LAMPIRAN

1. Tabel Waktu Penelitian

No.	Kegiatan	Jan 20				Feb 20				Mar 20				Apr 20			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Survey Lapangan																
2	Penentuan Judul																
3	Studi Literatur																
4	Pengolahan Data																
5	Analisis dan Evaluasi																

6	Penyusunan Laporan															

2. Program-program

a. Arduino

```
float humiTanah;

float temp,humi;

float light;

#include "DHT.h"

#define DHTPIN A0

#define HUMIPIN A1

#define LDRPIN A2

#define DHTTYPE DHT11

#define RelayKipas 12

#define RelayLampu 11

#define RelaySemprot 10
```

```
#define ledConnect 13

#define time_upload 60000

#define time_dht 1000

#define time_adc 500

#define time_kedip 1000

#define time_connect 15000

#define time_update_setting 5000

DHT dht(DHTPIN, DHTTYPE);

unsigned long timeUpload;

unsigned long timeDht;

unsigned long timeAdc;

unsigned long timeConnect;

unsigned long timeKedip;

unsigned long timeUpdateSetting;

unsigned char flagKedip;

char d,i;
```

```
int b=0,s=0,kode=0,data=0;

int flagKipas=0,flagLampu=0,flagSemprot=0;

int lastKipas, lastLampu, lastSemprot;

int flagStatConnection;

char kata[100];

int setSuhu,setLembabTanah,setLembab,setCahaya,mode=0;

String namaTanaman="Seledri";

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

humi=random(100);

light=random(100);

pinMode(RelayLampu,1);

pinMode(RelaySemprot,1);

pinMode(RelayKipas,1);

pinMode(ledConnect,13);

dht.begin();
```

```
digitalWrite(RelaySemprot,1);delay(1000);
digitalWrite(RelaySemprot,0);

digitalWrite(RelayLampu,1); delay(1000);
digitalWrite(RelayLampu,0);

digitalWrite(RelayKipas,1);delay(1000);
digitalWrite(RelayKipas,0);

timeConnect=millis()-15000;

updateLampu();

updateSemprot();

updateKipas();

}


void loop() {

/**

** untuk membaca data DHT dijalankan setiap interfal
time_dht

**/

if(millis()-timeDht>time_dht){

timeDht=millis();
```

```
temp=dht.readTemperature();

humi=dht.readHumidity();

b++; if(b%2==0){

Serial.print("notif=");

Serial.print(flagKipas);

Serial.print(flagLampu);

Serial.print(flagSemprot);

Serial.print(setSuhu);

Serial.print('-');

Serial.print(setCahaya);

Serial.print('-');

Serial.println(setLembabTanah);

}

}

/**

** untuk membaca data ADC dijalankan setiap interfal
time_adc

**/
```

```
if(millis()-timeAdc>time_adc){
timeAdc=millis();
humiTanah=analogRead(HUMIPIN);
light=1023-analogRead(LDRPIN);

}
/**
** untuk kontrol indikator led dijalankan berdasarkan flag status koneksi
**/
if(millis()-timeConnect<time_connect){
if(flagStatConnection==1){
digitalWrite(ledConnect,1);
}
else{
if(millis()-timeKedip>time_kedip){
flagKedip^=1;
timeKedip=millis();
```

```
}
digitalWrite(ledConnect,flagKedip);
}
}
else{
digitalWrite(ledConnect,0);
}

/**
** untuk upload dijalankan dlm interval time_upload
**/
if(millis()-timeUpload>time_upload){
Serial.print("insert;");
Serial.print(namaTanaman);
Serial.print(";");
Serial.print((int)humi);
Serial.print(";");
Serial.print((int)temp);
```

```
Serial.print(";");

Serial.print((int)light);

Serial.print(";");

Serial.println((int)humiTanah);

timeUpload=millis();

}

/**

** menerima protokol data dari raspberry

**/

if(Serial.available()){

d=Serial.read();

switch(s){

case 0: s=(d=='*')?1:0; break;

case 1:kode=d;s=2;break;

case 2:s=(d=='.')?3:0; data=0;break;

case 3:

if(d>=0x30 && d<=0x39)
```

```
data=data*10+(d-0x30);

else if(d==';'){

if(kode=='K'){ flagKipas=data;}

else if(kode=='L'){ flagLampu=data;}

else if(kode=='S'){ flagSemprot=data;}

else if(kode=='t'){ setSuhu=data;}

else if(kode=='B'){ setLembabTanah=data;}

else if(kode=='b'){ setLembab=data;}

else if(kode=='c'){ setCahaya=data;}

else if(kode=='m'){ mode=data;}

else if(kode=='C'){flagStatConnection=data;}

s=0;

timeConnect=millis();

}

break;

}

}
```

```
/**
* untuk update koneksi
*/
if(millis()-timeUpdateSetting>time_update_setting){
timeUpdateSetting=millis();
updateConn(i++);
}

/**
** untuk kontrol output
**/
if(mode==0){
flagKipas=(temp>setSuhu)?1:0;
flagLampu=(light<setCahaya)?1:0;
flagSemprot=(humiTanah>setLembabTanah)?1:0;

}
if(flagKipas!=lastKipas){
lastKipas=flagKipas;
```

```
updateKipas();

}

if(flagLampu!=lastLampu){

lastLampu=flagLampu;

updateLampu();

}

if(flagSemprot!=lastSemprot){

lastSemprot=flagSemprot;

updateSemprot();

}

digitalWrite(RelayKipas,(flagKipas)?1:0);

digitalWrite(RelayLampu,(flagLampu)?1:0);

digitalWrite(RelaySemprot,(flagSemprot)?1:0);


}

void updateKipas(){

Serial.print("u;");
```

```
Serial.print(namaTanaman);

Serial.print(";kipas;");

Serial.println(flagKipas);

}

void updateLampu(){

Serial.print("u;");

Serial.print(namaTanaman);

Serial.print(";led;");

Serial.println(flagLampu);

}

void updateSemprot(){

Serial.print("u;");

Serial.print(namaTanaman);

Serial.print(";semprot;");

Serial.println(flagSemprot);

}

void updateConn(int a){
```

```
Serial.print("u;");

Serial.print(namaTanaman);

Serial.print(";conn;");

Serial.println(a);

}

float randomData(float now,float dData){

float d= random(10);

float dperubahan= random(dData);

// Serial.print("=>");

// Serial.print(dperubahan);

// Serial.print(",");

// Serial.println(d);

if(now>0){

if( d>3){

//up

now+=dperubahan;

}
```

```
else{
//down
now-=dperubahan;
}
}
else{
now+=dperubahan;
}
return(now);
}
```

b. Raspberry

```
import FirebaseClass as firebase
import SerialCOMClass as serial
from tkinter import *
import time
import threading
import re
root = Tk()
```

```
fb=firebase.firebase("/gardeniot-c4aed")

ser=serial.serial(root,9600)

buff=""

tick=0

nameTanaman=""

def parseStr(buff1):

global buff

global nameTanaman

d = buff1.split(';')

buff=""

if len(d) >= 2:

command=d[0]

if(command=="insert"):

plant=d[1]

nameTanaman=plant

humi=str(int(re.search(r'\d+', d[2]).group()))

temp=str(int(re.search(r'\d+', d[3]).group()))
```

```
light=str(int(re.search(r'\d+', d[4]).group()))

humiTanah=str(int(re.search(r'\d+', d[5]).group()))

data=fb.constructData(humiTanah,humi,temp,light)

try:

fb.insertDataPlant(data, plant)

ser.write("*C.1;")

except:

print("connection problem!")

ser.write("*C.0;")

elif command== "u":

try:

fb.editDataSetting(d[1],d[2],d[3])

ser.write("*C.1;")

except:

print("connection problem!")

ser.write("*C.0;")

tick = 0
```

```
def terminal():

global buff

if ser.isConnected() == True:

while True:

c = ser.read()

if len(c) == 0:

break

d = ord(c)

if d == 13:

buff = buff

elif d == 10:

print(buff)

parseStr(buff)

buff = ""

else:

buff = buff + "%c" % d

root.after(100, terminal)

root.after(100,terminal)
```

```
def threadTimer():
global tick
global nameTanaman
while 1:
try:
control = fb.getSetting(nameTanaman)
ser.write("*C.1;")
if (control['mode'] == '1'):
ser.write("*m.1;")
print("mode Manual")
if (control['led'] == '1'):
ser.write("*L.1;")
print("Lampu ON")
else:
ser.write("*L.0;")

if (control['semprot'] == '1'):
ser.write("*S.1;")
```

```
print("Semprot ON")

else:

ser.write("*S.0;")

if (control['kipas'] == '1'):

ser.write("*K.1;")

print("Kipas ON")

else:

ser.write("*K.0;")

else:

ser.write("*m.0;")

dataSet = (control['valueSettingHThL'])

dataSets = dataSet.split('-')

# print("dataset=" )

# print(dataSets)

ser.write("*B." + dataSets[0] + ";")

ser.write("*t." + dataSets[1] + ";")

ser.write("*b." + dataSets[2] + ";")
```

```
ser.write("*c." + dataSets[3] + ";")
except:
ser.write("*C.0;")


time.sleep(0.5)

stopEvent = threading.Event()
thread = threading.Thread(target=threadTimer, args=())
thread.start()
root.after(100,terminal)
root.mainloop()
```

c. Firebase

```
from firebase import firebase as fr
import datetime
import csv

class firebase:
def __init__(self,alamat):
self.url='https:/'+alamat+'.firebaseio.com/'
```

```
self.alamat=alamat

self.firebase=fr.FirebaseApplication(self.url, None)

self.allID=[]

self.data={

'humidity':'50',

'temperature':'20',

'light':'100',

'humidityTanah':'50',

'created_at':'14-04-2020'

}

self.dataSetting={

'semprot':'0',

'led':'0',

'kipas': '0',

'mode': '0',

'conn': '0',

'valueSettingHThL':'100-200-300-400'

}
```

```
def constructData(self,humiTanah,humi,temp,light):

tgl= str(datetime.datetime.now())

tglArray= tgl.split(' ')

data=dict(zip(('humidityTanah','humidity',
'temperature','light','created_at','clock_at'), (humiTanah,humi,
temp,light,tglArray[0],tglArray[1])))

return(data)

def insertDataPlant(self,data,nama):

result= self.firebase.post(self.alamat+'/Plants/'+nama,data)

# print(result)

def insertDataSetting(self,data,nama):

result= self.firebase.post(self.alamat+'/Setting/'+nama,data)

# print(result)

def insertDataDump(self,data):

result= self.firebase.post('/sempro_fauqi/data',data)

# print(result)
```

```
def editDataPlant(self,id,param,data):

result = self.firebase.put(self.alamat+'/Plants/'+id,
param,data)

print(result)

def editDataSetting(self,name,param,data):

idData = ""

result = self.firebase.get(self.alamat+'/Setting/', '')

if name != '':

for id in result[name]:

idData = id

if (idData == ""):

print("not found data")

return ("not found data")

else:

print(result[name][idData])

print(idData)

result = self.firebase.put(self.alamat+'/Setting/'+name+'/' +
idData, param, data)
```

```
print(result)

def deletePlant(self):

idData = ""

name= 'PakCoy'

# result = self.firebase.get('/pythonfirebaseiot/Plants', '')

result = self.firebase.delete('/pythonfirebaseiot/Plants',
'PakCoy')

print(result)

# if name != '':

# for id in result[name]:

# idData = id

#

# if (idData == ""):

# print("not found data")

# return ("not found data")

# else:

# try:

# print(result[name][idData]['created_at'])
```

```
# if result[name][idData]['created_at'] == "2020-04-29":

# result = self.firebase.delete('/pythonfirebaseiot/Plants/' +
name , idData)

# print("deleted")

# print(result)

# elif result[name][idData]["created_at"] == "2020-04-30":

#

# print(result)

# print("deleted")

# except:

# try:

# result = self.firebase.delete('/pythonfirebaseiot/Plants/' +
name, idData)

# print("geje : deleted")

# except:

# print("g iso dihapus cuk")

def tampilkanSmuaPlant(self,result):

for plantID in result:
```

```
self.allID.append(plantID)

i=0

for plant in self.allID:

print("-----------plant"+plant+"------------- ")

print("name=> " + result[plant]["name"])

print("light=> " + str(result[plant]["light"]))

print("temperature> " + str(result[plant]["temperature"]))

print("humidity> " + str(result[plant]["humidity"]))

i=i+1

def getSetting(self,name=""):

idData=""

result = self.firebase.get(self.alamat+'/Setting/',")

if name!=":

for id in result[name]:

idData=id

if(idData==""):

print("not found data")
```

```
return("not found data")

else:

print(result[name][idData])

return (result[name][idData])

def getPlant(self,name=""):

result = self.firebase.get(self.alamat+'/Plants/Seledri',")

print(result)

return(result)

def writeResultToCSV(self,result):

allID = []

allID.clear()

for data in result:

allID.append(data)

with open('./export.csv', 'w') as csvFile:

writer = csv.writer(csvFile)

i = 0

for id in allID:
```

```
# print("data" + str(result[id]))

t = ["data%d" % (i + 1), str(id),str(result[id]['light']),
str(result[id]['temperature']),str(result[id]['humidity']),str(resu
lt[id]['humidityTanah']),str(result[id]['created_at'])]

writer.writerow(t)

i = i + 1

csvFile.close()

def initData(self,nama):

self.insertDataSetting(self.dataSetting, nama)

self.insertDataPlant(self.data, nama)

def initDataSetting(self,nama):

self.insertDataSetting(self.dataSetting, nama)

def initDataPlant(self,nama):

self.insertDataPlant(self.data,nama)
```