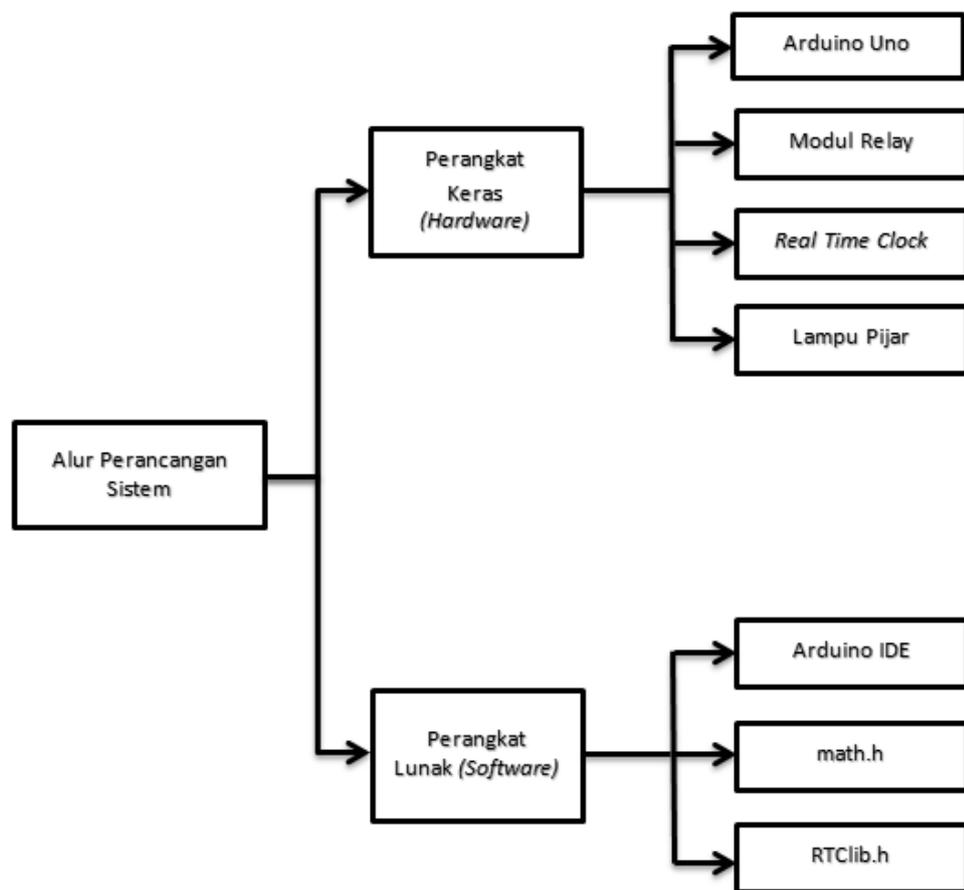


## BAB 5 PERANCANGAN DAN IMPLEMENTASI

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

### 5.1 Perancangan Sistem

Dalam tahap perancangan sistem ini dijabarkan cara perancangan yang terdapat pada sistem dimulai dari perancangan perangkat keras hingga perancangan perangkat lunak. Gambaran umum alur perancangan pada sistem ditunjukkan pada Gambar 5.1 berikut.

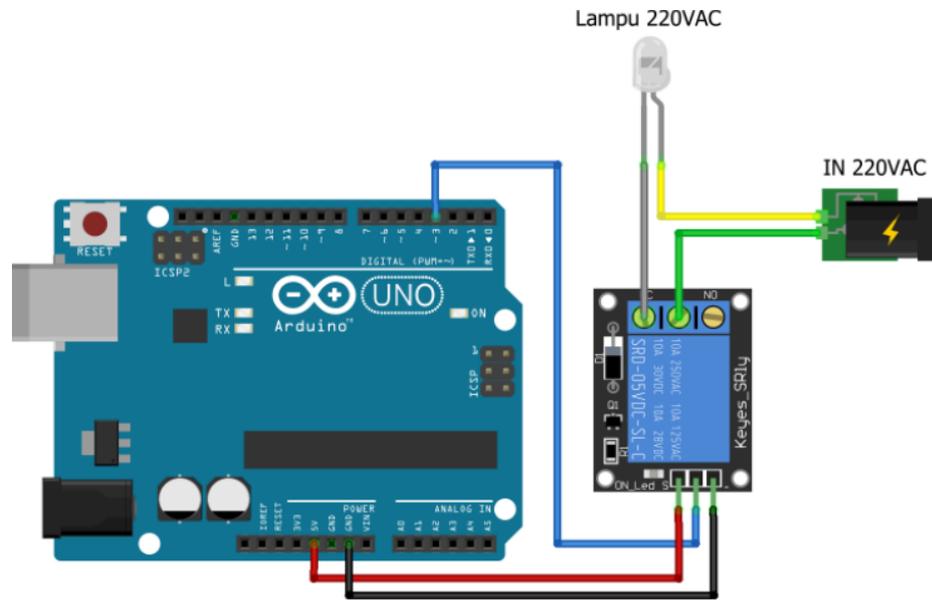


Gambar 5.1 Alur Perancangan Sistem

#### 5.1.1 Perancangan Perangkat Keras

Pada tahap perancangan perangkat keras akan dijabarkan secara detail mengenai hubungan skematik pin-pin yang digunakan antara tiap komponen perangkat keras, dimana dalam hal ini Mikrokontroler Arduino Uno yang

berfungsi sebagai pengolah data sehingga hasil olahan datanya dapat ditampilkan melalui lampu pijar yang merupakan keluaran dari sistem.

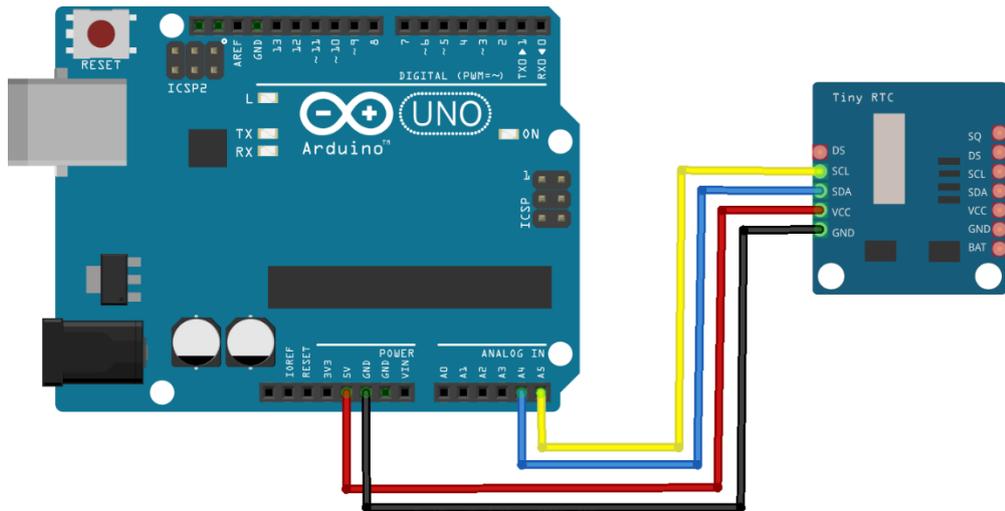


**Gambar 5.2 Rancangan Skematik Modul Relay**

**Tabel 5.1 Keterangan koneksi pin Modul Relay dengan Arduino Uno**

Pin Modul Relay	Pin Arduino Uno	Warna kabel
GND	GND	
IN	3	
VCC	5V	

Relay merupakan sebuah komponen elektronika yang berupa saklar atau switch dan dioperasikan dengan menggunakan listrik. Modul relay memiliki pin VCC, IN, dan GND. Pada Gambar 5.2 dapat dilihat bahwa kabel berwarna merah yang bertindak sebagai penghubung sumber daya tegangan antara pin VCC Modul Relay dengan pin 5V pada Arduino Uno, kabel hitam menghubungkan GND Modul Relay dengan GND yang terdapat pada Arduino Uno, dan pada pin IN menggunakan kabel biru dihubungkan dengan pin 3 pada Arduino Uno. Untuk keterangan pin pada rancangan rangkaian dari modul relay dan arduino uno secara jelas dapat dilihat pada Tabel 5.1.



**Gambar 5.3 Rancangan Skematik Real Time Clock (RTC)**

**Tabel 5.2 Keterangan koneksi pin RTC dengan Arduino Uno**

Pin RTC	Pin Arduino Uno	Warna kabel
GND	GND	■
VCC	5V	■
SDA	A4	■
SCL	A5	■

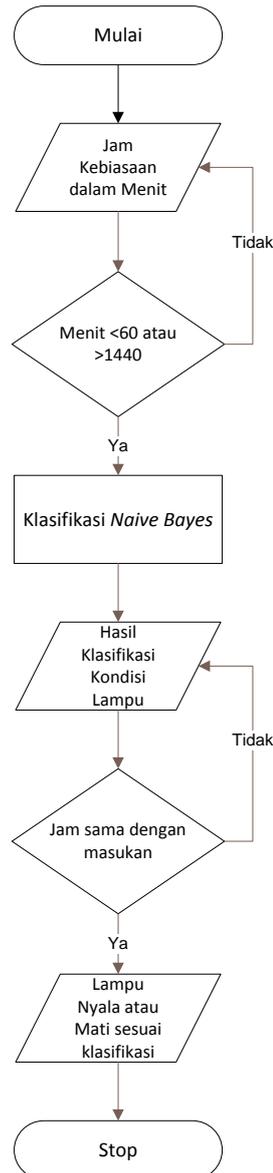
*Real Time Clock (RTC)* pada sistem ini akan digunakan sebagai jam elektronik untuk menghitung waktu secara real time sesuai dengan jam yang telah dimasukkan oleh *user*. Pada RTC memiliki 4 pin yang akan digunakan, yaitu pin GND, VCC, SDA dan SCL. Pada Pin DS sengaja tidak digunakan, karena pin tersebut berhubungan dengan sensor suhu. Berdasarkan Gambar 5.3 terlihat bahwa untuk pin GND pada RTC akan dihubungkan dengan pin GND pada Arduino Uno dan pin VCC pada RTC juga dihubungkan dengan pin VCC pada Arduino Uno. Selanjutnya, untuk pin SDA dan SCL pada RTC digunakan pin A4 dan A5 pada Arduino Uno. Pin A4 dan A5 pada Arduino Uno memiliki fungsi khusus sebagai pin I2C yang dapat digunakan, dengan pin A4 sebagai SDA dan pin A5 sebagai SCL. SDA berfungsi sebagai saluran data untuk komunikasi data antara mikrokontroler dengan RTC sedangkan SCL berfungsi sebagai saluran clock untuk komunikasi data antara Mikrokontroler dengan RTC. Untuk keterangan pin yang tersambung pada RTC dan Arduino Uno secara lebih jelas dapat dilihat pada Tabel 5.2.

## 5.1.2 Perancangan Perangkat Lunak

Pada subbab perancangan perangkat lunak akan dijabarkan secara lengkap perancangan perangkat lunak untuk melakukan proses klasifikasi dengan metode *Naive Bayes*.

### 5.1.2.1 Perancangan Program Arduino IDE

Pada perancangan program di Arduino IDE, terdapat tahapan yang harus dilakukan seperti yang terlihat pada Gambar 5.4.



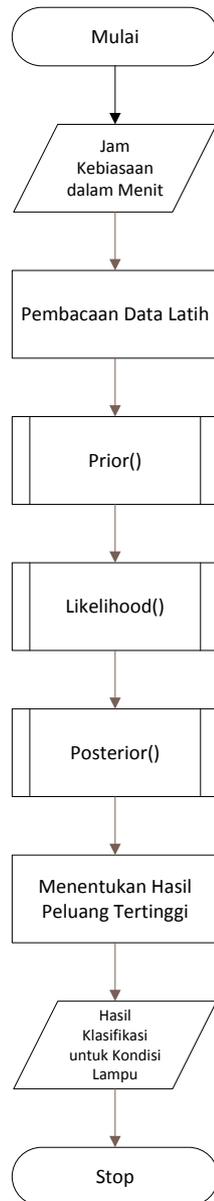
**Gambar 5.4 Diagram Alir Perancangan Program Arduino IDE**

Program pada Arduino IDE memiliki proses awal dengan memasukkan waktu atau jam dari *user* yang ingin di klasifikasikan melalui serial monitor. Proses yang dilakukan selanjutnya adalah pemilihan kondisi, apabila menit yang dimasukkan kurang dari 60 atau lebih dari 1440 sistem akan melanjutkan proses ke tahap

klasifikasi menggunakan metode *Naive Bayes*. Setelah hasil dari klasifikasi *Naive Bayes* didapat, tahap selanjutnya adalah mengecek jam yang telah dimasukkan oleh *user* sebelumnya dengan jam sekarang. Apabila jam tersebut sama, maka hasil dari klasifikasi tersebut akan di jalankan pada sebuah lampu pijar dan jika jam tersebut belum sama sistem akan terus melakukan *looping* sampai jam tersebut sama.

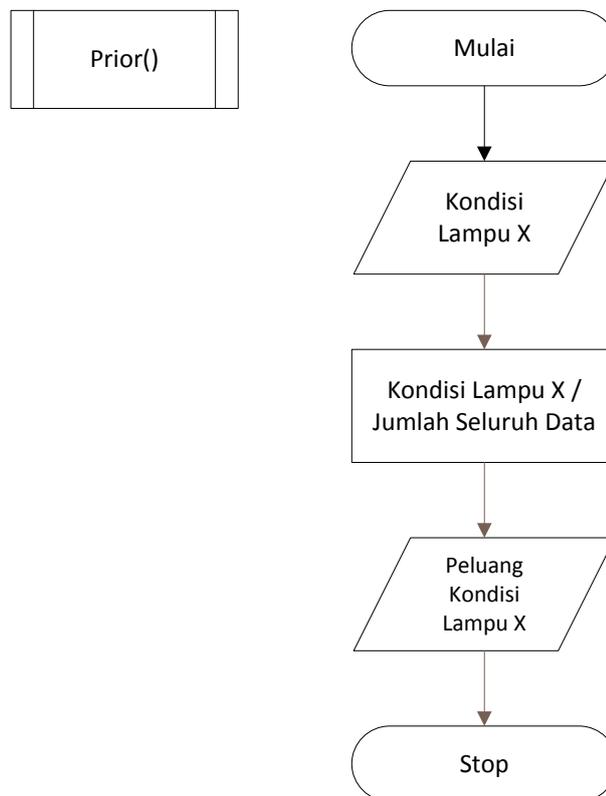
### 5.1.2.2 Perancangan Klasifikasi Naive Bayes

Dalam melakukan proses klasifikasi menggunakan metode Naive Bayes, terdapat tahapan yang akan dilakukan. Tahapan klasifikasi dapat dilihat pada Gambar 5.5.



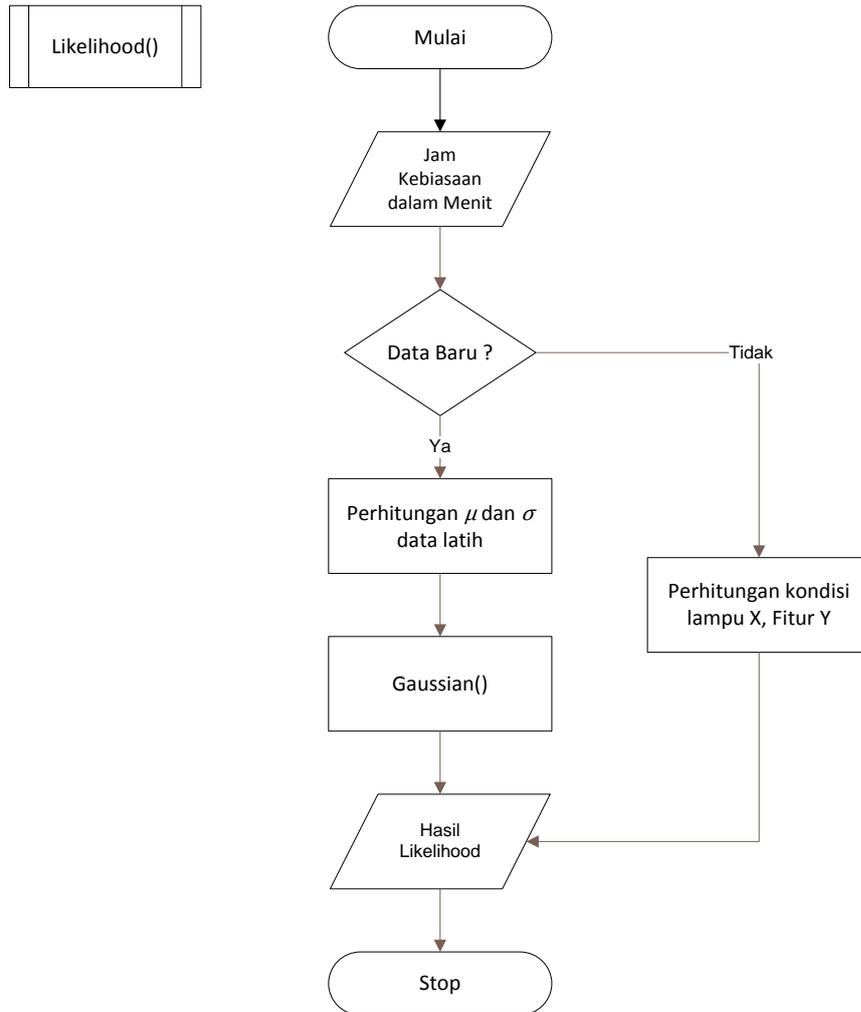
**Gambar 5.5 Diagram Alir Perancangan Klasifikasi *Naive Bayes***

Klasifikasi menggunakan metode Naive Bayes memiliki proses awal dengan memasukkan waktu atau jam dari user yang ingin di klasifikasikan. Jam inilah nantinya yang akan menjadi fitur dan dilakukan penentuan klasifikasi status pada lampu, selain itu hasil dari klasifikasi juga dipengaruhi oleh nilai pada data latih. Selanjutnya, proses untuk klasifikasi dilanjutkan dari menentukan hasil fungsi Prior(), menentukan hasil dari dari fungsi Likelihood(), menentukan hasil dari fungsi Posterior(), menentukan hasil dari peluang tertinggi hingga didapatkan hasil klasifikasi untuk status pada lampu. Untuk penjelasan dari masing-masing fungsi yang telah ditunjukkan, selanjutnya akan dibahas sesuai dengan beberapa diagram alir berikut ini.



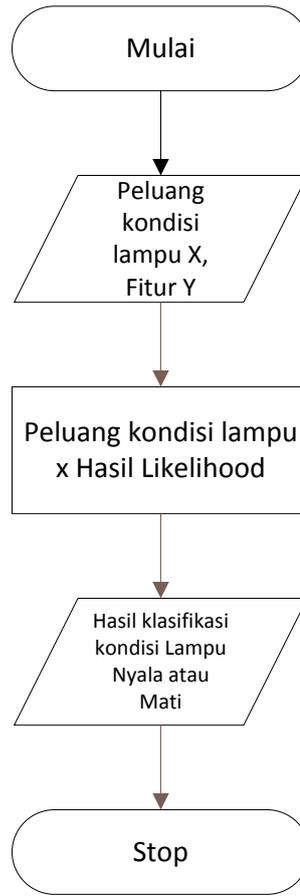
**Gambar 5.6 Diagram Alir Fungsi Probabilitas Prior**

Tahap pertama yang dilakukan dalam mengklasifikasikan jam kebiasaan terhadap kondisi lampu dengan metode Naive Bayes adalah menghitung nilai prior dari masing-masing kelas. Nilai prior merupakan nilai peluang terjadinya suatu kelas dengan cara membagi banyaknya data dalam suatu kelas dengan jumlah keseluruhan data yang ada. Data yang digunakan untuk melakukan perhitungan nilai prior merupakan data yang terdapat pada data latih.



**Gambar 5.7 Diagram Alir Fungsi Likelihood**

Tahap kedua yaitu menentukan nilai peluang dari masing-masing fitur. Dalam sistem ini, digunakan 1 fitur untuk melakukan klasifikasi terhadap kondisi lampu. Pada Gambar 5.7 terlihat masukan oleh user akan dibandingkan dengan data latih. Untuk data masukan berupa fitur yang sama dengan data yang terdapat di data latih, proses yang dilakukan adalah dengan mengkalikan jumlah fitur yang sama pada data latih tersebut dengan kondisi lampu X. Apabila masukan tersebut merupakan data yang baru dan belum terdapat pada data latih, proses yang dilakukan untuk melakukan perhitungan menentukan nilai peluang dari masing-masing fitur adalah menggunakan distribusi Gaussian. Sebelum dapat menentukan nilai peluang dari masing-masing fitur, terlebih dahulu akan dilakukan perhitungan *mean* dan standar deviasi dari data latih. Pada sistem ini, data latih yang berupa nilai mean dan standar deviasi telah langsung disimpan pada pemrograman Arduino IDE, supaya mempermudah sistem dalam melakukan proses klasifikasi. Tahap selanjutnya, akan dilakukan perhitungan menggunakan distribusi Gaussian.



**Gambar 5.8 Diagram Alir Fungsi Probabilitas Posterior**

Proses selanjutnya, yaitu mencari nilai dari peluang posterior, peluang posterior adalah peluang untuk menentukan besarnya peluang dari masing-masing kelas. Pada sistem ini untuk menentukan besarnya peluang dari masing-masing kondisi lampu akan terjadi ketika adanya masukan fitur oleh pengguna. Fitur dalam hal ini berupa waktu atau jam kebiasaan pengguna ketika menyalakan atau mematikan lampu yang telah dirubah kedalam menit. Prosesnya yaitu dengan melakukan perkalian antara hasil dari fungsi Prior() dengan fungsi Likelihood().

Tahap akhir dalam pengklasifikasian waktu atau jam kebiasaan pengguna dengan metode Naive Bayes ini adalah dengan membandingkan hasil nilai satu sama lain antar peluang posterior. Kondisi lampu dengan nilai peluang posterior paling tinggi merupakan hasil yang klasifikasi kondisi lampu yang akan dijalankan oleh sistem, dengan cara mematikan atau menyalakan lampu pijar.

Contoh perhitungan manual untuk melakukan klasifikasi terhadap kondisi lampu dengan data uji yang mempunyai fitur jam 06.00 (300 menit) berdasarkan data latih pada LAMPIRAN adalah sebagai berikut :

1. Menghitung peluang prior dari masing masing jenis kelas

$$P_{\text{Nyala}} = \frac{\text{Jumlah kelas dari kondisi lampu nyala}}{\text{Jumlah seluruh data}} = \frac{20}{40} = 0,5$$

$$P_{\text{Mati}} = \frac{\text{Jumlah kelas dari kondisi lampu mati}}{\text{Jumlah seluruh data}} = \frac{20}{40} = 0,5$$

Langkah pertama yang dilakukan dalam perhitungan manual ini adalah mencari nilai prior dari masing-masing kelas. Kelas yang akan digunakan pada sistem ini berjumlah 2 kelas, yaitu kelas lampu nyala dan lampu mati. Langkah untuk mencari nilai prior dari masing-masing kelas ini adalah dengan cara membagi seluruh data dari kelas nyala dengan jumlah dari seluruh data.

2. Menghitung nilai *Mean* dan *Standar Deviasi* dari masing masing kelas

- *Mean*

$$\begin{aligned}\bar{X}_{\text{Jam(Nyala)}} &= \frac{\sum_{i=1}^n \text{Jam}_i}{\text{Jumlah jenis kelas nyala}} \\ &= \frac{1026 + 908 + 1044 + 1216 + \dots + 1036 + 1145}{20} = 988,85\end{aligned}$$

$$\begin{aligned}\bar{X}_{\text{Jam(Mati)}} &= \frac{\sum_{i=1}^n \text{Jam}_i}{\text{Jumlah jenis kelas mati}} \\ &= \frac{310 + 992 + 285 + 1008 + \dots + 1037 + 345}{20} = 534,3\end{aligned}$$

Langkah selanjutnya yang dilakukan adalah menghitung nilai rata-rata dan *Standar Deviasi* dari masing masing kelas dan fitur atau atribut. Untuk fitur atau atribut, akan digunakan sebanyak 1 fitur atau atribut dalam melakukan klasifikasi. Perhitungan manual dalam mencari nilai rata-rata tiap kelas dilakukan dengan menjumlahkan seluruh nilai atribut setiap kelas dan membaginya dengan seluruh data dari masing-masing kelas.

- Standar Deviasi

$$\begin{aligned}\sigma_{\text{Jam(Nyala)}} &= \sqrt{\frac{\sum_{i=1}^n (\text{Jam}_i - \bar{x}_{\text{Jam(Nyala)}})^2}{\text{Jumlah jenis kelas nyala} - 1}} \\ &= \sqrt{\frac{(1026-988,85)^2+(908-988,85)^2+(1044-988,85)^2+(1216-988,85)^2+\dots+(1036-988,85)^2+(1145-988,85)^2}{25 - 1}} \\ &= 210,7063996\end{aligned}$$

$$\begin{aligned}\sigma_{\text{Jam(Mati)}} &= \sqrt{\frac{\sum_{i=1}^n (\text{Jam}_i - \bar{x}_{\text{Jam(Mati)}})^2}{\text{Jumlah jenis kelas mati} - 1}} \\ &= \sqrt{\frac{(310-534,3)^2+(992-534,3)^2+(285-534,3)^2+(1008-534,3)^2+\dots+(1037-534,3)^2+(345-534,3)^2}{25 - 1}} \\ &= 298,2773524\end{aligned}$$

3. Menghitung nilai *Likelihood* dari fitur yang diujikan (300 menit)

$$\begin{aligned}
 P(\text{Jam} = 300 | \text{Nyala}) &= \frac{1}{\sqrt{2\pi\sigma_{\text{Jam}(\text{Nyala})}^2}} e^{-\frac{(300 - \mu_{\text{Jam}(\text{Nyala})})^2}{2\sigma_{\text{Jam}(\text{Nyala})}^2}} \\
 &= \frac{1}{\sqrt{2 \times 3,14 \times (210,7063996)^2}} e^{-\frac{(300 - 988,85)^2}{2 \times 210,7063996^2}} \\
 &= 0,00000905
 \end{aligned}$$

$$\begin{aligned}
 P(\text{Jam} = 300 | \text{Mati}) &= \frac{1}{\sqrt{2\pi\sigma_{\text{Jam}(\text{Mati})}^2}} e^{-\frac{(300 - \mu_{\text{Jam}(\text{Mati})})^2}{2\sigma_{\text{Jam}(\text{Mati})}^2}} \\
 &= \frac{1}{\sqrt{2 \times 3,14 \times (298,2773524)^2}} e^{-\frac{(300 - 534,3)^2}{2 \times 298,2773524^2}} \\
 &= 0,000983
 \end{aligned}$$

4. Menghitung nilai peluang posterior dari masing-masing kelas

$$\begin{aligned}
 P(\text{Nyala} | \text{Jam}_{=300}) &= P_{\text{Nyala}} \times P(\text{Jam} = 300 | \text{Nyala}) \\
 &= 0,5 \times 0,00000905 \\
 &= 0,000004525
 \end{aligned}$$

$$\begin{aligned}
 P(\text{Mati} | \text{Jam}_{=300}) &= P_{\text{Mati}} \times P(\text{Jam} = 300 | \text{Mati}) \\
 &= 0,5 \times 0,000983 \\
 &= 0,0004915
 \end{aligned}$$

Terlihat pada hasil perhitungan peluang posterior diatas, status atau kondisi lampu yang mempunyai nilai peluang tertinggi adalah dengan **Kondisi Mati**, sehingga dapat disimpulkan bahwa kebiasaan pengguna ketika jam 06.00 merupakan kebiasaan untuk mematikan lampu.

## 5.2 Implementasi Sistem

Implementasi sistem merupakan tahap untuk merealisasikan pembuatan sistem berdasarkan semua perancangan yang telah dilakukan sebelumnya. Pada subbab ini menjelaskan secara terperinci mengenai implementasi prototype sistem dan implementasi perangkat lunak.

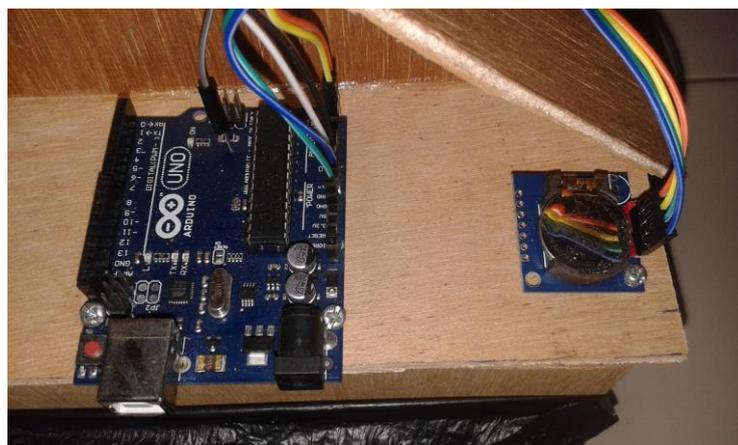
### 5.2.1 Implementasi Perangkat Keras

Tahap implementasi perangkat keras ini disesuaikan dengan perancangan yang dilakukan sebelumnya. Pada Gambar 5.9 dapat dilihat untuk tampilan dari implementasi perangkat keras antara modul relay dengan Arduino Uno. Dari gambar tersebut dapat terlihat bahwa pin VCC, IN dan GND pada modul relay telah disambungkan dengan pin VCC, pin 3 dan GND pada Arduino Uno.



**Gambar 5.9 Implementasi Modul Relay**

Pada Gambar 5.10 dapat dilihat untuk tampilan dari implementasi perangkat keras antara RTC dengan Arduino Uno. Dari gambar tersebut dapat terlihat, bahwa Pin pada RTC dan Arduino Uno telah dihubungkan satu sama lain berdasarkan perancangan yang telah dilakukan sebelumnya.



**Gambar 5.10 Implementasi *Real Time Clock* (RTC)**

Rangkaian tersebut telah disesuaikan dengan perancangan perangkat keras yang telah dijelaskan pada subbab 5.1.1. Dimana rangkaian elektronika saling terhubung membentuk sistem otomatisasi lampu terhadap kebiasaan. Untuk mengolah data yang telah dimasukkan oleh user, digunakan Mikrokontroler Arduino Uno dan selanjutnya akan dilakukan proses klasifikasi dan hasil

klasifikasi tersebut akan ditampilkan pada sebuah lampu pijar yang akan nyala atau mati sesuai dengan waktu yang telah dimasukkan oleh pengguna.

### 5.2.2 Implementasi Perangkat Lunak

Pada tahap ini akan menjelaskan bagaimana proses dalam menerapkan program terhadap sistem otomatisasi lampu sesuai dengan apa yang telah dirancang sebelumnya. Dalam implementasi ini, software Arduino IDE 1.6.6 menjadi program utama yang akan mengupload kepada mikrokontroler arduino. Pada Tabel 5.3 ditunjukkan tahap awal program dengan melakukan inisialisasi variabel dan juga library yang dibutuhkan untuk mempermudah pemrograman pada beberapa fungsi tertentu. Pengimplementasian *library* pada sistem ini, diantaranya adalah library “RTClib.h” untuk penggunaan *Real Time Clock* dan library “math.h” untuk melakukan perhitungan matematika. Selanjutnya, pada baris ke-4 sampai baris ke-11 merupakan kode inisialisasi untuk RTC, pin pada Arduino Uno dan variabel yang akan digunakan nantinya. Kode inisialisasi dibawah merupakan sebagai variabel global sehingga semua fungsi dari program dapat menggunakan variabel tersebut.

**Tabel 5.3 Kode pemrograman inisialisasi *library***

Baris	Kode program
1	#include <math.h>
2	#include "RTClib.h"
3	
4	RTC_DS1307 rtc;
5	
6	float pjln, pjlm;
7	float gaussian1, gaussian2;
8	float lampunyaala, lampumati;
9	
10	int menit, result, masukan;
11	int pinInput = 3;
12	

Pada tabel 5.4 dalam void setup perlu deklarasikan “Serial.begin(9600)”. Hal itu berguna untuk menentukan kecepatan pengiriman dan penerimaan data melalui port serial. Selanjutnya, “pinMode(pinInput, OUTPUT)” yang berguna untuk mengkonfigurasi pin pada variabel “pinInput” agar bertindak sebagai output. Dalam hal ini output yang akan digunakan adalah sebuah lampu pijar. Kemudian pada RTC dengan menuliskan “rtc.begin()” yang berfungsi untuk menginisialisasikan dari RTC tersebut.

**Tabel 5.4 Kode pemrograman untuk menjalankan keseluruhan sistem**

Baris	Kode program
1	void setup() {
2	
3	Serial.begin(9600);
4	pinMode(pinInput, OUTPUT);
5	rtc.begin();
6	
7	Serial.println("==> Masukkan Jam (Dalam Menit).");
8	}

### 5.2.2.1 Implementasi Kode Program Masukan User

Dalam melakukan implementasi untuk masukan user menggunakan serial monitor digunakan fungsi *Serial.parseInt()*. Fungsi tersebut berguna untuk mengambil data integer pertama di buffer serial. Tabel 5.4 menunjukkan pada baris ke-1 untuk melakukan perulangan berdasarkan suatu kondisi, baris ke-3 untuk mengambil data integer buffer serial, baris ke-6 hingga baris ke-7 untuk menampilkan perintah masukkan jam.

**Tabel 5.5 Kode pemrograman Masukan User**

Baris	Kode program
1	<code>while(Serial.available() &gt; 0)</code>
2	<code>{</code>
3	<code>  masukan = Serial.parseInt();</code>
4	<code>  naivebayes(masukan);</code>
5	<code>  {</code>
6	<code>    Serial.println("");</code>
7	<code>    Serial.println("==&gt; Masukkan Jam (Dalam Menit).");</code>
8	<code>  }</code>
9	<code>}</code>

### 5.2.2.2 Implementasi Kode Program Naive Bayes

Tahap pertama pada metode Naive Bayes adalah menentukan nilai peluang prior masing-masing kondisi lampu seperti yang telah dijelaskan pada diagram alir Gambar 5.5. Kode program pada tabel 5.4 menunjukkan baris ke-1 sampai ke-3 menerangkan inisialisasi jumlah keseluruhan data latih dan jumlah data pada masing-masing kondisi lampu. Jumlah data latih yang digunakan pada sistem sebanyak 40 data. Untuk baris ke-6 sampai ke-7 melakukan perhitungan untuk menentukan nilai peluang prior masing-masing kondisi lampu. Pada baris ke-8 sampai ke-12 menerangkan suatu variabel untuk nilai *mean* dan nilai standar deviasi dari masing-masing kondisi lampu yang nantinya nilai ini dapat digunakan untuk melakukan perhitungan *Likelihood*. Kode inisialisasi dibawah merupakan sebagai variabel global sehingga semua fungsi dari program dapat menggunakan variabel tersebut.

**Tabel 5.6 Kode pemrograman variabel nilai *prior*, *mean* & *standar deviasi***

Baris	Kode program
1	<code>float jcd = 40; //jumlah class/data</code>
2	<code>float lpanyala = 20; // jumlah data dengan lampu nyala</code>
3	<code>float lpmati = 20; // jumlah data dengan lampu mati</code>
4	
5	<code>float pNyala = lpanyala / jcd;</code>
6	<code>float pMati = lpmati / jcd;</code>
7	
8	<code>float meannyala= 988.85;</code>
9	<code>float meanmati = 534.3;</code>
10	
11	<code>float stdevnyala = 210.7063996;</code>
12	<code>float stdevmati = 298.2773524;</code>
13	

Berdasarkan Tabel 5.5 yang menunjukkan proses untuk klasifikasi jam kebiasaan pada user, dapat diketahui bahwa untuk proses pemrograman klasifikasi ini memiliki proses awal di baris ke-2 untuk inialisasi variabel untuk perhitungan pada Gaussian. Dari baris ke-4 sampai baris ke-35 dalam tabel tersebut merupakan proses pemilihan suatu kondisi yang terjadi dalam masukan yang dilakukan oleh pengguna dan juga perhitungan mencari peluang untuk setiap kelas kondisi lampu. Apabila masukan oleh pengguna merupakan data yang telah ada di data latih, akan dilakukan perhitungan pada baris tersebut, dengan cara mengkalikan jumlah fitur yang sama dengan masukan pengguna tersebut dengan setiap kondisi lampu. Selanjutnya, seluruh peluang setiap atribut tersebut akan dikalikan dengan masing-masing atribut yang memiliki status atau kondisi lampu setara.

**Tabel 5.7 Kode pemrograman untuk klasifikasi *Naive Bayes***

Baris	Kode program
1	void naivebayes (int waktu) {
2	double d, e, f, g;
3	
4	if (waktu < 60) { //menit min
5	Serial.println("Input waktu dalam menit (60 - 1440)");
6	
7	} else if (waktu > 1440) { //menit max
8	Serial.println("Input waktu dalam menit (60 - 1440)");
9	
10	} else if (waktu == 393) {
11	pjln = 1 / lpnyala;
12	pjlm = 0 / lpmati;
13	lampunyala = (pNyala * pjln);
14	lampumati = (pMati * pjlm);
15	
16	} else if (waktu == 465) {
17	pjln = 1 / lpnyala;
18	pjlm = 0 / lpmati;
19	lampunyala = (pNyala * pjln);
20	lampumati = (pMati * pjlm);
21	
22	....
23	
24	} else if (waktu == 1037) {
25	pjln = 0 / lpnyala;
26	pjlm = 1 / lpmati;
27	lampunyala = (pNyala * pjln);
28	lampumati = (pMati * pjlm);
29	
30	} else if (waktu == 1041) {
31	pjln = 0 / lpnyala;
32	pjlm = 1 / lpmati;
33	lampunyala = (pNyala * pjln);
34	lampumati = (pMati * pjlm);
35	
36	} else {
37	{ d = sqrt (2 * 3.14 * (pow(stdevnyala, 2)));
38	e = -((pow((waktu - meannyala), 2)) / (2 *
39	pow(stdevnyala, 2)));
40	f = pow(2.718282, e) * 1000; // dikali 1000
40	pembulatan
41	g = 1 / d * 1000; // dikali 1000 pembulatan
41	gaussian1 = f * g;
42	lampunyala = (pNyala * gaussian1);

Baris	Kode program
43	Serial.print("Hasil Likelihood 1 = ");
44	Serial.println(gaussian1);
45	Serial.print("Hasil Peluang Nyala = ");
46	Serial.println(lampunya);
47	}
48	{ d = sqrt (2 * 3.14 * (pow(stdevmati, 2)));
59	e = -((pow((waktu - meanmati), 2)) / (2 * pow(stdevmati, 2)));
60	f = pow(2.718282, e) * 1000; // dikali 1000 pembulatan
61	g = 1 / d * 1000; // dikali 1000 pembulatan
62	gaussian2 = f * g;
63	lampumati = (pMati * gaussian2);
64	
65	Serial.print("Hasil Likelihood 2 = ");
66	Serial.println(gaussian2);
67	Serial.print("Hasil Peluang Mati = ");
68	Serial.println(lampumati);
69	}
70	}

Tahap selanjutnya apabila data tersebut tergolong data yang baru dan belum terdapat pada data latih, akan dilakukan perhitungan peluang jam kebiasaan terhadap setiap kondisi lampu menggunakan distribusi Gaussian. Baris ke-37 sampai baris ke-59 merupakan proses perhitungan gaussian berdasarkan Persamaan (2.3) secara berurutan. Mulai baris ke-37 sampai baris ke-47 merupakan Gaussian untuk status kondisi lampu nyala, sedangkan baris ke-49 sampai baris ke-59 merupakan Gaussian untuk status kondisi lampu mati. Dari hasil yang telah diperoleh disetiap variabel, akan dibandingkan dengan variabel yang lain.

**Tabel 5.8 Kode pemrograman untuk menentukan hasil klasifikasi**

Baris	Kode program
1	if (lampumati > lampunya) {
2	Serial.println("Lampu Mati");
3	} else if (lampunya > lampumati) {
4	Serial.println("Lampu Nyala");
5	} else {
6	Serial.println("Error");
7	}
8	}
9	

Tabel 5.8 merupakan perbandingan disetiap variabel yang akan dicari nilai dari variabel mana yang merupakan variabel paling besar. Hal tersebut merupakan program tahap akhir dari keseluruhan sistem otomatisasi lampu menggunakan *naive bayes*. Seperti yang dijelaskan sebelumnya bahwa untuk hasil akhir dalam pengklasifikasian adalah dengan membandingkan nilai akhir dari setiap variabel dan mencari nilai mana yang paling besar dibanding dengan yang lain. Ketika variabel "lampunya" lebih besar maka kondisi lampu tersebut adalah akan menyala, dan jika variabel "lampumati" lebih besar maka kondisi lampu tersebut adalah akan mati.

### 5.2.2.3 Implementasi Kode Program untuk mengecek Jam

Pada tabel 5.9 merupakan kode program untuk mengecek waktu sekarang dengan waktu yang telah dimasukkan oleh pengguna. Baris ke-1 sampai baris ke-4 berguna untuk mengambil waktu sekarang dan akan dirubah kedalam menit. Untuk baris ke-6 sampai baris ke-9 merupakan kode program yang berguna untuk pemilihan kondisi.

**Tabel 5.9 Kode pemrograman untuk Aktuator secara *Real Time***

Baris	Kode program
1	<code>DateTime now = rtc.now();</code>
2	
3	<code>menit = now.hour() * 60;</code>
4	<code>result = menit + now.minute();</code>
5	
6	<code>if (masukan == result &amp;&amp; lampumati &gt; lampunya){</code>
7	<code>digitalWrite(pinInput, LOW);</code>
8	<code>} else if (masukan == result &amp;&amp; lampunya &gt; lampumati){</code>
9	<code>digitalWrite(pinInput, HIGH);</code>
10	<code>}</code>
11	<code>return;</code>
12	
13	