|  |  |
| --- | --- |
| **Project Smart Hidroponik** | |
| Pendahuluan | **Hidroponik** adalah salah satu metode dalam budidaya menanam dengan memanfaatkan air tanpa menggunakan media tanah dengan menekankan pada pemenuhan kebutuhan hara nutrisi bagi tanaman. Kebutuhan air pada hidroponik lebih sedikit daripada kebutuhan air pada budidaya dengan tanah. Sedangkan **Smart Hidroponik** adalah metode hidroponik yang menggunakan teknologi Internet of Things sehingga pengguna dapat control dan monitoring sistem hidroponik melalui webserver. |
| Tujuan | 1. Menerapkan teknologi Internet of Things melalui raspberry pi, pemrograman python dan project Smart IoT hidroponik. 2. Memudahkan pemantauan suhu air dan kelembapan yang cocok bagi tanaman hidroponik melalui webserver. 3. Mengatur nyala mesin melalui webserver. |
| **Tahap Pengerjaan** | |
| Alat dan bahan | 1. Raspberry PI 4B 2. LCD HDMI 3. Module relay 4. 1 buah breadboard 5. 2 buah resistor 6. 18 buah kabel jumper 7. 1 buah sensor DHT22 8. 1 buah sensor DS18B20 9. 1 buah sensor LDR 10. 1 buah arduino uno |
| Rangkaian Kerja | Detail untuk sensor DHT22    Keterangan :  Ground sambungkan ke pin 9 raspberry  VCC sambungkan ke pin 17 raspberry  Data sambungkan ke pin 12 (GPIO 18) raspberry  Resistor 10k antara data dan VCC  Detail untuk sensor DS18B20    Keterangan :  Ground sambungkan ke pin 6 raspberry  VCC sambungkan ke pin 1 raspberry  Data sambungkan ke pin 3 (GPIO 2) raspberry  Resistor 10k antara VCC dan data  Detail untuk module relay    Keterangan :  Ground sambungkan ke pin 39 raspberry  VCC sambungkan ke pin 2 raspberry  Data sambungkan ke pin 40 (GPIO 21) raspberry  Detail untuk sensor ldr    Keterangan  Satu sisi ldr disambungkan pada tegangan 3.3V, kemudian sisi lainnya disambungkan pada output(GPIO 27)  Elco disambungkan pada ldr setelah kabel output, sisi negatif sambungkan pada ground(pin 30)  **Rangkaian pada arduino uno**    Keterangan  VCC sambungkan pada 3.3V  Ground sambungkan pada ground  Output sambungkan pada A0 |
| **Setting UP DHT22 Sensor** | |
|  | Install CircuitPython-DHT library melalui terminal  pip3 install adafruit-circuitpython-dht  sudo apt-get install libgpiod2 |
|  | Buat file python: dht22.py   |  | | --- | | # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries  # SPDX-License-Identifier: MIT  import time  import board  import adafruit\_dht  # Initial the dht device, with data pin connected to:  dhtDevice = adafruit\_dht.DHT22(board.D4)  # you can pass DHT22 use\_pulseio=False if you wouldn't like to use pulseio.  # This may be necessary on a Linux single board computer like the Raspberry Pi,  # but it will not work in CircuitPython.  # dhtDevice = adafruit\_dht.DHT22(board.D18, use\_pulseio=False)  while True:  try:  # Print the values to the serial port  temperature\_c = dhtDevice.temperature  temperature\_f = temperature\_c \* (9 / 5) + 32  humidity = dhtDevice.humidity  print(  "Temp: {:.1f} F / {:.1f} C Humidity: {}% ".format(  temperature\_f, temperature\_c, humidity  )  )  except RuntimeError as error:  # Errors happen fairly often, DHT's are hard to read, just keep going  print(error.args[0])  time.sleep(2.0)  continue  except Exception as error:  dhtDevice.exit()  raise error  time.sleep(2.0) | |
|  | Tampilan program setelah dijalankan  https://lh6.googleusercontent.com/PwV2Y5BgaNn6njpBq3hNBctkIlxo1LjWpRoRQWSpERMEo1FDEJGYuqrLcRYxDJ-AdXuNGJe7RJ0oLu_uJA1MSkoCc3yALHmZTIjebwv-uYKXi9NvcyJymlMeAHXZRVYO85eh2AqNS1yE3DSAHgLfmUM |
|  | **Setting Up Sensor DS18B20** |
|  | Aktifkan modul kernel 1-wire. Buka terminal Raspberry PI kemudian ketikkan  sudo nano /boot/config.txt  Tambahkan dtoverlay=w1-gpio,gpiopin=2 dibawah baris #Additional overlays and parameters are documented.... Ketik ctrl+x dan ctrl+y untuk keluar dari nano, jika belum tersimpan dapat menggunakan ctrl+s dan ctrl+x kemudian reboot raspberry. |
|  | Install w1thermsensor  pip3 install python3-w1thermsensor –y |
|  | Buat file python DS18B20.py |
|  | Hasil program ketika dijalankan |
| **Setting Up Sensor TDS, arduino uno, dan raspberry** | |
|  | Download arduino ide pada link berikut <https://www.arduino.cc/en/software> |
|  | Install arduino ide pada laptop, kemudian buka aplikasi dan ketikkan kode   |  | | --- | | // Original source code: https://wiki.keyestudio.com/KS0429\_keyestudio\_TDS\_Meter\_V1.0#Test\_Code  // Project details: https://RandomNerdTutorials.com/arduino-tds-water-quality-sensor/  #define TdsSensorPin A0  #define VREF 5.0 // analog reference voltage(Volt) of the ADC  #define SCOUNT 30 // sum of sample point  int analogBuffer[SCOUNT]; // store the analog value in the array, read from ADC  int analogBufferTemp[SCOUNT];  int analogBufferIndex = 0;  int copyIndex = 0;  float averageVoltage = 0;  float tdsValue = 0;  float temperature = 16; // current temperature for compensation  // median filtering algorithm  int getMedianNum(int bArray[], int iFilterLen){  int bTab[iFilterLen];  for (byte i = 0; i<iFilterLen; i++)  bTab[i] = bArray[i];  int i, j, bTemp;  for (j = 0; j < iFilterLen - 1; j++) {  for (i = 0; i < iFilterLen - j - 1; i++) {  if (bTab[i] > bTab[i + 1]) {  bTemp = bTab[i];  bTab[i] = bTab[i + 1];  bTab[i + 1] = bTemp;  }  }  }  if ((iFilterLen & 1) > 0){  bTemp = bTab[(iFilterLen - 1) / 2];  }  else {  bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;  }  return bTemp;  }  void setup(){  Serial.begin(115200);  pinMode(TdsSensorPin,INPUT);  }  void loop(){  static unsigned long analogSampleTimepoint = millis();  if(millis()-analogSampleTimepoint > 40U){ //every 40 milliseconds,read the analog value from the ADC  analogSampleTimepoint = millis();  analogBuffer[analogBufferIndex] = analogRead(TdsSensorPin); //read the analog value and store into the buffer  analogBufferIndex++;  if(analogBufferIndex == SCOUNT){  analogBufferIndex = 0;  }  }    static unsigned long printTimepoint = millis();  if(millis()-printTimepoint > 800U){  printTimepoint = millis();  for(copyIndex=0; copyIndex<SCOUNT; copyIndex++){  analogBufferTemp[copyIndex] = analogBuffer[copyIndex];    // read the analog value more stable by the median filtering algorithm, and convert to voltage value  averageVoltage = getMedianNum(analogBufferTemp,SCOUNT) \* (float)VREF / 1024.0;    //temperature compensation formula: fFinalResult(25^C) = fFinalResult(current)/(1.0+0.02\*(fTP-25.0));  float compensationCoefficient = 1.0+0.02\*(temperature-25.0);  //temperature compensation  float compensationVoltage=averageVoltage/compensationCoefficient;    //convert voltage value to tds value  tdsValue=(133.42\*compensationVoltage\*compensationVoltage\*compensationVoltage - 255.86\*compensationVoltage\*compensationVoltage + 857.39\*compensationVoltage)\*0.5;    //Serial.print("voltage:");  //Serial.print(averageVoltage,2);  //Serial.print("V ");  //Serial.print("TDS Value:");  Serial.println(tdsValue,0);  //Serial.println("ppm");  }  }  } | |
|  | Hubungkan arduino dengan laptop menggunakan kabel USB. Buka arduino ide kemudian upload kode yang telah dibuat pada board arduino. Lihat hasilnya pada serial monitor di pojok kanan atas arduino ide. Set baudrate menjadi 115200. Jika nilai tds sudah muncul maka proses uploa telah berhasil. |
|  | Lepas kabel USB pada laptop dan sambungkan arduino pada raspberry menggunakan kabel USB. |
|  | Masuk pada thonny di rasppberry kemudian ketikkan kode berikut untuk menangkap data pada arduino   |  | | --- | | import serial  ser = serial.Serial('/dev/ttyACM0', 115200, timeout=1)  ser.flush()  try:  while True:  if ser.in\_waiting > 0:  nutrisi = ser.readline().decode('utf-8').rstrip()  nutrisi3 = int(float(nutrisi))  print(nutrisi3)  except:  nutrisi = 0  print(nutrisi) |   Jalankan program tersebut untuk melihat apakah raspberry sudah menangkap data dari arduino |
| **Tahap Pengerjaan Aplikasi** | |
|  | Buat folder utama aplikasi, dalam folder tersebut, buat file templates dan static. |
|  | Download bootstrap pada link berikut <https://github.com/ColorlibHQ/AdminLTE/archive/refs/tags/v3.1.0.zip> |
|  | Ekstrak file zip, kemudian copy folder docs, build, dist, dan plugins ke dalam folder static pada folder utama aplikasi. |
|  | Buat file base.html dan simpan pada folder templates.   |  | | --- | | <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <meta http-equiv="refresh" content="10">  <title>Smart Hydroponic</title>  <!-- Google Font: Source Sans Pro -->  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">  <!-- Font Awesome -->  <link rel="stylesheet" href="../static/plugins/fontawesome-free/css/all.min.css">  <!-- Ionicons -->  <link rel="stylesheet" href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">  <!-- Tempusdominus Bootstrap 4 -->  <link rel="stylesheet" href="../static/plugins/tempusdominus-bootstrap-4/css/tempusdominus-bootstrap-4.min.css">  <!-- iCheck -->  <link rel="stylesheet" href="../static/plugins/icheck-bootstrap/icheck-bootstrap.min.css">  <!-- JQVMap -->  <link rel="stylesheet" href="../static/plugins/jqvmap/jqvmap.min.css">  <!-- Theme style -->  <link rel="stylesheet" href="../static/dist/css/adminlte.min.css">  <!-- overlayScrollbars -->  <link rel="stylesheet" href="../static/plugins/overlayScrollbars/css/OverlayScrollbars.min.css">  <!-- Daterange picker -->  <link rel="stylesheet" href="../static/plugins/daterangepicker/daterangepicker.css">  <!-- summernote -->  <link rel="stylesheet" href="../static/plugins/summernote/summernote-bs4.min.css">  <link rel="stylesheet" href="../static/tombol.css">  <link rel="shortcut icon" href="../static/dist/img/favicon.png"/>    <link href="../static/css/bootstrap4-toggle.css" rel="stylesheet">  <link href="../static/css/bootstrap4-toggle.min.css" rel="stylesheet">  <script src="../static/js/bootstrap4-toggle.js"></script>  <script src="../static/js/bootstrap4-toggle.min.js"></script>  <script src="../static/js/bootstrap4-toggle.min.js.map"></script>  <script src='https://cdn.jsdelivr.net/npm/chart.js@3.7.0/dist/chart.min.js'></script>  </head>  <body class="hold-transition sidebar-mini layout-fixed">  <div class="wrapper">  <!-- Navbar -->  <nav class="main-header navbar navbar-expand navbar-white navbar-light">  <!-- Left navbar links -->  <ul class="navbar-nav">  <li class="nav-item">  <a class="nav-link" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars"></i></a>  </li>  <li class="nav-item d-none d-sm-inline-block">  <a href="{{ url\_for('index') }}" class="nav-link active">Dashboard</a>  </li>  </ul>  <!-- Right navbar links -->  <ul class="navbar-nav ml-auto">  <!-- Navbar Search -->  <li class="nav-item">  <a class="nav-link" data-widget="navbar-search" href="#" role="button">  <i class="fas fa-search"></i>  </a>  <div class="navbar-search-block">  <form class="form-inline">  <div class="input-group input-group-sm">  <input class="form-control form-control-navbar" type="search" placeholder="Search"  aria-label="Search">  <div class="input-group-append">  <button class="btn btn-navbar" type="submit">  <i class="fas fa-search"></i>  </button>  <button class="btn btn-navbar" type="button" data-widget="navbar-search">  <i class="fas fa-times"></i>  </button>  </div>  </div>  </form>  </div>  </li>  <!-- Notifications Dropdown Menu -->  <li class="nav-item">  <a class="nav-link" data-widget="fullscreen" href="#" role="button">  <i class="fas fa-expand-arrows-alt"></i>  </a>  </li>  <li class="nav-item dropdown">  <a class="nav-link" data-toggle="dropdown" href="#">  <i class="far fa-user"></i>  </a>  <div class="dropdown-menu dropdown-menu-lg dropdown-menu-right">  <span class="dropdown-item dropdown-header">{{ session.username }}</span>  <div class="dropdown-divider"></div>  <a href="/logout" class="dropdown-item">  <i class="far fa-user"></i> Logout from {{ session.username }}? </a>  </div>  </li>  </ul>  </nav>  <!-- /.navbar -->  <!-- Main Sidebar Container -->  <aside class="main-sidebar sidebar-dark-primary elevation-4">  <!-- Brand Logo -->  <a href="#" class="brand-link">  <img src="../static/dist/img/HydroponicLogo.png"  alt="Hydroponic Logo" class="brand-image img-circle elevation-3" style="opacity: .8">  <span class="brand-text font-weight-light">Smart Hydroponic</span>  </a>  <!-- Sidebar -->  <div class="sidebar">  <!-- Sidebar user panel (optional) -->  <div class="user-panel mt-3 pb-3 mb-3 d-flex">  <div class="image">  <img src="../static/dist/img/VEDCLogo.png" class="img-circle elevation-2" alt="User Image">  </div>  <div class="info">  <a href="#" class="d-block">{{ session.username }}</a>  </div>  </div>  <!-- SidebarSearch Form -->  <div class="form-inline">  <div class="input-group" data-widget="sidebar-search">  <input class="form-control form-control-sidebar" type="search" placeholder="Search"  aria-label="Search">  <div class="input-group-append">  <button class="btn btn-sidebar">  <i class="fas fa-search fa-fw"></i>  </button>  </div>  </div>  </div>  <!-- Sidebar Menu -->  <nav class="mt-2">  <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu"  data-accordion="false">  <!-- Add icons to the links using the .nav-icon class  with font-awesome or any other icon font library -->  <li class="nav-item">  <a href="#" class="nav-link active">  <i class="nav-icon fas fa-tachometer-alt"></i>  <p>  Dashboard  </p>  </a>  </li>  </ul>  </nav>  <!-- /.sidebar-menu -->  </div>  <!-- /.sidebar -->  </aside>  <!-- Content Wrapper. Contains page content -->  <div class="content-wrapper">  <!-- Content Header (Page header) -->  <section class="content-header">  <div class="container-fluid">  <div class="row mb-2">  <div class="col-sm-6">  <h1>Dashboard IOT Hidroponik</h1>  </div>  <div class="col-sm-6">  <ol class="brand float-sm-right">  <img src="../static/dist/img/VEDCLogo.png" alt="Hydroponic Logo" width="150" height="100">  </ol>  </div>  </div>  </div><!-- /.container-fluid -->  </section>  <!-- Main content -->  <section class="content">  <div class="container-fluid">  <!-- Small boxes (Stat box) -->  <div class="row">  {% block content %} {% endblock %}  </div>  <!-- /.row (main row) -->  </div><!-- /.container-fluid -->  </section>  <!-- /.content -->  </div>  <!-- /.content-wrapper -->  <footer class="main-footer">  <strong>AdminLTE </strong>    <div class="float-right d-none d-sm-inline-block">  <b>July-August, 2022</b>  </div>  </footer>  <!-- Control Sidebar -->  <aside class="control-sidebar control-sidebar-dark">  <!-- Control sidebar content goes here -->  </aside>  <!-- /.control-sidebar -->  </div>  <!-- ./wrapper -->  <!-- jQuery -->  <script src="../static/plugins/jquery/jquery.min.js"></script>  <!-- jQuery UI 1.11.4 -->  <script src="../static/plugins/jquery-ui/jquery-ui.min.js"></script>  <!-- Resolve conflict in jQuery UI tooltip with Bootstrap tooltip -->  <script>  $.widget.bridge('uibutton', $.ui.button)  </script>  <!-- Bootstrap 4 -->  <script src="../static/plugins/bootstrap/js/bootstrap.bundle.min.js"></script>  <!-- ChartJS -->  <script src="../static/plugins/chart.js/Chart.min.js"></script>  <!-- Sparkline -->  <script src="../static/plugins/sparklines/sparkline.js"></script>  <!-- JQVMap -->  <script src="../static/plugins/jqvmap/jquery.vmap.min.js"></script>  <script src="../static/plugins/jqvmap/maps/jquery.vmap.usa.js"></script>  <!-- jQuery Knob Chart -->  <script src="../static/plugins/jquery-knob/jquery.knob.min.js"></script>  <!-- daterangepicker -->  <script src="../static/plugins/moment/moment.min.js"></script>  <script src="../static/plugins/daterangepicker/daterangepicker.js"></script>  <!-- Tempusdominus Bootstrap 4 -->  <script src="../static/plugins/tempusdominus-bootstrap-4/js/tempusdominus-bootstrap-4.min.js"></script>  <!-- Summernote -->  <script src="../static/plugins/summernote/summernote-bs4.min.js"></script>  <!-- overlayScrollbars -->  <script src="../static/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js"></script>  <!-- AdminLTE App -->  <script src="../static/dist/js/adminlte.js"></script>  <script src="../static/dist/js/pages/dashboard.js"></script>  <script src="../static/dist/js/chart-area-demo.js"></script>  <script src="../static/dist/js/chart-pie-demo.js"></script>  <!-- JustGage -->  <script type="text/javascript"  src="https://cdnjs.cloudflare.com/ajax/libs/raphael/2.1.4/raphael-min.js"></script>  <script type="text/javascript"  src="https://cdnjs.cloudflare.com/ajax/libs/justgage/1.2.9/justgage.min.js"></script>  <script type="text/javascript"  src="https://cdn.jsdelivr.net/npm/chart.js@3.7.0/dist/chart.min.js"></script>  </body>  </html>  <script>  var g1 = new JustGage({  id: "g1",  value: {{ temp }},  min: 0,  max: 80,  symbol: '°C',  levelColors: ["#00fff6", "#ff00fc", "#1200ff"],  pointer: true,  pointerOptions: {  toplength: -15,  bottomlength: 10,  bottomwidth: 12,  color: '#8e8e93',  stroke: '#ffffff',  stroke\_width: 3,  stroke\_linecap: 'round'  },  gaugeWidthScale: 0.6,  counter: true,  relativeGaugeSize: true,  donut: true  });  var g2 = new JustGage({  id: "g2",  value: {{ temp\_c }},  min: 0,  max: 80,  symbol: '°C',  pointer: true,  pointerOptions: {  toplength: -15,  bottomlength: 10,  bottomwidth: 12,  color: '#8e8e93',  stroke: '#ffffff',  stroke\_width: 3,  stroke\_linecap: 'round'  },  gaugeWidthScale: 0.6,  counter: true,  relativeGaugeSize: true,  donut: true  });  var g3 = new JustGage({  id: "g3",  value: {{ humi }},  min: 0,  max: 120,  symbol: '%',  pointer: true,  pointerOptions: {  toplength: -15,  bottomlength: 10,  bottomwidth: 12,  color: '#8e8e93',  stroke: '#ffffff',  stroke\_width: 3,  stroke\_linecap: 'round'  },  gaugeWidthScale: 0.6,  counter: true,  relativeGaugeSize: true,  donut: true  });  var g4 = new JustGage({  id: "g4",  value: {{ nutrisi }},  min: 0,  max: 3000,  symbol: '',  pointer: true,  pointerOptions: {  toplength: -15,  bottomlength: 10,  bottomwidth: 12,  color: '#8e8e93',  stroke: '#ffffff',  stroke\_width: 3,  stroke\_linecap: 'round'  },  gaugeWidthScale: 0.6,  counter: true,  relativeGaugeSize: true,  donut: true  });  </script> | |
|  | Membuat file sensor.html sebagai tampilan dashboard aplikasi. File sensor.html akan mengekstends file base.html.   |  | | --- | | {% extends 'base2.html' %}  {% block title %} Dashboard Hidroponik {% endblock %}  {% block content %}  <div class="container-fluid">  <div class="row">  <div class="col-md-4">  <!-- AREA CHART -->  <div class="card card-primary">  <div class="card-header">  <h3 class="card-title">Suhu Air</h3>  <div class="card-tools">  <button type="button" class="btn btn-tool" data-card-widget="collapse">  <i class="fas fa-minus"></i>  </button>  <button type="button" class="btn btn-tool" data-card-widget="remove">  <i class="fas fa-times"></i>  </button>  </div>  </div>  <div class="card-body">  <div class="chart">  <div id="g1"  style="min-height: 250px; height: 250px; max-height: 250px; max-width: 100%;"></div>  </div>  </div>  <!-- /.card-body -->  </div>  <!-- /.card -->  </div>  <!-- /.col (LEFT) -->  <div class="col-md-4">  <!-- BAR CHART -->  <div class="card card-success">  <div class="card-header">  <h3 class="card-title">Suhu Ruangan</h3>  <div class="card-tools">  <button type="button" class="btn btn-tool" data-card-widget="collapse">  <i class="fas fa-minus"></i>  </button>  <button type="button" class="btn btn-tool" data-card-widget="remove">  <i class="fas fa-times"></i>  </button>  </div>  </div>  <div class="card-body">  <div class="chart">  <div id="g2"  style="min-height: 250px; height: 250px; max-height: 250px; max-width: 100%;"></div>  </div>  </div>  <!-- /.card-body -->  </div>  <!-- /.card -->  </div>  <div class="col-md-4">  <!-- DONUT CHART -->  <div class="card card-secondary">  <div class="card-header">  <h3 class="card-title">Kelembapan</h3>  <div class="card-tools">  <button type="button" class="btn btn-tool" data-card-widget="collapse">  <i class="fas fa-minus"></i>  </button>  <button type="button" class="btn btn-tool" data-card-widget="remove">  <i class="fas fa-times"></i>  </button>  </div>  </div>  <div class="card-body">  <div id="g3"  style="min-height: 250px; height: 250px; max-height: 250px; max-width: 100%;"></div>  </div>  <!-- /.card-body -->  </div>  <!-- /.card -->  </div>  <!-- /.col (RIGHT) -->  </div>  <!-- /.row -->  <div class="row">  <div class="col-md-4">  <!-- DONUT CHART -->  <div class="card card-info">  <div class="card-header">  <h3 class="card-title">Kadar Nutrisi</h3>  <div class="card-tools">  <button type="button" class="btn btn-tool" data-card-widget="collapse">  <i class="fas fa-minus"></i>  </button>  <button type="button" class="btn btn-tool" data-card-widget="remove">  <i class="fas fa-times"></i>  </button>  </div>  </div>  <div class="card-body">  <div id="g4"  style="min-height: 250px; height: 250px; max-height: 250px; max-width: 100%;"></div>  </div>  <!-- /.card-body -->  </div>  <!-- /.card -->  </div>  <div class="col">  <!-- DONUT CHART -->  <div class="card card-secondary">  <div class="card-header">  <h3 class="card-title">Pencahayaan</h3>  <div class="card-tools">  <button type="button" class="btn btn-tool" data-card-widget="collapse">  <i class="fas fa-minus"></i>  </button>  <button type="button" class="btn btn-tool" data-card-widget="remove">  <i class="fas fa-times"></i>  </button>  </div>  </div>  <div class="card-body">  <div class="row no-gutters align-items-center">  <div class="col">  <div class="h2 mb-0 font-weight-bold text-gray-800">Pencahayaan<br>Ruangan </div>  {% if cahaya == 1 %}  <h5>Terang</h5>  {% else %}  <h5>Gelap</h5>  {% endif %}  <div class="h2 mb-0 font-weight-bold text-gray-800">Status Lampu </div>  {% if cahaya == 1 %}  <h5>Redup</h5>  {% else %}  <h5>Menyala</h5>  {% endif %}  <div class="h2 mb-0 font-weight-bold text-gray-800">Sensor Cahaya</div>  <h5>{{ value }}</h5>  </div>  <div class="col">  <div class="row">  <div class="col"></div>  <div class="col">  {% if cahaya == 1 %}  <img src="../static/dist/img/Sun.png" alt="status" width="160" height="160">  {% else %}  <img src="../static/dist/img/Moon.png" alt="status" width="160" height="160">  {% endif %}  </div>  <div class="col"></div>  </div>  </div>  </div>  </div>  <!-- /.card-body -->  </div>  <!-- /.card -->  </div>  </div>  </div><!-- /.container-fluid -->  <div class="col-12">  {% if status == 1 %}  <div class="card card-danger">  {% else %}  <div class="card card-success">  {% endif %}  <div class="card-header">  <div class="card-title">  <h3>Grafik Penggunaan Motor</h3>  </div>  <div class="card-tools">    {% if status == 1 %}  <a href="/motoron" class="btn btn-danger">ON <i class="fas fa-power-off"></i></a>  {% else %}  <a href="/motoron" class="btn btn-success">ON <i class="fas fa-power-off"></i></a>  {% endif %}    {% if status == 1 %}  <a href="/motoroff" class="btn btn-danger">OFF <i class="fas fa-times"></i></a>  {% else %}  <a href="/motoroff" class="btn btn-success">OFF <i class="fas fa-times"></i></a>  {% endif %}  </div>  </div>  <div class="card-body">  <div class="chart">  <canvas id="cobabangzz" style="min-height: 250px; height: 250px; max-height: 250px; max-width: 100%;"></canvas>  <script>  var ctx = document.getElementById("cobabangzz").getContext("2d");  var lineChart = new Chart(ctx, {  type: "line",  data: {  labels: {{ later | safe }},  datasets: [  {  label: "Daya (watt)",  data: {{ daya }},  fill: false,  borderColor: "rgb(211, 211, 211)",  lineTension: 0.1  },  {  label: "Waktu penggunaan (jam)",  data: {{ total }},  fill: false,  borderColor: "rgb(0, 96, 255)",  backgroundColor:'rgb(0, 96, 255)',  lineTension: 0.1  }  ]  },  options: {  responsive: true  }  });    </script>  </div>  </div>  <!-- /.card-body -->  </div>    {% block chart %} {% endblock %}  {% endblock %} |   File coba.html akan mengekstends file base.html. Simpan file sensor.html pada folder templates. |
|  | Buat file login.html, register.html, dan edit.html dan simpan pada folder templates   |  | | --- | | Login.html  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>Smart HIDROPONIK | Log in</title>  <!-- Google Font: Source Sans Pro -->  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">  <!-- Font Awesome -->  <link rel="stylesheet" href="../static/plugins/fontawesome-free/css/all.min.css">  <!-- icheck bootstrap -->  <link rel="stylesheet" href="../static/plugins/icheck-bootstrap/icheck-bootstrap.min.css">  <!-- Theme style -->  <link rel="stylesheet" href="../static/dist/css/adminlte.min.css">  </head>  <body class="hold-transition login-page">  <div class="login-box">  <div class="login-logo">  <a href="/"><b>Smart</b>HIDROPONIK</a>  </div>  <!-- /.login-logo -->  <div class="card">  <div class="card-body login-card-body">  <p class="login-box-msg">Sign in to start your session</p>  {% with messages = get\_flashed\_messages(with\_categories=true) %}  {% if messages %}  {% for category, message in messages %}  <div class="alert alert-{{ category }}" role="alert"> {{ message }}  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>  {% endfor %}  {% endif %}  {% endwith %}  <form action="/loginProses" method="post">  <div class="input-group mb-3">  <input type="email" class="form-control" placeholder="Email" name="email">  <div class="input-group-append">  <div class="input-group-text">  <span class="fas fa-envelope"></span>  </div>  </div>  </div>  <div class="input-group mb-3">  <input type="password" class="form-control" placeholder="Password" name="password">  <div class="input-group-append">  <div class="input-group-text">  <span class="fas fa-lock"></span>  </div>  </div>  </div><br>  <div class="row">  <div class="col-8">  <div class="icheck-primary">  <input type="checkbox" id="remember" name="remember" value="remember">  <label for="remember">  Remember me  </label>  </div>  </div>  <!-- /.col -->  <div class="col-4">  <button type="submit" class="btn btn-primary btn-block">Sign In</button>  </div>  <!-- /.col -->  </div>  </form><br>  <p class="mb-1">  <a href="/edit">I forgot my password</a>  </p>  <p class="mb-0">  <a href="/register" class="text-center">Register a new membership</a>  </p>  </div>    <!-- /.login-card-body -->  </div>  </div>  <!-- /.login-box -->  <!-- jQuery -->  <script src="../static/plugins/jquery/jquery.min.js"></script>  <!-- Bootstrap 4 -->  <script src="../static/plugins/bootstrap/js/bootstrap.bundle.min.js"></script>  <!-- AdminLTE App -->  <script src="../static/dist/js/adminlte.min.js"></script>  </body>  </html> |  |  | | --- | | Register.html  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>Smart HIDROPONIK | Registration Page</title>  <!-- Google Font: Source Sans Pro -->  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">  <!-- Font Awesome -->  <link rel="stylesheet" href="../static/plugins/fontawesome-free/css/all.min.css">  <!-- icheck bootstrap -->  <link rel="stylesheet" href="../static/plugins/icheck-bootstrap/icheck-bootstrap.min.css">  <!-- Theme style -->  <link rel="stylesheet" href="../static/dist/css/adminlte.min.css">  </head>  <body class="hold-transition register-page">  <div class="register-box">  <div class="register-logo">  <a href="/""><b>Smart</b>HIDROPONIK</a>  </div>  <div class="card">  <div class="card-body register-card-body">  <p class="login-box-msg">Register a new membership</p>  {% with messages = get\_flashed\_messages(with\_categories=true) %}  {% if messages %}  {% for category, message in messages %}  <div class="alert alert-{{ category }}" role="alert"> {{ message }}  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>  {% endfor %}  {% endif %}  {% endwith %}  <form action="/registerProses" method="post">  <div class="input-group mb-3">  <input type="text" class="form-control" placeholder="Full name" name="name">  <div class="input-group-append">  <div class="input-group-text">  <span class="fas fa-user"></span>  </div>  </div>  </div>  <div class="input-group mb-3">  <input type="email" class="form-control" placeholder="Email" name="email">  <div class="input-group-append">  <div class="input-group-text">  <span class="fas fa-envelope"></span>  </div>  </div>  </div>  <div class="input-group mb-3">  <input type="password" class="form-control" placeholder="Password" name="password">  <div class="input-group-append">  <div class="input-group-text">  <span class="fas fa-lock"></span>  </div>  </div>  </div>  <div class="row">  <div class="col-sm">  </div><br>  <!-- /.col -->  <div class="col-sm">  <button type="submit" class="btn btn-primary btn-block">Register</button>  </div>  <!-- /.col -->  <div class="col-sm">  </div><br>  </div>  </form><br>    <a href="/" class="text-center">I already have a membership</a>  </div>  <!-- /.form-box -->  </div><!-- /.card -->  </div>  <!-- /.register-box -->  <!-- jQuery -->  <script src="../static/plugins/jquery/jquery.min.js"></script>  <!-- Bootstrap 4 -->  <script src="../static/plugins/bootstrap/js/bootstrap.bundle.min.js"></script>  <!-- AdminLTE App -->  <script src="../static/dist/js/adminlte.min.js"></script>  </body>  </html> |  |  | | --- | | Edit.html  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>Smart HIDROPONIK | Change password</title>  <!-- Google Font: Source Sans Pro -->  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">  <!-- Font Awesome -->  <link rel="stylesheet" href="../static/plugins/fontawesome-free/css/all.min.css">  <!-- icheck bootstrap -->  <link rel="stylesheet" href="../static/plugins/icheck-bootstrap/icheck-bootstrap.min.css">  <!-- Theme style -->  <link rel="stylesheet" href="../static/dist/css/adminlte.min.css">  </head>  <body class="hold-transition login-page">  <div class="login-box">  <div class="login-logo">  <a href="/"><b>Smart</b>HIDROPONIK</a>  </div>  <!-- /.login-logo -->  <div class="card">  <div class="card-body login-card-body">  <p class="login-box-msg">Forgot my password</p>  {% with messages = get\_flashed\_messages(with\_categories=true) %}  {% if messages %}  {% for category, message in messages %}  <div class="alert alert-{{ category }}" role="alert"> {{ message }}  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  <span aria-hidden="true">&times;</span>  </button>  </div>  {% endfor %}  {% endif %}  {% endwith %}  <form action="/editProses" method="post">  <div class="input-group mb-3">  <input type="email" class="form-control" placeholder="Email" name="email">  <div class="input-group-append">  <div class="input-group-text">  <span class="fas fa-envelope"></span>  </div>  </div>  </div>  <div class="input-group mb-3">  <input type="password" class="form-control" placeholder="Password" name="password">  <div class="input-group-append">  <div class="input-group-text">  <span class="fas fa-lock"></span>  </div>  </div>  </div><br>  <div class="row">    <!-- /.col -->  <div class="col">  <button type="submit" class="btn btn-primary btn-block">Save changes</button>  </div>  <!-- /.col -->  </div>  </form><br>    </div>  <!-- /.login-card-body -->  </div>  </div>  <!-- /.login-box -->  <!-- jQuery -->  <script src="../static/plugins/jquery/jquery.min.js"></script>  <!-- Bootstrap 4 -->  <script src="../static/plugins/bootstrap/js/bootstrap.bundle.min.js"></script>  <!-- AdminLTE App -->  <script src="../static/dist/js/adminlte.min.js"></script>  </body>  </html> | |
|  | Install flask sqlalchemy melalui terminal dengan mengetikkan  pip3 install flask-sqlalchemy |
|  | Buat file main.py dan simpan pada folder utama aplikasi. File ini akan berfungsi sebagai file utama yang akan dijalankan oleh sistem. |
|  | Import beberapa library yang dibutuhkan oleh aplikasi. |
|  | Setting output pada raspberry. |
|  | Buat database untuk menyimpan data berupa waktu dan daya penggunaan motor, data user, dan data sensor.    Database akan tersimpan secara otomatis pada folder utama dengan nama “hidro.sqlite3”. Pada database tersebut terdapat tabel bernama “waktu”, “user”, dan “suhu” sebagai hasil output dari class waktu pada file main.py |
|  | Buat fungsi cahaya untuk menangkap output sensor ldr.    Fungsi ini akan membaca sensor ldr dan mengembalikan value atau nilai sensor yang akan ditampilkan pada halaman web. |
|  | Buat file nutrisi.py yang berfungsi untuk membaca sensor tds pada arduino dan mengirimkan datanya pada file main.py yang kemudian akan ditmpilkan pada web |
|  | Membuat route utama (“/index”) sebagai menu utama aplikasi. Pada route ini terdapat fungsi untuk membaca data waktu dan daya yang tersimpan pada database serta membaca data dari sensor DHT22 dan sensor DS18B20.      Pada route ini juga terdapat sintaks untuk mengambil data penggunaan motor dan sensor yang telah disimpan pada database untuk kemudian ditampilkan dalam bentuk grafik.    Pada perulangan while True untuk membaca data dari sensor dht dan dsb, tambahkan sintaks untuk membaca sensor ldr dan menyalakan lampu secara otomatis.    Tambahkan juga fungsi untuk mengambil data nutrisi dari file nutrisi.py di dalam perulangan while True    Kemudian tambahkan sintaks untuk memasukkan data sensor ke dalam database, sintaks ini juga berada di dalam perulangan while True.    Fungsi index akan mengembalikan file sensor3.html yang tersimpan pada folder templates serta data yang telah didefinisikan dan menampilkannya pada halaman web. |
|  | Buat file sensor3.html yang berfungsi untuk menampilkan data sensor dalam bentuk grafik. File ini akan mengekstends file sensor.html   |  | | --- | | {% extends 'sensor2.html' %}  {% block chart %}  <div class="col-12">  {% if status == 1 %}  <div class="card card-danger">  {% else %}  <div class="card card-success">  {% endif %}  <div class="card-header">  <div class="card-title">  <h3>Grafik Sensor</h3>  </div>  <div class="card-tools">    {% if status == 1 %}  <a href="/index" class="btn btn-danger">Suhu <i class=" "></i></a>  {% else %}  <a href="/index" class="btn btn-success">Suhu <i class=" "></i></a>  {% endif %}    {% if status == 1 %}  <a href="/convert" class="btn btn-danger">Prediksi <i class=" "></i></a>  {% else %}  <a href="/convert" class="btn btn-success">Prediksi <i class=" "></i></a>  {% endif %}  </div>  </div>  <div class="card-body">  <div class="chart">  <canvas id="coba" style="min-height: 250px; height: 250px; max-height: 250px; max-width: 100%;"></canvas>  <script>  var ctp = document.getElementById("coba").getContext("2d");  var lineChart = new Chart(ctp, {  type: "line",  data: {  labels: {{ suhuww }},  datasets: [  {  label: "Suhu",  data: {{ suhuww }},  fill: false,  borderColor: "rgb(0, 96, 255)",  backgroundColor:'rgb(0, 96, 255)',  lineTension: 0.1  },  {  label: "Kelembaban",  data: {{ humiww }},  fill: false,  borderColor: "rgb(153, 51, 255)",  backgroundColor:'rgb(153, 51, 255)',  lineTension: 0.1  },  {  label: "Cahaya",  data: {{ valueww }},  fill: false,  borderColor: "rgb(255, 255, 0)",  backgroundColor:'rgb(255, 255, 0)',  lineTension: 0.1  },  {  label: "Nutrisi",  data: {{ nutrisiww }},  fill: false,  borderColor: "rgb(0, 153, 0)",  backgroundColor:'rgb(0, 153, 0)',  lineTension: 0.1  }  ]  },  options: {  responsive: true  }  });  </script>  </div>  </div>  <!-- /.card-body -->  </div>    {% endblock %} | |
|  | Membuat route (“/motoron”) sebagai action dari button yang berfungsi untuk menyalakan motor dan menyimpan data waktu diawal motor digunakan. |
|  | Membuat route (“/motoroff”) sebagai action dari button yang berfungsi untuk mematikan motor dan menyimpan data diakhir motor digunakan, serta menghitung total waktu dan daya penggunaan motor yang akan disimpan pada database. |
|  | Buat route dengan nama (“/convert”). Route ini akan mengubah file data dari file sql menjadi file csv yang akan berfungsi untuk penetapan prediksi suhu yang akan mendatang. |
|  | Buat route(“/prediksi”) untuk menyatakan prediksi yang memanfaatkan data science dalam penerapannya. Para route ini, buat fungsi yang sama persis seperti pada fungsi index, kemudian hapus bagian mengambil data dari tabel suhu, dan tambah fungsi untuk menentukanprediksi suhu.      Fungsi ini akan mebuat grafik dan menampilkannya pada halaman prediksi.html |
|  | Aplikasi akan berjalan di alamat web yang sesuai dengan IP address raspberry. |
|  | Secara keseluruhan, file main.py akan seperti berikut   |  | | --- | | from flask import Flask, request, flash, session, url\_for, redirect, render\_template  import time  #from w1thermsensor import W1ThermSensor  import time  import board  import adafruit\_dht  import serial  import sys  import serial  from time import sleep  from flask\_sqlalchemy import SQLAlchemy  from datetime import datetime  import RPi.GPIO as GPIO  GPIO.setmode(GPIO.BCM)  GPIO.setup(26, GPIO.OUT)  GPIO.setup(17, GPIO.OUT)  GPIO.setup(22, GPIO.OUT)  GPIO.output(26, True)  GPIO.output(17, True)  app = Flask(\_\_name\_\_)  app.config['SECRET\_KEY'] = 'mysecret'  app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///hidro.sqlite3'  app.config['SECRET\_KEY'] = "random string"  db = SQLAlchemy(app)  class User(db.Model):  id = db.Column(db.Integer, primary\_key=True)  email = db.Column(db.String(100), unique=True)  password = db.Column(db.String(100))  name = db.Column(db.String(1000))  class waktu(db.Model):  id = db.Column(db.Integer, primary\_key=True)  first = db.Column(db.String)  later = db.Column(db.String)  total = db.Column(db.Float)  daya = db.Column(db.Float)    class suhu(db.Model):  id = db.Column(db.Integer, primary\_key=True)  humi = db.Column(db.Float)  value = db.Column(db.Float)  temp\_c = db.Column(db.Float)  nutrisi = db.Column(db.Float)    db.create\_all()  @app.route('/')  def login():  return render\_template('login.html')  @app.route('/edit')  def edit():  return render\_template('edit.html')  @app.route('/editProses', methods=['POST'])  def edit\_proses():  email = request.form.get('email')  password = request.form.get('password')  user = User.query.filter\_by(email=email).first()    if not user:  flash('Email belum terdaftar')  return redirect(url\_for('register'))  num\_rows\_updated = User.query.filter\_by(email=email).update(dict(password=password))  db.session.commit()  return redirect(url\_for('login'))    @app.route('/register')  def register():  return render\_template('register.html')  @app.route('/registerProses', methods=['POST'])  def proses\_register():  email = request.form.get('email')  name = request.form.get('name')  password = request.form.get('password')  user = User.query.filter\_by(email=email).first()  if user:  flash('Email Sudah ada')  return redirect(url\_for('register'))  new\_user = User(email=email, name=name, password=password)    db.session.add(new\_user)  db.session.commit()  return redirect(url\_for('login'))  @app.route('/loginProses', methods=['POST'])  def proses\_login():  email = request.form.get('email')  password = request.form.get('password')  user = User.query.filter\_by(email=email).first()    if (user.password != password):  flash('Please check your login details and try again.')  return redirect(url\_for('login'))    session['username'] = user.name  return redirect(url\_for('index'))  @app.route('/logout')  def logout():  session.pop('username', None)  return redirect(url\_for('login'))  @app.route("/cahaya")  def rc\_time():  count=0  GPIO.setup(27, GPIO.OUT)  GPIO.output(27, GPIO.LOW)  time.sleep(1)  GPIO.setup(27, GPIO.IN)  while (GPIO.input(27) == GPIO.LOW):  count += 1  return count  @app.route("/index")  def index():  #sensor = W1ThermSensor()  dhtDevice = adafruit\_dht.DHT22(board.D21, use\_pulseio=False)    global waktu  data = waktu.query.all()  first = []  later = []  total = []  daya = []  for amounts in data:  first.append(amounts.first)  later.append(amounts.later)  total.append(amounts.total)  daya.append(amounts.daya)    global suhu  data1 = suhu.query.all()  suhuww = []  humiww = []  valueww = []  nutrisiww = []    for amounts in data1:  suhuww.append(amounts.temp\_c)  humiww.append(amounts.humi)  valueww.append(amounts.value)  nutrisiww.append(amounts.nutrisi)  while True:  try:  value = rc\_time()  valuep = round((value/10000)\*100, 1)  print(value)  if (value > 10000):  cahaya = 0  GPIO.output(22, False)  if (value < 10000):  cahaya = 1  GPIO.output(22, True)    #temp = sensor.get\_temperature()  temp\_c = dhtDevice.temperature  print(temp\_c)  temp\_f = temp\_c \* (9 / 5) + 32  humi = dhtDevice.humidity    from nutrisi import nutrisi  nutrisi = nutrisi()  print(nutrisi)    new\_suhu = suhu(humi=humi, value=value, temp\_c=temp\_c, nutrisi=nutrisi)  db.session.add(new\_suhu)  db.session.commit()    except RuntimeError as error:  print(error.args[0])  time.sleep(5)  continue    return render\_template('sensor3.html', suhuww=suhuww, humiww=humiww, valueww=valueww, nutrisiww=nutrisiww, temp=27,temp\_c=temp\_c,humi=humi, nutrisi=nutrisi, data=data, first=first, later=later, total=total, daya=daya, cahaya=cahaya, value=value, status=GPIO.input(17))    @app.route("/motoron")  def motoron():  GPIO.output(17, False)  first\_time = datetime.now()  session['first'] = first\_time  return redirect(url\_for('index'))  @app.route("/motoroff")  def motoroff():  GPIO.output(17, True)    first\_time = session['first']  later\_time = datetime.now()  later\_time = later\_time.replace(microsecond=0)    first = str(first\_time)  later = str(later\_time)    tawal = (later\_time.hour\*3600)+(later\_time.minute\*60)+(later\_time.second)  takhir = (first\_time.hour\*3600)+(first\_time.minute\*60)+(first\_time.second)  time\_diff = ((tawal - takhir)/3600)  total = round(time\_diff, 6)  daya = round(17\*total, 6)    new\_user = waktu(total=total, first=first, later=later, daya=daya)  db.session.add(new\_user)  db.session.commit()    return redirect(url\_for('index'))  @app.route("/convert")  def convert():  import sqlite3  import csv  con = sqlite3.connect('hidro.sqlite3')  outfile = open('hidroponik.csv', 'w')  outcsv = csv.writer(outfile)  cursor = con.execute('select humi, value, temp\_c from suhu')  # dump column titles (optional)  outcsv.writerow(x[0] for x in cursor.description)  # dump rows  outcsv.writerows(cursor.fetchall())  outfile.close()  return redirect(url\_for('prediksi'))  @app.route("/prediksi")  def prediksi():  #sensor = W1ThermSensor()  dhtDevice = adafruit\_dht.DHT22(board.D21, use\_pulseio=False)  ser = serial.Serial('/dev/ttyACM0', 115200, timeout=1)  ser.flush()    global waktu  data = waktu.query.all()  first = []  later = []  total = []  daya = []  for amounts in data:  first.append(amounts.first)  later.append(amounts.later)  total.append(amounts.total)  daya.append(amounts.daya)    while True:  try:  value = rc\_time()  valuep = round((value/10000)\*100, 1)  print(value)  if (value > 10000):  cahaya = 0  GPIO.output(22, False)  if (value < 10000):  cahaya = 1  GPIO.output(22, True)    #temp = sensor.get\_temperature()  temp\_c = dhtDevice.temperature  print(temp\_c)  temp\_f = temp\_c \* (9 / 5) + 32  humi = dhtDevice.humidity    from nutrisi import nutrisi  nutrisi = nutrisi()  print(nutrisi)    new\_suhu = suhu(humi=humi, value=value, temp\_c=temp\_c, nutrisi=nutrisi)  db.session.add(new\_suhu)  db.session.commit()    except RuntimeError as error:  print(error.args[0])  time.sleep(5)  continue    #import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  import os  import io  import base64  dataset = pd.read\_csv('hidroponik.csv')    x = dataset.iloc[:,[0,1]].values  y = dataset.loc[:, 'temp\_c'].values  y = y.astype('int')    from sklearn.model\_selection import train\_test\_split  x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.25, random\_state=0)    from sklearn.preprocessing import StandardScaler  sc= StandardScaler()  x\_train = sc.fit\_transform(x\_train)  x\_test = sc.fit\_transform(x\_test)    from sklearn.naive\_bayes import GaussianNB  classifier = GaussianNB()  classifier.fit(x\_train, y\_train)    y\_pred = classifier.predict(x\_test)  y = y\_pred[-1]  print(y)  #plt.plot(y\_pred)  from matplotlib.backends.backend\_agg import FigureCanvasAgg as FigureCanvas  from matplotlib.figure import Figure    fig = Figure()  axis = fig.add\_subplot(1, 1, 1)  axis.set\_title("")  axis.set\_xlabel("")  axis.set\_ylabel("")  axis.grid()  axis.plot(y\_pred)  pngImage = io.BytesIO()  FigureCanvas(fig).print\_png(pngImage)    # Encode PNG image to base64 string  pngImageB64String = "data:image/png;base64,"  pngImageB64String += base64.b64encode(pngImage.getvalue()).decode('utf8')  '''i = 'plot1'  plt.savefig('static/dist/{}'.format(i))  plt.close()'''  return render\_template('prediksi.html', image=pngImageB64String, y=y, temp=27,temp\_c=temp\_c,humi=humi, nutrisi=nutrisi, data=data, first=first, later=later, total=total, daya=daya, cahaya=cahaya, value=value, status=GPIO.input(17))  if \_\_name\_\_ == "\_\_main\_\_":  app.run(host="0.0.0.0", port=5000) |   Terdapat folder lengkap pada github <https://github.com/ekoboe/IOT-Hidroponik.git> |
| **Tampilan aplikasi** | |
| Jika telah mengikuti langkah-langkah dengan benar, tampilan aplikasi akan seperti berikut, | |