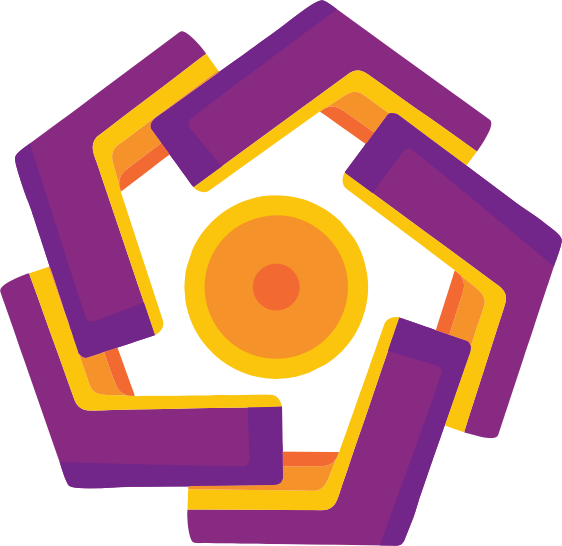
**LAPORAN FINAL PROJECT**

**PENERAPAN NAIVE BAYES PROBABILITAS DISTRIBUSI NORMAL DALAM MONITORING KEADAAN SUHU BERBASIS IOT**



**Dosen Pembimbing : Hartatik, S.T, M.Cs**

**Disusun oleh :**

1. **Mahesa Putra Baskoro Cahyo Ramadhan 19.11.2900**
2. **Leon Ellen 19.11.2907**

**FAKULTAS ILMU KOMPUTER S1 INFORMATIKA**

**UNIVERSITAS AMIKOM YOGYAKARTA**

**2021/2022**

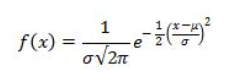
1. **Pendahuluan**

Pada era digitalisasi ini dunia teknologi informasi dan komunikasi berkembang sangat pesat di semua bidang, dampak dari berkembang pesatnya teknologi dan informasi ini berdampak pertukaran informasi antar manusia di seluruh dunia sangat cepat. Ada berbagai banyak teknologi yang ada sekarang ini seperti *Data Mining*, *Artificial Intelligence*, *Internet of Things* , *Augmented Reality* dan lain-lain. Ada pun fungsi dalam teknologi tersebut banyak salah satunya seperti memprediksi, membuat optimasi serta mengklasifikasikan suatu masalah. Dalam implementasinya teknologi-teknologi tersebut sangat beragam seperti halnya diterapkan untuk mengklasifikasikan keadaan seperti suhu dan kelembaban. Secara langsung kita sebagai manusia sangat peka terhadap suhu terlebih lagi apabila suatu suhu semakin tinggi maka untuk keadaan tubuh kita dalam meresponnya merasa semakin panas, begitu pula sebaliknya apabila suatu suhu semakin rendah maka untuk keadaan tubuh kita dalam meresponnya semakin dingin, hal ini pun berlaku juga pada suatu benda yang ada dalam kehidupan kita sehari-hari. Pada suatu kasus tertentu dalam keadaan suhu tinggi manusia dapat merespon dengan melakukan berbagai cara, akan tetapi apabila suatu benda dalam keadaan suhu tinggi tidak dapat merespon seperti halnya manusia, hal tersebut apabila dilakukan terus menerus terhadap suatu benda yang ada tanpa adanya tindakan respon manusianya, dapat menimbulkan suatu kerusakan yang berarti terhadap benda tersebut dikarenakan *overheat* terhadap temperatur suhu yang diterima. Pada kasus lainnya manusia memang dapat merespon perubahan suhu dengan berbagai cara akan tetapi terkadang manusia tidak dapat mengetahui besaran suhu yang diterimanya dan memunculkan sebuah kebingungan apabila akan melakukan suatu tindakan terhadap suatu benda apabila suhu berubah secara drastis. Melihat persoalan tersebut salah satu solusinya adalah dengan membuat sebuah alat yang dapat membaca suhu serta kelembaban dan dapat mengklasifikasikan keadaan tersebut secara otomatis sehingga dapat memberikan sebuah informasi suhu dan kelembaban kepada manusia dengan menerapkan suatu algoritma untuk membuat kepastian yang akurat dalam memberikan sebuah informasi. Algoritma yang cocok untuk diterapkan dalam alat pembaca suhu dan kelembaban tersebut adalah *Algorithm Naive Bayes Probability Normal Distribution* dikarenakan dengan algoritma tersebut dapat mengklasifikasikan keadaan dengan akurat sesuai dengan dataset yang telah dibuat. Selain dari algoritma yang akan ditanamkan pada alat, nantinya alat tersebut akan menggunakan teknologi *Internet of Things* yang berfungsi agar suatu tempat yang sudah dipasangkan alat dapat dengan mudah diambil informasi suhunya dimana saja.

1. **Tinjauan Pustaka**

*Naive bayes* adalah sebuah metode klasifikasi yang berakar teorema bayes, *Naive bayes* merupakan salah satu model klasifikasi yang sangat baik dibanding model klasifikasi lainnya hal ini dibuktikan oleh Xhemali, Hinde Stone dalam jurnalnya *“Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages”*. Suatu keuntungan penggunaan Naive bayes adalah dalam implementasinya metode ini hanya membutuhkan jumlah data training yang sedikit dalam menentukan estimasi parameter yang dibutuhkan untuk proses klasifikasinya. Tak hanya itu dalam proses penghitungannya naive bayes mudah dipahami dan mudah dibuat sehingga dalam menentukan klasifikasinya pengguna dapat mengerti bagaimana suatu keputusan itu ditetapkan untuk klasifikasi.

Dalam mekanisme pengklasifikasiannya pada naive bayes ini dengan menggunakan metode probabilitas dan statistik. Probabilitas adalah suatu kemungkinan atau pun peluang dari suatu kejadian. Dalam ilmu matematika probabilitas termasuk ke dalam teori peluang serta kemungkinan, terjadinya suatu peluang atau tidak dari suatu kejadian tersebut berpeluang untuk terjadi. Adapun jenis dari probabilitas salah satunya adalah probabilitas distribusi normal. Probabilitas Distribusi normal adalah jenis distribusi dengan variabel acak yang kontinu, distribusi normal ini pun biasanya dapat disebut juga distribusi *Gaus*, pada distribusi normal ini terdapat persamaan terkait fungsi densitas. Adapun rumus dari distribusi normal :



π : konstanta dengan nilai 3,14159. . .

e : bilangan eksponensial dengan nilai 2,7183 . . .

µ : rata-rata (mean) dari data

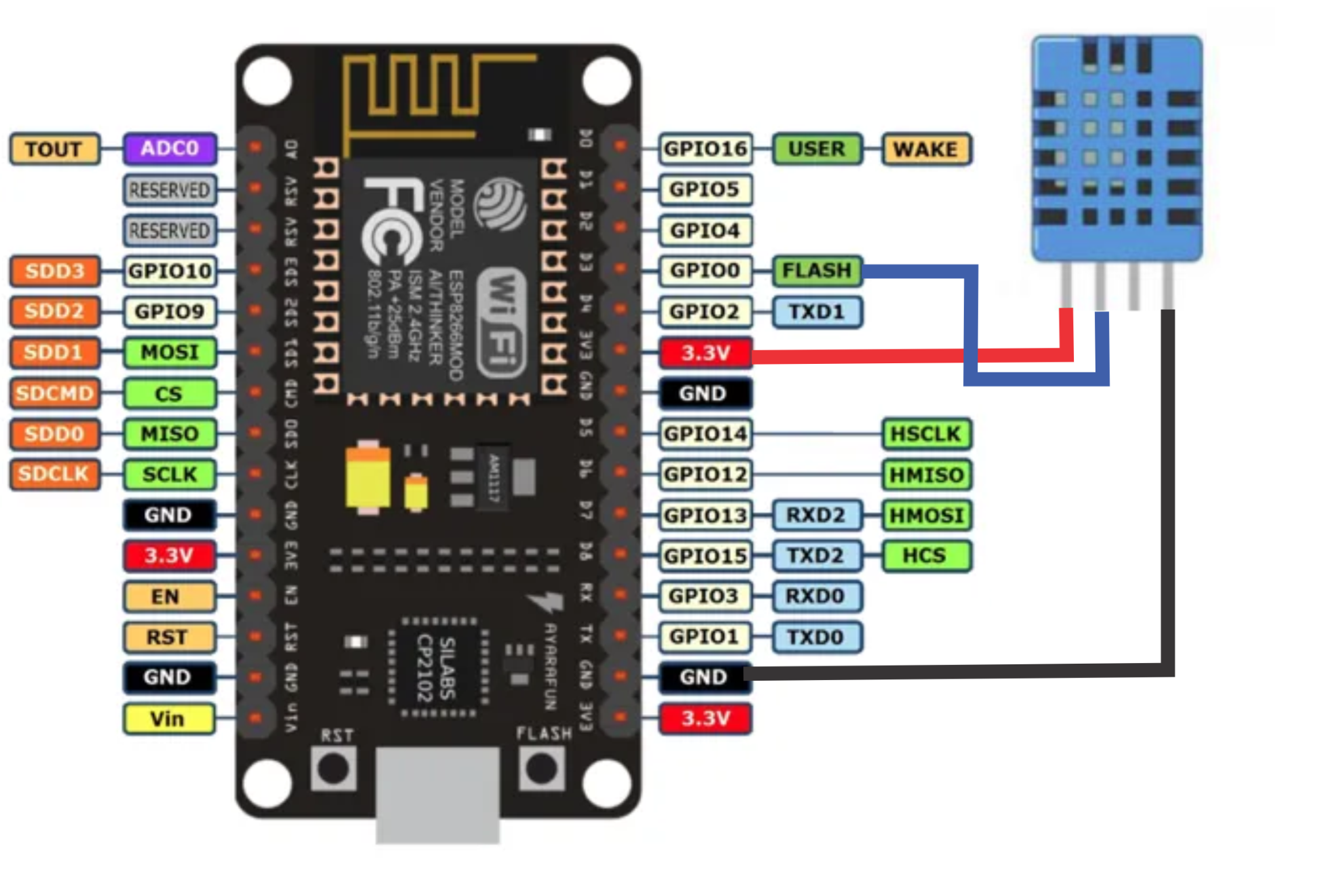
σ : simpangan baku data berdistribusi normal

1. **Proses Perancangan Alat,**

**Proses Kerja Algoritma dan Implementasi Coding**

**3.1 Perancangan Alat**

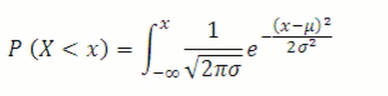
Bagian ini menjelaskan bagaimana proses perancangan Algoritma terhadap sistem alat yang akan buat. kita menggunakan Microcontroller *NodeMCU* dengan IC ESP8266-12E sebagai board kontrollernya, serta sensor DHT11 sebagai sensor untuk membaca suhu adapun wiringnya skematiknya seperti berikut ini:



Bila dilihat rangkaian diatas kita menggunakan pin D3 atau pun GPIO0 sebagai pin untuk menyalurkan datanya masuk kedalam board controller *NodeMCU*, untuk data suhu dan kelembaban yang masuk ke dalam board controller langsung di proses menggunakan algoritma naive bayes yang sudah diprogram sebelumnya menggunakan arduino ide sebagai editor dan compiler. Untuk penampilan keputusan dari algoritma naive bayesnya akan ditampilkan melalui aplikasi Blynk dengan teknologi internet of things.

**3.2 Proses kerja Algoritma**

Dalam tinjauan pustaka, kita telah menjabarkan mengenai probabilitas distribusi normal. Adapun distribusi normal pada tinjauan pustaka tersebut, disini kita memodifikasi rumus tersebut dengan menggunakan integral dikarenakan distribusi normal untuk pendistribusian datanya sangat rapat karena bersifat kontinu oleh karena itu kita menggunakan integral mulai dari min tak hingga sampai batas yang ditentukan untuk mengatasi hal ini, dalam rumus probabilitas yang ada akan seperti dibawah ini :



Untuk penerapan dari rumus tersebut kita memiliki penghitungan dari dataset yang kita miliki untuk diterapkan dalam mengklasifikasikan keadaan suatu suhu menggunakan naive bayes. Dalam implementasinya sebagai berikut :

| **Kondisi (F)** | **Suhu (X)** | **Kelembapan (Y)** | **Mean (X' or Y') =(∑𝑋 or Y)/(∑𝐹)** | |
| --- | --- | --- | --- | --- |
| **Suhu** | **Kelembapan** |
| Normal | 28.4 | 76 | 29.66666667 | 73.66666667 |
| Normal | 28.9 | 75 |
| Normal | 29.7 | 74 |
| Normal | 29.7 | 74 |
| Normal | 30.5 | 72 |
| Normal | 30.8 | 71 |
| Peringatan | 32.6 | 68 | 33.06666667 | 66.66666667 |
| Peringatan | 32.7 | 67 |
| Peringatan | 32.9 | 67 |
| Peringatan | 33.2 | 67 |
| Peringatan | 33.4 | 66 |
| Peringatan | 33.6 | 65 |
| Bahaya | 34.1 | 63 | 34.96666667 | 59 |
| Bahaya | 34.2 | 62 |
| Bahaya | 34.3 | 61 |
| Bahaya | 35.5 | 58 |
| Bahaya | 35.8 | 56 |
| Bahaya | 35.9 | 54 |

| **X-X'** | **(X-X')^2** | **Y-Y'** | **(Y-Y')^2** | **∑(X-X')^2** | **∑(Y-Y')^2** |
| --- | --- | --- | --- | --- | --- |
|
| -1.266666667 | 1.604444444 | 2.333333333 | 5.444444444 | 4.173333333 | 17.33333333 |
| -0.766666667 | 0.587777778 | 1.333333333 | 1.777777778 |
| 0.033333333 | 0.001111111 | 0.333333333 | 0.111111111 |
| 0.033333333 | 0.001111111 | 0.333333333 | 0.111111111 |
| 0.833333333 | 0.694444444 | -1.666666667 | 2.777777778 |
| 1.133333333 | 1.284444444 | -2.666666667 | 7.111111111 |
| -0.466666667 | 0.217777778 | 1.333333333 | 1.777777778 | 0.793333333 | 5.333333333 |
| -0.366666667 | 0.134444444 | 0.333333333 | 0.111111111 |
| -0.166666667 | 0.027777778 | 0.333333333 | 0.111111111 |
| 0.133333333 | 0.017777778 | 0.333333333 | 0.111111111 |
| 0.333333333 | 0.111111111 | -0.666666667 | 0.444444444 |
| 0.533333333 | 0.284444444 | -1.666666667 | 2.777777778 |
| -0.866666667 | 0.751111111 | 4 | 16 | 3.633333333 | 64 |
| -0.766666667 | 0.587777778 | 3 | 9 |
| -0.666666667 | 0.444444444 | 2 | 4 |
| 0.533333333 | 0.284444444 | -1 | 1 |
| 0.833333333 | 0.694444444 | -3 | 9 |
| 0.933333333 | 0.871111111 | -5 | 25 |

| **(∑(X-X')^2)/(N-1)** | **(∑(Y-Y')^2)/(N-1)** | **Standar Deviasi (o)** | |
| --- | --- | --- | --- |
| **√((∑(X-X')^2)/(N-1))** | **√((∑(Y-Y')^2)/(N-1))** |
| 0.834666667 | 3.466666667 | 0.913600934 | 1.861898673 |
|
|
|
|
|
| 0.158666667 | 1.066666667 | 0.398329847 | 1.032795559 |
|
|
|
|
|
| 0.726666667 | 12.8 | 0.852447457 | 3.577708764 |
|
|
|
|
|

Penghitungan Probabilitasnya :

Rumus suhu :

(1/(SQRT(2\*3.14\*”StandarDeviasiSuhu”)))\*EXP(-((“SuhuSekarang”-rata”suhu)^2))/(2\*(”StandarDeviasiSuhu”)^2)

Rumus kelembaban :

(1/(SQRT(2\*3.14\*”StandarDeviasiKelembaban”)))\*EXP(-((“KelembabanSekarang”-rata” kelembaban)^2))/(2\*(”StandarDeviasiKelembabab”)^2)

Dalam pengambilan keputusannya :

1. Jika total suhu normal > total suhu peringatan && total suhu normal> total suhu bahaya

Maka kondisi suhu normal

1. Jika total suhu peringatan > total suhu normal && total suhu peringatan > total suhu bahaya

Maka kondisi suhu peringatan

1. Jika total suhu bahaya > total suhu normal && total suhu bahaya > total suhu peringatan

Maka kondisi suhu bahaya

Jika ada sebuah data dari dataset dengan suhu 28.9 dan kelembabannya adalah 75, berada dalam kondisi apakah sekarang ini?

pSuhu normal :

(1/(SQRT(2\*3.14\*0.913600934)))\*EXP(-((28.9-29.66666667)^2))/(2\*(0.913600934)^2) = 0.1389409287

pKelembapan normal :

(1/(SQRT(2\*3.14\*1.861898673)))\*EXP(-((75-73.66666667)^2))/(2\*(1.861898673)^2) = 0.007128874964

ptotal normal =pSuhu normal+pKelembapan normal= 0.1460698037

pSuhu Peringatan:

(1/(SQRT(2\*3.14\*0.398329847)))\*EXP(-((28.9-33.06666667)^2))/(2\*(0.398329847)^2) = 0.00000005748417349

pKelembaban Peringatan :

(1/(SQRT(2\*3.14\*1.032795559)))\*EXP(-((75-66.66666667)^2))/(2\*(1.032795559)^2) = 0

ptotal Peringatan = pSuhu Peringatan+pKelembaban Peringatan = 0.00000005748417349

pSuhu Bahaya :

(1/(SQRT(2\*3.14\*0.852447457)))\*EXP(-((28.9-34.96666667)^2))/(2\*(0.852447457)^2) = 0

pKelembaban Bahaya :

(1/(SQRT(2\*3.14\*3.577708764)))\*EXP(-((75-59)^2))/(2\*(3.577708764)^2) = 0

Ptotal Bahaya = pSuhu Bahaya+pKelembaban Bahaya= 0

Dari penghitung diatas dapat kita tarik kesimpulan bahwa kondisi kasus diatas adalah dalam kondisi normal, hal ini sesuai dengan total suhu normal lebih besar dari total suhu peringatan dan total suhu bahaya.

**3.1 Implementasi Coding**

Dalam penerapan *Algoritma naive bayes* pada *NodeMCU* diperlukan penghitungan probabilitas. Adapun penghitungan probabilitas pada microcontroller kurang lebih hampir sama dengan penghitungan di excel menggunakan rumus probabilitas yang sudah ditetapkan, akan tetapi ada beberapa fungsi yang harus disesuaikan pada syntaxnya seperti fungsi akar, bila ditulis dalam arduino ide akan menjadi seperti dibawah ini :

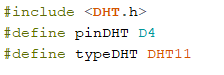
Untuk suhu :

(1/(sqrt(2\*3.14\*”Std suhu”)))\*exp(-((pow((DatasensorSuhu-(rata-rata suhu)),2)))/(2\*(pow((Std suhu),2)));

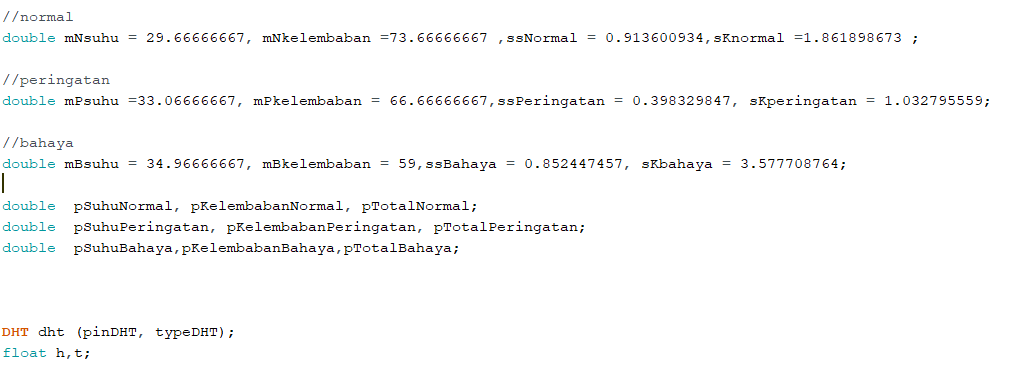
Untuk kelembaban :

(1/(sqrt(2\*3.14\*”Std kelembaban”)))\*exp(-((pow((DatasensorKelembaban-(rata-rata kelembaban)),2)))/(2\*(pow((Std kelembaban),2)));

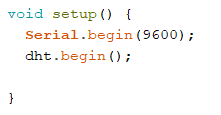
Bila diimplementasi langsung dan dijabarkan secara rinci hal pertama dalam membuat program pada nodemcu adalah dengan memasukan terlebih dahulu library sensor yang akan digunakan, adapun untuk memasukan library tersebut seperti gambar dibawah ini :



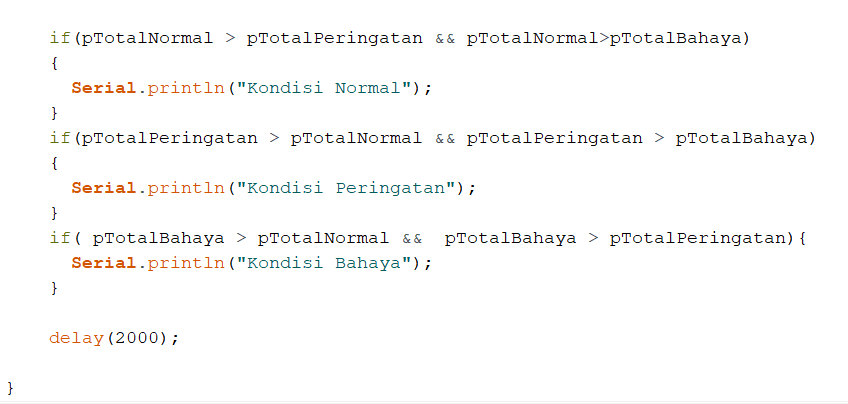
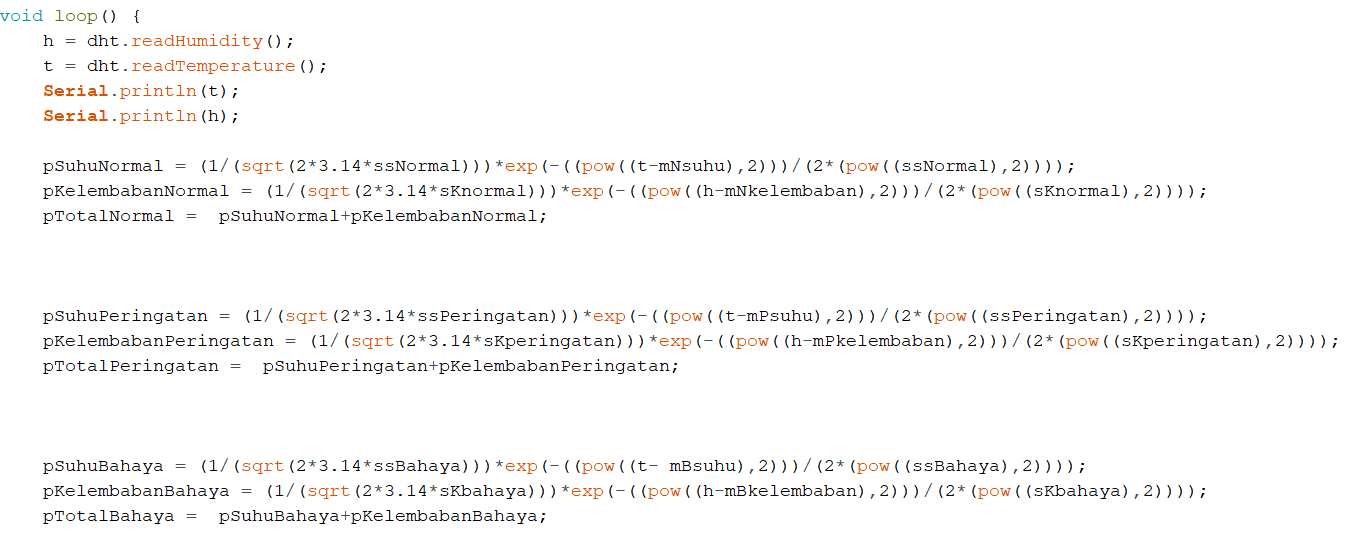
Apabila sudah memasukan library dari sensor yang akan digunakan, selanjutnya adalah dengan memasukan nilai penghitungan standar deviasi dan rata-rata dari suhu serta kelembaban dengan masing-masing kondisi yaitu kondisi suhu normal, peringatan, bahaya yang sudah dihitung sebelumnya pada tabel dataset diatas dengan menggunakan excel. Apabila hasil penghitungannya sudah dimasukan jangan lupa untuk membuat fungsi instance dari library sensornya. Adapun penjabaran yang dimaksud seperti gambar dibawah ini :



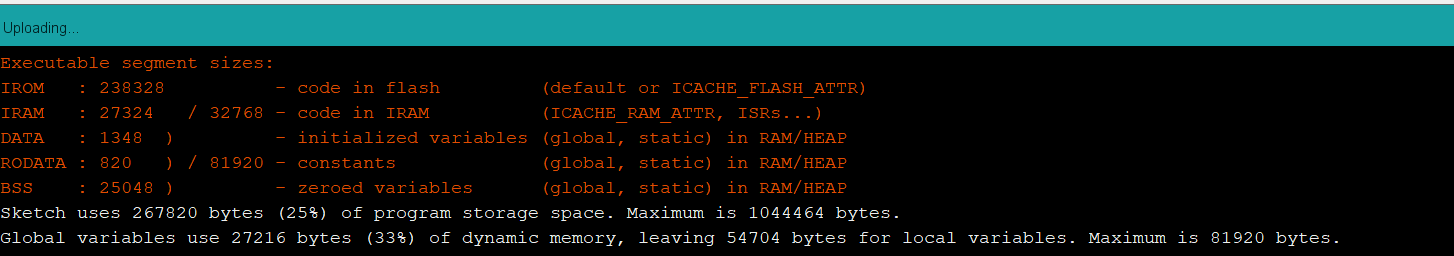
Selanjutnya adalah menguji fungsi setup sebagai inisialisasi untuk mengaktifkan serial monitor yang dimiliki oleh arduino ide yang bertujuan untuk memonitoring sementara hasil klasifikasi yang sudah diproses yang nantinya akan dikirim ke aplikasi dan pada fungsi setup ini kita akan juga akan menginisialisasikan untuk mengaktifkan library sensor yang sudah di intance sebelumnya. Adapun penjabaran yang dimaksud seperti gambar dibawah ini :



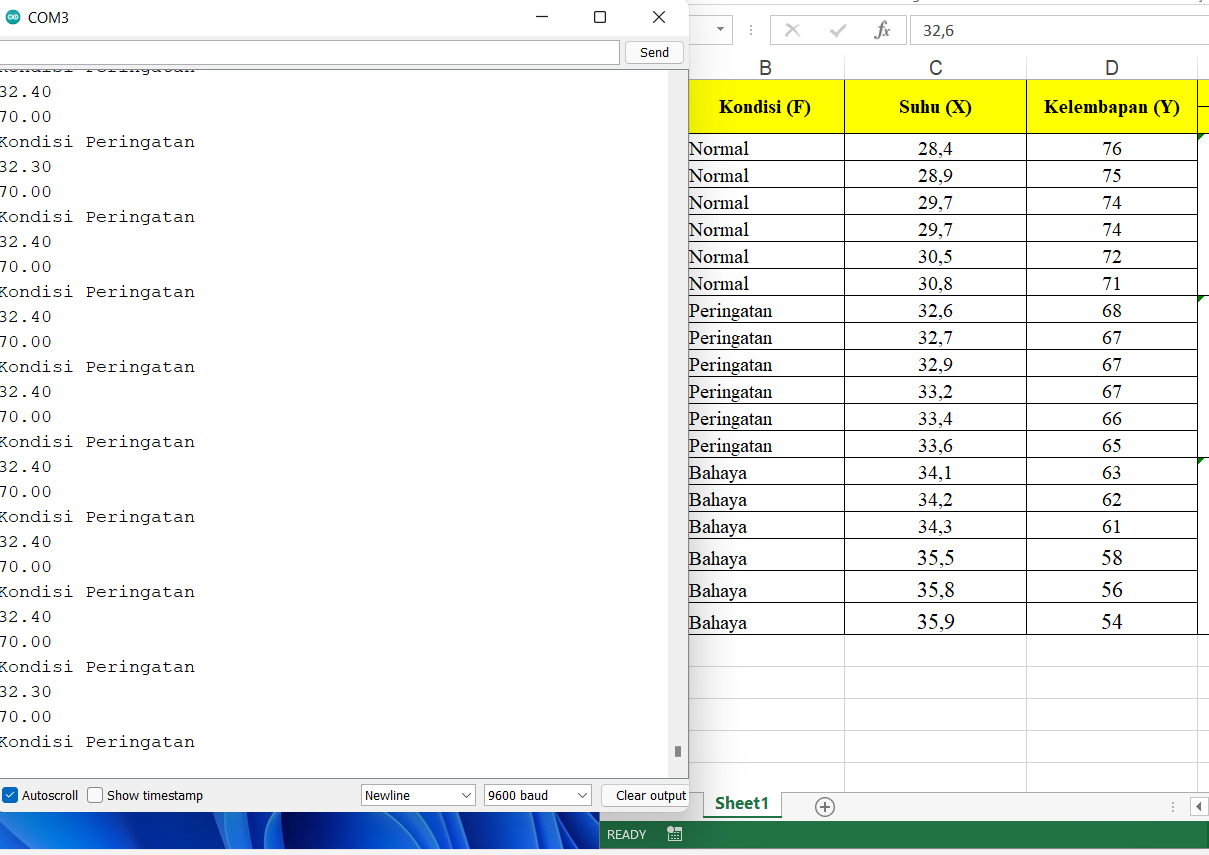
Tahapan pengisian fungsi selanjutnya adalah fungsi loop, fungsi loop ini akan terus dijalankan berulang-ulang sampai alat tersebut dimatikan. Fungsi loop ini memiliki peranan penting dalam proses pengklasifikasian dan penentuan sistem dalam mengklasifikasi. Rumus Probabilitas yang sudah dijabarkan sebelumnya akan dimasukan pada fungsi ini, serta nantinya hasil probabilitas tersebut nantinya akan kita buat sebagai penentuan sistem dengan menggunakan percabangan. Adapun penjabaran yang dimaksud seperti gambar dibawah ini :



Apabila tahap pembuatan coding sudah, selanjutnya adalah tahap pengcompilan program ke dalam nodemcu dari arduino ide. Proses ini memakan waktu agak lama, mengingat memori yang terdapat pada mikrokontroller rata-rata relatif kecil.



Bila kita melihat kedalam serial monitor dan membandingkannya dengan dataset yang sudah kami miliki, untuk hasil klasifikasi berjalan dengan baik dan akurat. Hasil perolehan klasifikasi akan diteruskan ke server blynk yang memiliki tujuan, agar dapat dimonitoring melalui handphone. Adapun hasil klasifikasinya Seperti gambar dibawah ini :



1. **Referensi**

<https://binus.ac.id/bandung/2019/12/algoritma-naive-bayes/>

<https://citec.amikom.ac.id/main/index.php/citec/article/view/49>

<https://repository.unsri.ac.id/8619/2/RAMA_56201_09011381520050_0015107201_01_front_ref.pdf>

<https://ecampus.pelitabangsa.ac.id/pb/AmbilLampiran?ref=22968&jurusan=&jenis=Item&usingId=false&download=false&clazz=ais.database.model.file.LampiranLain>

1. **Lampiran**

**5.1 Link Youtube**

<https://www.youtube.com/watch?v=Np3epzxQtks>

**5.2 Source Code**

**#include <DHT.h>**

**#define pinDHT D4**

**#define typeDHT DHT11**

**//normal**

**double mNsuhu = 29.66666667, mNkelembaban =73.66666667 ,ssNormal = 0.913600934,sKnormal =1.861898673 ;**

**//peringatan**

**double mPsuhu =33.06666667, mPkelembaban = 66.66666667,ssPeringatan = 0.398329847, sKperingatan = 1.032795559;**

**//bahaya**

**double mBsuhu = 34.96666667, mBkelembaban = 59,ssBahaya = 0.852447457, sKbahaya = 3.577708764;**

**double pSuhuNormal, pKelembabanNormal, pTotalNormal;**

**double pSuhuPeringatan, pKelembabanPeringatan, pTotalPeringatan;**

**double pSuhuBahaya,pKelembabanBahaya,pTotalBahaya;**

**DHT dht (pinDHT, typeDHT);**

**float h,t;**

**void setup() {**

**Serial.begin(9600);**

**dht.begin();**

**}**

**void loop() {**

**h = dht.readHumidity();**

**t = dht.readTemperature();**

**Serial.println(t);**

**Serial.println(h);**

**pSuhuNormal = (1/(sqrt(2\*3.14\*ssNormal)))\*exp(-((pow((t-mNsuhu),2)))/(2\*(pow((ssNormal),2))));**

**pKelembabanNormal = (1/(sqrt(2\*3.14\*sKnormal)))\*exp(-((pow((h-mNkelembaban),2)))/(2\*(pow((sKnormal),2))));**

**pTotalNormal = pSuhuNormal+pKelembabanNormal;**

**pSuhuPeringatan = (1/(sqrt(2\*3.14\*ssPeringatan)))\*exp(-((pow((t-mPsuhu),2)))/(2\*(pow((ssPeringatan),2))));**

**pKelembabanPeringatan = (1/(sqrt(2\*3.14\*sKperingatan)))\*exp(-((pow((h-mPkelembaban),2)))/(2\*(pow((sKperingatan),2))));**

**pTotalPeringatan = pSuhuPeringatan+pKelembabanPeringatan;**

**pSuhuBahaya = (1/(sqrt(2\*3.14\*ssBahaya)))\*exp(-((pow((t- mBsuhu),2)))/(2\*(pow((ssBahaya),2))));**

**pKelembabanBahaya = (1/(sqrt(2\*3.14\*sKbahaya)))\*exp(-((pow((h-mBkelembaban),2)))/(2\*(pow((sKbahaya),2))));**

**pTotalBahaya = pSuhuBahaya+pKelembabanBahaya;**

**if(pTotalNormal > pTotalPeringatan && pTotalNormal>pTotalBahaya)**

**{**

**Serial.println("Kondisi Normal");**

**}**

**if(pTotalPeringatan > pTotalNormal && pTotalPeringatan > pTotalBahaya)**

**{**

**Serial.println("Kondisi Peringatan");**

**}**

**if( pTotalBahaya > pTotalNormal && pTotalBahaya > pTotalPeringatan){**

**Serial.println("Kondisi Bahaya");**

**}**

**delay(2000);**

**}**