ESP8266 untuk mengambil data curah hujan, DHT11 untuk mengambil data suhu dan kelembaban, *Push Button* untuk memberikan *trigger* pada sistem untuk prediksi hama wereng, serta saklar yang digunakan untuk memutus tegangan dari mikronkontroler saat proses *upload* program dilakukan.

### 5.2.2 Implementasi Perangkat Keras

Tahap implementasi perangkat keras akan menjelaskan mengenai proses implementasi Mikrokontroler Arduino Uno, sensor DHT11, Modul ESP8266, *Push Button*, Saklar, dan LCD 16x2. Secara keseluruhan komponen tersebut dirangkai menjadi satu kesatuan menggunakan project board sebagai papan penghubung antar tiap komponen melalui media kabel jumper. Keadaan didalam kotak alat dapat dilihat pada gambar 5.9 dibawah ini.



**Gambar 5.9 Implementasi Perangkat Keras nampak dalam**

Pada Gambar 5.9 terlihat hasil implementasi perangkat keras secara keseluruhan dilihat dari dalam kotak alat. Mikrokontroler Arduino Uno dihubungkan dengan tiap komponen sesuai dengan perancangan pada subbab 5.1.2 yang menjelaskan hubungan tiap pin pada setiap komponen. Pada komponen sensor DHT11, Modul ESP8266, *Push Button*, Saklar, dan LCD 16x2 ditempatkan dan dihadapkan keatas pada bagian sisi atas kotak alat. Selain sebagai antar muka antara alat dan pengguna, penempatan dengan menghadapkan komponen keatas juga agar sensor DHT11 lebih baik dalam membaca kondisi suhu dan kelembaban diluar kotak atau disekitar alat tersebut berada. Begitu juga dengan ESP8266 disisi antena dihadapkan keatas atau keluar kotak agar ESP8266 lebih baik dalam menerima transmisi data internet dari koneksi wifi.



**Gambar 5.10 Implementasi Perangkat Keras nampak atas**

Pada Gambar 5.10 terlihat bagian atas alat yang terdiri dari LCD 16x2, *push button*, ESP8266 dan DHT11 sebagai komponen utama yang bekerja sesuai kebutuhan fungsionalnya dan sesuai perancangan pada subbab 5.1.2

### 5.2.3 Implementasi Perangkat Lunak

Pada implementasi perangkat lunak menjelaskan mengenai realisasi kode program dan proses perangkat lunak lainnya sesuai dengan perancangan yang telah dilakukan pada subbab 5.1.3. Untuk implementasi kode program dilakukan di software Arduino IDE 1.8.0 baik pada Mikrokontroler Arduino Uno atau ESP8266. Dalam proses pemrograman Arduino terdapat beberapa library yang digunakan untuk mempermudah implementasi pada fungsi tertentu, diantaranya yaitu library “dht.h” untuk menjalankan sensor DHT11, Library “math.h” digunakan untuk melakukan perhitungan matematika yang rumit atau kompleks, Library “ESP8266WiFi.h” untuk menjalankan modul Wifi ESP8266, library “LiquidCrystal\_I2C.h” digunakan untuk menjalankan LCD 16x2 yang berkomunikasi dengan Arduino Uno melalui modul I2C.

#### 5.2.3.1 Implementasi Kode Program Pengambilan Data Uji

Pada tahap awal implementasi proses pengambilan data uji dari sensor DHT11 dan modul ESP8266, setelah menambahkan *library* perlu dilakukan inisialisasi objek, variabel dan konfigurasi pin yang digunakan pada sensor. Penggunaan pin sensor disesuaikan dengan perancangan perangkat keras dan perancangan perangkat lunak pada subbab sebelumnya. Gambar 5.11 menunjukkan pada baris ke-2 digunakan untuk inisialisasi pin pada LCD 16x2, baris ke-5 dan baris ke-6 untuk deklarasi objek sensor dan pin yang digunakan oleh sensor DHT11, baris ke-9 hingga baris ke-19 merupakan inisialisasi variabel, secara detail pada baris ke-9 hingga baris ke-12 merupakan variabel dengan tipe data *char* dan *String* yang digunakan untuk membaca, menyeleksi dan menyimpan data dari ESP8266 melalui komunikasi serial. Kemudian baris ke-13 hingga baris ke-15 digunakan untuk menyimpan nilai awal dari parameter suhu, kelembaban dan curah hujan dari DHT11 dan ESP8266. Baris ke-16 hingga baris ke-18 merupakan variabel dengan tipe data *array*, dimana variabel “*baca*” dan “*baca1*” digunakan untuk menyimpan setiap nilai parameter dari tipe integer ke array, variabel masukan digunakan untuk menyimpan nilai data uji yang akan dijadikan *input* untuk klasifikasi. Baris ke-19 digunakan untuk cek data curah hujan, apakah ESP8266 sudah mengirimkan data nilai curah hujan atau belum.



```
Skripsi_V.5 §
1 // Inisialisasi library dengan alamat pin
2 LiquidCrystal_I2C lcd(0x3F , 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
3
4 // Deklarasi objek
5 dht DHT;
6 #define DHT11_PIN 2
7
8 // Deklarasi variabel
9 char bacaChar;
10 String readString;
11 String CHujan;
12 String cekNilai;
13 int nilaiSuhu;
14 int nilaiKelm;
15 int nilaiCHujan;
16 float baca[3];
17 float bacal[2];
18 float masukkan[3];
19 int cek = 0;
```

**Gambar 5.11 Inisialisasi variabel pembacaan data uji**

Pada Gambar 5.12 prose implementasi pembacaan nilai suhu dan kelembaban dari sensor DHT11 dilakukan dengan membaca *input* dari pin data yang digunakan seperti pada baris ke-2, kemudian diolah oleh *library* “dht.h” dan menghasilkan nilai suhu dan kelembaban. Baris ke-4 dan ke-5 menampilkan nilai data suhu dan menyimpannya ke variabel “nilaiSuhu”, pada baris ke-6 nilai pada variabel “nilaiSuhu” dimasukkan ke variabel “arraybaca1” index ke “0” dengan menambahkan angka 2 pada setiap hasilnya. Penambahan angka 2 berarti menjumlahkan nilai dari sensor dengan angka 2, hal tersebut dilakukan untuk mengkalibrasi nilai suhu sehingga meminimalkan nilai error yang berarti hasil penambahan tersebut semakin mendekati dengan nilai suhu yang dihasilkan oleh alat ukur.

Pada baris ke-9 hingga baris ke-11 akan menampilkan nilai kelembaban dan menyimpannya ke variabel “nilaiSuhu”, yang selanjutnya disimpan pada variabel “arraybaca1” index ke “1” dengan menambahkan nilai 20 untuk kalibrasi agar nilai hasil penjumlahan antara nilai sensor dan angka 20 semakin mendekati hasil dari alat ukur yang dijadikan pembanding untuk pengujian sensor pada penelitian ini.

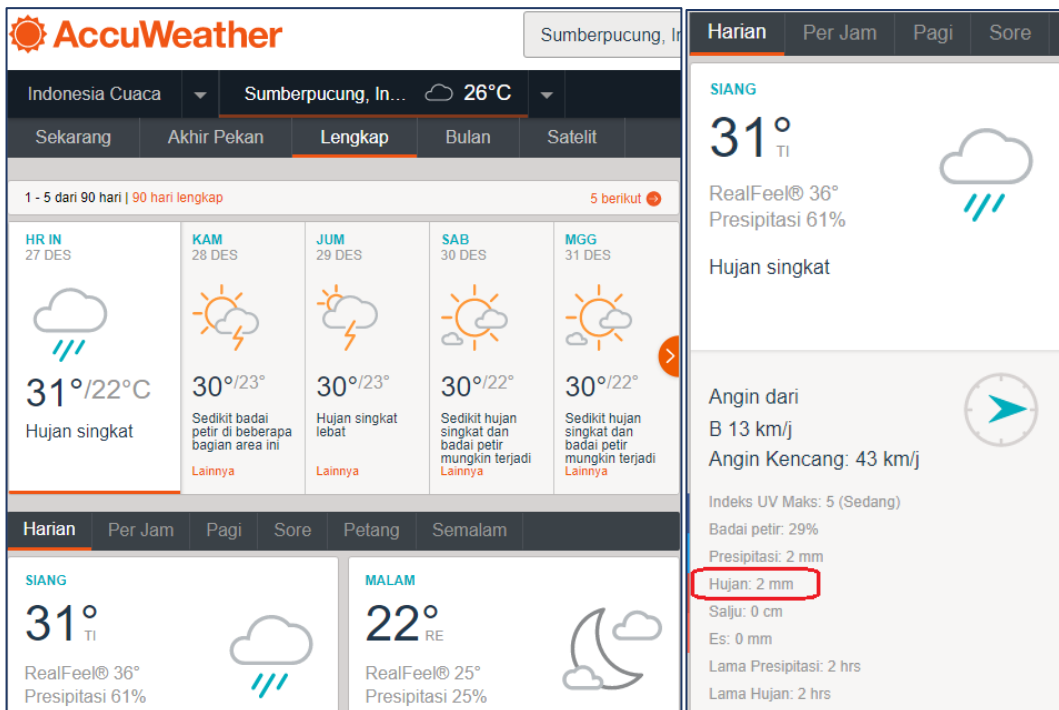
```

DHT11 §
1 void baca_DHT(float arraybacal[1]) {
2   int chk = DHT.read11(DHT11_PIN);
3   // Baca data Suhu
4   Serial.println(DHT.temperature);
5   nilaiSuhu = DHT.temperature;
6   arraybacal[0] = nilaiSuhu + 2;
7
8   // Baca data kelembaban
9   Serial.println(DHT.humidity);
10  nilaiKelm = DHT.humidity;
11  arraybacal[1] = nilaiKelm + 20;
12 }

```

**Gambar 5.12 Kode program pembacaan nilai sensor DHT11**

Untuk implementasi pembacaan nilai curah hujan dilakukan dengan membaca data yang diperoleh ESP8266 ketika berhasil *me-request* URL tertentu dan hasilnya kemudian dibaca oleh Arduino Uno melalui komunikasi serial.



**Gambar 5.13 Situs [www.accuweather.com](http://www.accuweather.com) sebagai sumber data curah hujan**

Secara detail, proses pertama yang dilakukan yaitu mencari situs yang menyediakan data curah hujan di daerah tertentu sesuai dengan kebutuhan, dimana pada sistem ini mengakses situs [www.accuweather.com](http://www.accuweather.com) sebagai sumber nilai data curah hujan, dari situs tersebut dicari data curah hujan seperti pada Gambar 5.13 yang ditandai dengan lingkaran merah. [www.accuweather.com](http://www.accuweather.com) dipilih karena mempunyai cakupan wilayah yang luas dimana memiliki data di lokasi yang dibutuhkan pada penelitian ini dan memiliki akurasi data yang baik.

Selanjutnya meng-copy Xpath yang didapatkan dengan cara melakukan blok pada data curah hujan kemudian klik kanan pilih inspect element dan cari element data curah hujan kemudian copy Xpath. Untuk mengambil nilai curah hujan dari situs diperlukan aplikasi tambahan sebagai jembatan agar data dapat diambil dengan menggunakan protokol HTTP, dalam hal ini menggunakan [www.thingspeak.com](http://www.thingspeak.com) sebagai Application Programming Interface (API) agar menghasilkan alamat yang digunakan untuk mendapatkan nilai curah hujan. Pada [www.thingspeak.com](http://www.thingspeak.com) digunakan fitur "ThingHTTP" dengan memasukkan alamat situs penyedia curah hujan dan Parse String yang didapat dari nilai Xpath kemudian memilih metode "Get" untuk mengambil data. Dari proses ini menghasilkan alamat atau URL yang jika diakses akan menghasilkan data nilai curah hujan dimana nilainya dapat berubah secara dinamis.

```

sketch_sep28aESP8266 $
1 #include <ESP8266WiFi.h>
2 const char* ssid      = "PD";
3 const char* password = "isiendewe";
4 const char* host = "api.thingspeak.com";
5 void setup() {
6   Serial.begin(115200);
7   delay(10);
8   Serial.print("Connecting to ");
9   Serial.println(ssid);
10  WiFi.begin(ssid, password);
11  while (WiFi.status() != WL_CONNECTED) {
12    delay(500);
13    Serial.print(".");
14  }
15  Serial.println(WiFi.localIP());
16 }
17 void loop() {
18   WiFiClient client;
19   const int httpPort = 80;
20   if (!client.connect(host, httpPort)) {
21     Serial.println("Connection failed");
22     return;
23   }
24   String url = "/apps/thinghttp/send_request?api_key=KJOBRF5714BQ4RJO";
25   client.print(String("GET ") + url + "&headers=false" + \ //lanjutan dibawah
26 " HTTP/1.1\r\n" + "Host: " + host + "\r\n" + "Connection: close\r\n\r\n");
27   delay(500);
28   int lines_recv = 0;
29   while (client.available()) {
30     String line = client.readStringUntil('\n');
31     if (lines_recv == 13) {
32       Serial.println(line.substring(0, line.length()));
33     }
34     lines_recv++;
35   }
36 }

```

Gambar 5.14 Kode program mengambil data curah pada ESP8266

Pada Gambar 5.14 diatas merupakan program yang digunakan pada ESP8266, dimana program tersebut di upload ke ESP8266 menggunakan tambahan modul FTDI dan menggunakan board “Generic ESP8266 Module” pada IDE Arduino 1.8.0. Secara detail pada baris ke-1 merupakan penambahan library “ESP8266.h”, baris ke-2 dan ke-3 merupakan inialisasi nama dan password SSID, baris ke-4 merupakan host atau domain yang akan diakses, pada fungsi void setup di baris ke-5 sampai baris ke-16 terdapat proses untuk mengkoneksikan ESP8266 dengan jaringan internet melalui sambungan SSID yang didalam void setup terletak pada baris ke-11 sampai baris ke-13. Jika ESP8266 sudah terkoneksi maka akan menampilkan nomor *internet protocol* yang didapat dari SSID.

Pada fungsi void loop, baris ke-18 menunjukkan bahwa ESP8266 bekerja pada mode *client*, baris ke-19 adalah pemberian port yang digunakan pada koneksi HTTP, baris ke-20 sampai baris ke-23 merupakan kondisi jika tidak berhasil melakukan koneksi ke host menggunakan port yang ada, baris ke-24 merupakan alamat atau URL yang akan diakses untuk mengambil data pada situs, kemudian baris ke-25 adalah perintah untuk I “Get” atau pengambilan data pada URL yang ada dengan beberapa atribut yang menyertai proses *request* data curah hujan tersebut, baris ke-27 digunakan delay untuk menjeda proses cetak hasil pengambilan data, pada baris ke-29 sampai baris ke-35 digunakan untuk mencetak hasil pengambilan data ke serial monitor, data dicetak dengan perulangan sampai baris terakhir data yang diperoleh, baris ke-31 terdapat kondisi untuk menampilkan hasil pada baris ke-13 saja karena pada baris ke-13 nilai data curah hujan diperoleh, selain baris ke-13 merupakan atribut yang menyertai proses pengambilan data, baris ke-34 merupakat increment dari variabel *line\_rcv* yang berarti selalu bertambah mengikuti perulangan yang terjadi.

```

Baca_esp8266
1 void baca_data(float arraybaca[2]) {
2   while (Serial.available()) {
3     bacaChar = Serial.read();
4     readString += bacaChar;
5     delay(2);
6   }
7   if (readString.length() > 0) {
8     cekNilai = readString;
9     CHujan = readString;
10    if (cekNilai.substring(4, 5) == "m") {
11      Serial.println("-- ADA DATA --");
12      nilaiCHujan = CHujan.toFloat() + 1;
13      cekNilai = "";
14      CHujan = "";
15      readString = "";
16      arraybaca[2] = nilaiCHujan;
17    } else {
18      Serial.println("-- BELUM ADA DATA --");
19    }
20  }
21 }

```

Gambar 5.15 Kode program membaca data curah dari ESP8266

Pada Gambar 5.15 berisi tentang program untuk membaca nilai data curah hujan yang sebelumnya sudah diambil dari situs melalui ESP8266. Pembacaan data curah hujan ini dilakukan oleh Arduino Uno dimana data tersebut dijadikan masukan untuk diklasifikasikan. Baris ke-1 sampai baris ke-6 merupakan perulangan yang digunakan untuk membaca data dari ESP secara serial, awalnya setiap data disimpan di variabel "bacaChar", kemudian data digabungkan dengan menyimpannya di variabel "readString". Pada baris ke-7 merupakan kondisi jika pada variabel "readString" sudah ada data maka akan melanjutkan kondisi selanjutnya yaitu terdapat variabel "cekNilai" dan "CHujan" yang berisikan data yang sama dengan "readstring". Kemudian terdapat kondisi untuk memfilter data, dimana secara normal jika data curah hujan telah ada maka akan menampilkan nilai curah hujan seperti contoh "23 ml". Kondisi akan membaca apakah data pada karakter ke 4 terdapat karakter "m", jika kondisi tersebut terpenuhi berarti data nilai curah hujan sudah ada dan data tersebut akan disimpan di variabel "nilaiCHujan" yang ditambahkan angka 1 untuk menghindari data 0. Setelah itu isi dari variabel "cekNilai, CHujan, readString" dinormalkan kembali dengan mengkosongkan karakter yang sebelumnya ada. Data akhir pada variabel "nilaiCHujan" akan disimpan pada "arraybaca" index ke 2 untuk digunakan sebagai masukan klasifikasi.

```

Program $
1 void loop() {
2   int set_data = digitalRead(13);
3   if (set_data == HIGH) {
4     lcd.setCursor(2, 0);
5     lcd.print("Tombol Sudah");
6     lcd.setCursor(5, 1);
7     lcd.print("DITEKAN");
8     delay(1000);
9     lcd.clear();
10    baca_data(baca);
11    masukan[0] = baca[0];
12    masukan[1] = baca[1];
13    masukan[2] = baca[2];
14    nBayes(masukkan);
15    delay(1000);
16  } else {
17    baca_data(baca);
18    if (cek == 1) {
19      lcd.setCursor(4, 0);
20      lcd.print("Silahkan");
21      lcd.setCursor(2, 1);
22      lcd.print("Tekan Tombol");
23      delay(1000);
24      lcd.clear();
25    }
26  }
27 }

```

Gambar 5.16 Kode program membaca *trigger Push button*

Pada program untuk pembacaan *trigger*, jika kondisi *trigger* terpenuhi dengan menekan *push button* maka akan mengeksekusi program baris ke-3 sampai ke-15, sedangkan jika belum ada *trigger* akan memberikan peringatan untuk menekan *push button* terlebih dahulu untuk memulai klasifikasi.

### 5.2.3.2 Implementasi Kode Program *Naive Bayes*

Pada tahap implementasi kode program *Naive Bayes* akan direalisasikan penggunaan kode program untuk pengambilan keputusan prediksi ada atau tidaknya hama wereng. Diawal program perlu dilakukan ini

```
Program $
1 float pAda, pTidakAda, jTAda = 6, jAda = 7 ;
2 float dL_Ada [3][9] = {
3   {25.2438, 25.6067, 26.275, 26.4467, 26.1, 25.7714, 25.6733},
4   {87.6875, 84.8, 82.8333, 83.1333, 81.2667, 80.5714, 82.5333},
5   {13.8666, 16.0909, 9.7777, 13.8, 27.6, 16.6667, 8}
6 };
7 float dL_TidakAda [3][9] = {
8   {26.2333, 26.1846, 26.3769, 25.3466, 24.6267, 24.3},
9   {84.2, 85.5385, 83.1538, 80.4667, 82.5333, 81.645},
10  {31.1, 20, 9.125, 18, 2.5, 6}
11 };
12 float hasilMeanStd[2][3];
13 float s [3] = {0}, jData[3];
14 float ada [2][3];
15 float tidak_ada [2][3];
16 float gaussian[2][3];
17 float hasil[2];
18 float tertinggi = -1.000; //set default
19 int index = 0;
20 int gauske = 0;
```

**Gambar 5.17** Kode program inisialisasi variabel perhitungan *Naive Bayes*

Pada Gambar 5.17 diatas merupakan proses inisialisasi variabel beserta tipe data yang digunakan pada perhitungan *Naive Bayes* . Inisialisasi variabel menggunakan variabel bebas agar lebih memudahkan dalam implementasi keseluruhan sistem. Pada baris ke-1 adalah inisialisasi variabel untuk menghitung nilai peluang *prior* data latih sesuai dengan jenis kelas yang ada. Baris ke-2 hingga baris ke-11 adalah inisialisasi data latih dengan tipe data array 2 dimensi dari masing – masing kelas yang berisi nilai data suhu, kelembaban dan curah hujan. Variabel pada baris ke-12 dan ke-13 digunakan untuk proses perhitungan *mean* dan standar deviasi. Hasil perhitungan mean dan standar deviasi akan disimpan pada variabel baris ke-14 dan ke-15 untuk dijadikan masukkan proses perhitungan selanjutnya. Variabel pada baris ke-16 untuk menyimpan hasil perhitungan *gaussian*. Baris ke-17 untuk menyimpan hasil dari perhitungan peluang posterior.

Baris ke-18 hingga ke-20 adalah variabel untuk perhitungan lanjutan dari *naive Bayes*.

```
Program §  
1 void probPrior() {  
2     pAda = jAda / (jAda + jTAda);  
3     pTidakAda = jTAda / (jAda + jTAda);  
4 }
```

**Gambar 5.17 Kode program perhitungan fungsi ProbPrior()**

Gambar 5.17 menunjukkan implementasi program untuk menghitung nilai peluang *prior*. Pada fungsi probPior diatas, untuk menghitung peluang *prior* “ada” serangan dengan cara membagi jumlah data “ada” serangan dibagi dengan total keseluruhan data, begitu juga untuk peluang prio “tidak ada” serangan.

```
Program §  
1 void meanStd(int y, float dLatih[3][9], int z) {  
2     for (int j = 0; j < 3; j++) {  
3         for (int i = 0; i < y; i++) {  
4             jData[j] = jData[j] + dLatih[j][i];  
5         }  
6         hasilMeanStd[0][j] = jData[j] / y;  
7     }  
8     for (int j = 0; j < 3; j++) {  
9         for (int i = 0; i < y; i++) {  
10            s[j] = s[j] + (pow(dLatih[j][i] - hasilMeanStd[0][j], 2));  
11        }  
12        hasilMeanStd[1][j] = sqrt(s[j] / (y - 1));  
13        if (z == 0) {  
14            ada [0][j] = hasilMeanStd[0][j];  
15            ada [1][j] = hasilMeanStd[1][j];  
16        } else {  
17            tidak_ada [0][j] = hasilMeanStd[0][j];  
18            tidak_ada [1][j] = hasilMeanStd[1][j];  
19        }  
20        jData[j] = {0};  
21        s [j] = 0;  
22    }  
23 }
```

**Gambar 5.18 Kode program perhitungan mean dan standar deviasi**

Pada Gambar 5.18 digunakan untuk mencari nilai mean dan standar deviasi dari tiap fitur. Baris ke-2 hingga ke-7 adalah program untuk menghitung nilai mean, sedangkan baris ke-8 hingga akhir untuk menghitung standar deviasi, dimana didalamnya terdapat program untuk menyimpan perhitungan hasil mean dan standar deviasi ke variabel lain untuk diproses ke perhitungan *gaussian*.

```

sketch_dec09a$
1 void gaussian(float data_uji[3], float data_latih[2][3]) {
2   double a, b, c, d, k, l, z;
3   //mencari nilai gaussian setiap fitur
4   for (int i = 0; i < 3; i++) {
5     a = 2 * 3.14 * (pow(data_latih[1][i], 2)); //s.dev
6     b = (pow((data_uji[i] - data_latih[0][i]), 2)); //mean
7     c = (2 * pow(data_latih[1][i], 2)); //s.dev
8     d = - (b / c);
9     e = pow(2.718282, 1);
10    f = 1 / sqrt(a);
11    gaussian[gauske][i] = e * f;
12  }
13  gauske++;
14 }

```

**Gambar 5.19** Kode program perhitungan fungsi *gaussian*

Selanjutnya yaitu proses implementasi perhitungan fungsi *gaussian* yang sebelumnya pada diagram alir Gambar 5.6 telah dibahas. Pada Gambar 5.19 di baris ke-1 adalah inisialisasi fungsi *gaussian* dengan parameter data uji yaitu nilai suhu, kelembaban, curah hujan dan data latih yang terdiri dari *mean* dan standar deviasi yang telah dihitung sebelumnya. Perhitungan *gaussian* terdapat pada baris ke-4 sampai baris ke-11, dimana perhitungan yang dilakukan disesuaikan dengan **persamaan (2.3)** yang menghitung nilai *gaussian* dari pembacaan data uji yang ada.

```

Program$
1 void probPosterior(float prior, int i) {
2   for (int j = 0; j < 3; j++) {
3     if (j == 0) {
4       hasil[i] = (gaussian[i][j] * 1000); //dikali 1000
5     } else {
6       hasil[i] = hasil[i] * gaussian[i][j] * 1000;
7     }
8   }
9   hasil[i] = hasil[i] * prior;
10  Serial.println(hasil[i]);
11 }

```

**Gambar 5.20** Kode program perhitungan fungsi *probPosterior*

Pada Gambar 5.20 diatas menunjukkan implementasi kode program untuk menghitung nilai hasil peluang prediksi ada atau tidak nya hama wereng dari data uji yang ada. Pada baris ke-1 adalah inisialisasi fungsi *probPosterior* dengan parameter nilai peluang dari setiap *prior* dan nilai perhitungan *gaussian* yang dihitung sebelumnya dari setiap fitur. Kemudian pada baris ke-2 hingga baris ke-8 melakukan perkalian pada nilai *gaussian* dikalikan dengan angka 1000 yang



digunakan agar angka dibelakang koma tidak hilang. Pada baris ke 9 merupakan perkalian nilai *gaussian* setiap fitur dengan peluang *prior* sehingga didapatkan nilai peluang posterior.

```
sketch_dec09a $
1 void kesimpulan() {
2   for (int i = 0; i < 2; i++) {
3     if (i == 0) {
4       tertinggi = hasil[i];
5       index = i + 1;
6     } else if (tertinggi < hasil[i]) {
7       tertinggi = hasil[i];
8       index = i + 1;
9     }
10  }
11  if (index == 1) {
12    Serial.println("Kesimpulan = ADA WERENG");
13    lcd.setCursor(3, 1);
14    lcd.print("ADA WERENG");
15  } else {
16    Serial.println("Kesimpulan = TIDAK ADA WERENG");
17    lcd.setCursor(0, 1);
18    lcd.print("TIDAK ADA WERENG");
19  }
20  delay(3000);
21 }
```

**Gambar 5.21 Kode program pengambilan kesimpulan**

Untuk proses terakhir yaitu pengambilan kesimpulan apakah data yang diujikan memiliki peluang ada hama atau tidak ada hama wereng. Pengambilan kesimpulan dilakukan dengan membandingkan antar nilai posterior untuk dicari mana yang memiliki nilai tertinggi. Terlihat pada Gambar 5.21 baris ke-2 hingga baris ke-10 merupakan program untuk membandingkan nilai posterior dan memilih nilai tertinggi yang dijadikan hasil klasifikasi pada sistem ini. Hasil kesimpulan akan ditampilkan oleh program baris ke-11 hingga ke-19.