



BALAI BESAR PENGEMBANGAN  
PENJAMINAN MUTU PENDIDIKAN VOKASI  
BIDANG OTOMOTIF DAN ELEKTRONIKA

**MERDEKA  
BELAJAR**

# PEMROGRAMAN WEB SERVER IOT BERBASIS MICROPYTHON

## #4. Microframework Microdot

Penyusun:

Dr. Eko Subiyantoro, S.Pd, S.ST, MT

Ilham Athaariq Gistanda



Jl. Teluk Mandar, Arjosari Tromol Pos 5 Malang 65126  
Telp. (0341) 491239, 495849 Fax. (0341) 491342  
Instagram : bbppmpvboe.kemdikbud  
Website : bbppmpvboe.kemdikbud.go.id/  
Facebook : Bbppmpvboe.kemdikbud  
Youtube : bbppmpv boe  
Tiktok : bbppmpvboemlg



You can **STOP**  
**CORRUPTION**



## PRAKTIKUM 4

### IOT RASPBERRY PI PICO W DENGAN MICRODOT

#### 4.1. Tujuan Praktikum

Peserta pelatihan mampu:

1. Memahami perangkat raspberry pico pi W
2. Menerapkan microframework microdot
3. Merancang tampilan Web Server IoT
4. Membuat projek IoT.

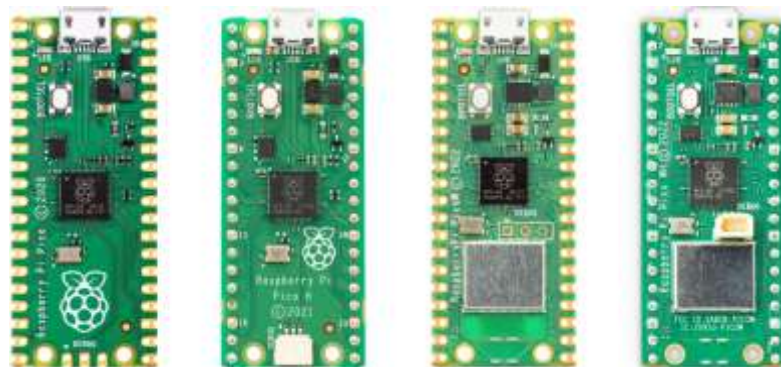
#### 4.2. Pendahuluan

##### 4.2.1 Pico W

Single-Board Computer seperti Raspberry Pi umumnya memiliki sistem operasinya sendiri dan dirancang untuk mengerjakan tugas-tugas komputasi, sedangkan microcontroller adalah yang bertugas mengatasi input analog. Pada tahun 2021, Raspberry Pi rupanya sudah siap memperluas portofolio produknya. Mereka baru saja memperkenalkan Raspberry Pi Pico, microcontroller perdananya yang mengemas chip buatan mereka sendiri. Chip yang dimaksud adalah RP2040, yang dideskripsikan sebagai chip mungil tapi berkinerja tinggi, dengan kapabilitas I/O yang fleksibel.

Bahasa pemrograman yang digunakan pada Pico bahasa pemrograman C, tetapi juga mendukung bahasa lain yang lebih spesifik seperti MicroPython. Total ada 30 pin GPIO pada tubuh Pico – empat di antaranya bisa digunakan sebagai input analog – dan ia turut dibekali port micro-USB yang mendukung mode mass storage.

Varian Raspberry Pi Pico saat ini terdiri dari empat yaitu; Raspberry Pi Pico (paling kiri), Pico H (kiri tengah), Pico W (kanan tengah), dan Pico WH (paling kanan).



Gambar 4.1 Varian Rapsberry Pi Pico



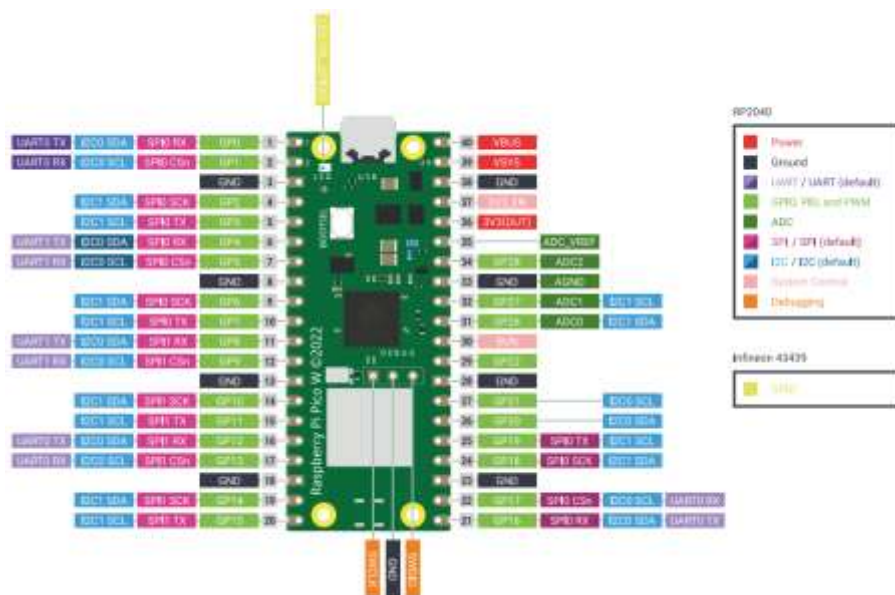
Raspberry Pi Pico W menambahkan antarmuka nirkabel 2.4GHz pita tunggal terpasang (802.11n) menggunakan Infineon CYW43439 dengan tetap mempertahankan faktor bentuk Pico. Antarmuka nirkabel 2.4GHz on-board memiliki beberapa fitur berikut:

- Wireless (802.11n), single-band (2.4 GHz)
- WPA3
- Mampu akses hingga empat client
- Bluetooth 5.2
  - Dukungan untuk Bluetooth LE Central dan Peripheral
  - Dukungan untuk Bluetooth Klasik

Antena adalah antena onboard berlisensi dari ABRACON dan antarmuka nirkabel terhubung melalui SPI ke mikrokontroler RP2040.

Karena keterbatasan pin, beberapa pin antarmuka nirkabel digunakan bersama. CLK dibagikan dengan monitor VSYS, jadi hanya jika tidak ada transaksi SPI yang sedang berlangsung, VSYS dapat dibaca melalui ADC. Infineon CYW43439 DIN/DOUT dan IRQ berbagi satu pin pada RP2040. Ketika transaksi SPI tidak sedang berlangsung saatnya untuk memeriksa IRQ. Antarmuka biasanya berjalan pada 33MHz.

Untuk kinerja nirkabel terbaik, antena harus berada di ruang kosong. Misalnya, meletakkan logam di bawah atau di dekat antena dapat mengurangi kinerjanya baik dalam hal penguatan maupun bandwidth. Menambahkan logam yang diarde ke sisi antena dapat meningkatkan lebar pita antena.



Gambar 4.2 Pinout Pico W

#### 4.2.2 Microdot

Framework Flask digunakan untuk mengembangkan beberapa aplikasi IoT yang dapat berjalan dengan baik di Raspberry Pi Single Board Computer (SBC). Ketika menggunakan MicroPython pada Pico W, framework Flask tidak dapat dijalankan, banyak contoh sayangnya menggunakan soket dalam mengembangkan Server Web mereka sendiri.

Mikro Framework MicroDot yang dikembangkan oleh Miguel Grinberg dapat diterapkan untuk dijalankan di MicroController. MicroDot berjalan pada varian CPython dan MicroPython. tetapi fokus dari posting ini adalah bagaimana menggunakannya di lingkungan MicroPython. Miguel Grinberg membuat quote sebagai berikut,

*"The impossibly small web framework for Python and MicroPython"*

Microdot adalah kerangka web Python minimalis yang terinspirasi oleh Flask, dan dirancang untuk berjalan pada sistem dengan sumber daya terbatas seperti mikrokontroler. Ini berjalan pada Python standar dan pada MicroPython.

Microdot adalah framework aplikasi web yang sangat dapat dikembangkan. Ekstensi yang dijelaskan di bagian ini dikelola sebagai bagian dari proyek Microdot dan dapat diperoleh dari repositori kode sumber yang sama.

#### Instalasi Microdot

Untuk proyek standar Python (CPython), Microdot dan semua ekstensi dapat dipasang dengan pip:

```
pip install microdot
```

Untuk MicroPython, Anda dapat menginstalnya dengan upip jika opsi itu tersedia, tetapi pendekatan yang disarankan adalah meng-copy secara manual microdot.py dan file sumber ekstensi opsional yang diinginkan dari repositori GitHub ke perangkat Pico W.

#### Simple Web Server

Perhatikan contoh kode program berikut ini,

```
1 from microdot import Microdot
2 app = Microdot()
3
4 @app.route('/')
5 def index(request):
6     return 'Hello, world!'
7
8 if __name__ == "__main__":
9     app.run(debug=True)
```

Analisa Program:

Baris ke-1 memanggil module microdot ke dalam program python atau micropython

Baris ke-2 membuat objek Microdot dengan nama app. Maksud dari baris ini membuat sebuah objek sebagai sebuah instance dari kelas Microdot yang diimpor dari package microdot.

Baris ke-3 @app.route akan membuat sebuah hubungan antara URL yang diberikan sebagai argumen dengan fungsi di bawahnya. Artinya setiap kali sebuah web browser mengakses salah satu dari URL ini, Microdot akan memanggil fungsi di bawahnya dan mengirimkan nilai balikan (perintah return) yang sudah dibuat sebagai sebuah response.

Baris ke-5 mendefinisikan fungsi index (request) ketika seseorang mengakses URL root ('/') dari server. Dalam hal ini, hanya kirimkan teksnya "Helo World!" ke browser web klien dengan perintah return.

Baris 8-9 Perintah untuk menjalankan kode program yang di atasnya, server mulai "mendengarkan" pada port default 5000 dan melaporkan kesalahan apa pun.

Output pada browser sebagai berikut,



### Fungsi route()

Dekorator route() digunakan untuk mengaitkan URL aplikasi dengan fungsi yang menanganinya. Satu-satunya yang dibutuhkan argumen ke dekorator adalah bagian jalur dari URL. Perhatikan contoh berikut membuat rute untuk aplikasi root URL,

```
@app.route('/')
def index(request):
    return 'Hello, world!'
```

Ketika klien meminta URL root (misalnya, http://localhost:5000/), Microdot akan memanggil fungsi index(), meneruskannya sebagai objek permintaan. Nilai balik dari fungsi adalah respons yang dikirim ke klien. Di bawah ini adalah contoh lain, ini dengan rute untuk URL dengan dua komponen di jalurnya:

```
1 from microdot import Microdot
2 app = Microdot()
3
4 @app.route('/')
5 def index(request):
6     return 'Hello, world!'
7
8 @app.route('/users/active')
9 def active_users(request):
10    return 'Active users: Susan, Joe, and Bob'
11
12 if __name__ == "__main__":
13     app.run(debug=True)
```

Output pada browser sebagai berikut,



## Struktur Pada Microdot

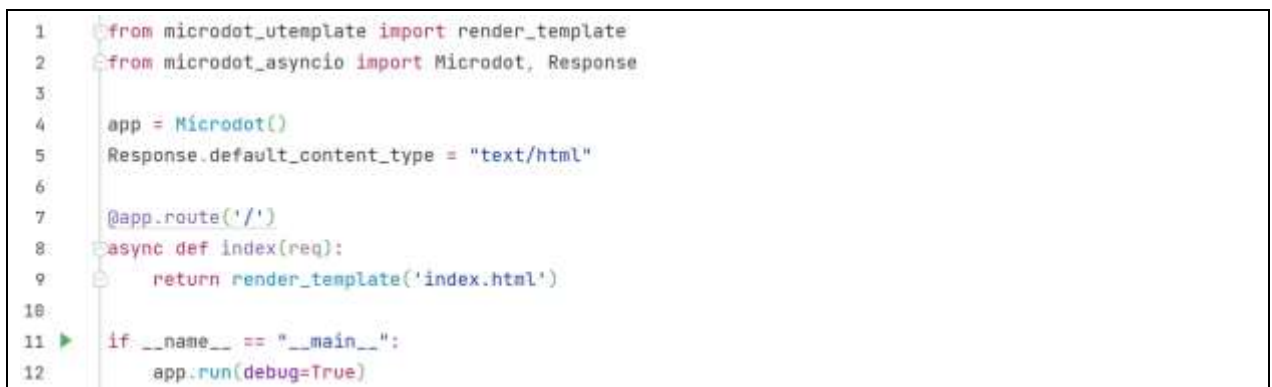
Struktur file dalam sebuah proyek Microdot sebagai berikut,



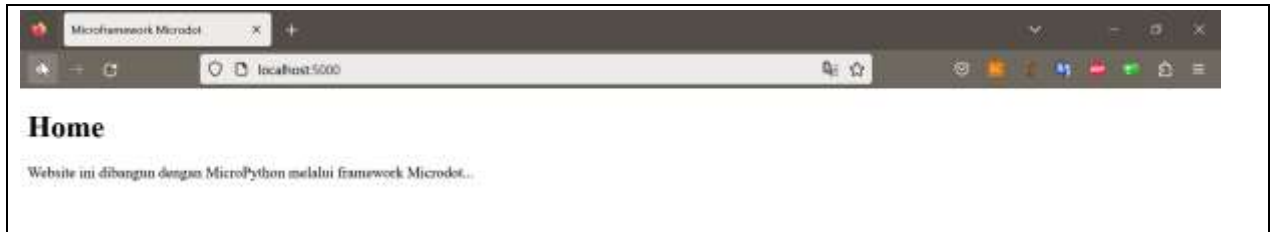
Folder templates untuk ditempatkan file-file html. Perhatikan contoh kode program index.html berikut ini,



Perhatikan contoh kode program app.py berikut ini,



Perhatikan tampilan kode program app.py berikut ini,



## Menambahkan lebih banyak halaman ke situs web

Cara cara menambahkan lebih banyak halaman ke situs web microdot dapat mengikuti langkah-langkah berikut.

- Membuat file about.html selanjutnya simpan di folder templates

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Microframework Microdot</title>
6 </head>
7 <body>
8   <h1>About me</h1>
9   <p>Ini adalah situs portfolio...</p>
10 </body>
11 </html>

```

- Menambahkan routing about dan fungsi about pada file app.py berikut,

```

1 from microdot_template import render_template
2 from microdot_asyncio import Microdot, Response
3
4 app = Microdot()
5 Response.default_content_type = "text/html"
6
7 @app.route('/')
8 async def index(req):
9     return render_template('index.html')
10
11 @app.route('/about')
12 async def about(req):
13     return render_template('about.html')
14
15 if __name__ == "__main__":
16     app.run(debug=True)

```

Perhatikan tampilan kode program app.py berikut ini,



Satu hal terakhir yang dapat kami tingkatkan sebelum menutup pelajaran ini adalah membungkus tag judul dan paragraf di dalam tag div. Itu akan membuat mereka dikelompokkan di dalam bagian yang sama di dokumen HTML sehingga nantinya kita dapat menerapkan gaya CSS ke bagian itu. Berikut tampilan index.html dan about.html yang diperbarui:

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> 1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="en"&gt; 3  &lt;head&gt; 4      &lt;meta charset="UTF-8"&gt; 5      &lt;title&gt;Microframework Microdot&lt;/title&gt; 6  &lt;/head&gt; 7  &lt;body&gt; 8      &lt;div class="home"&gt; 9          &lt;h1&gt;Home&lt;/h1&gt; 10         &lt;p&gt;Website MicroPython melalui framework Microdot...&lt;/p&gt; 11     &lt;/div&gt; 12 &lt;/body&gt; 13 &lt;/html&gt; </pre> <p style="text-align: center;">index.html</p> | <pre> 1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="en"&gt; 3  &lt;head&gt; 4      &lt;meta charset="UTF-8"&gt; 5      &lt;title&gt;Microframework Microdot&lt;/title&gt; 6  &lt;/head&gt; 7  &lt;body&gt; 8      &lt;div class="about"&gt; 9          &lt;h1&gt;About me&lt;/h1&gt; 10         &lt;p&gt;Ini adalah situs portofolio...&lt;/p&gt; 11     &lt;/div&gt; 12 &lt;/body&gt; 13 &lt;/html&gt; </pre> <p style="text-align: center;">about.html</p> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Menjalankan skrip Python harus menampilkan hasil yang sama seperti sebelumnya saat membuka dua URL. Keren, tapi ada sesuatu yang hilang yang mungkin sudah mulai Anda perhatikan, dan itu adalah menu navigasi situs web.

#### Menambahkan menu navigasi ke situs web

Pada titik ini, kami memiliki dua halaman HTML dan skrip Python yang merender halaman HTML tersebut ke dua URL yang berbeda – Home dan About URL. Sekarang kita perlu menambahkan menu agar pengguna dapat dengan mudah menavigasi halaman web hanya dengan mengklik tautannya. Dua halaman web di atas berbagi area umum, yang terdiri dari header (yaitu, aplikasi web Ardit) dan menu navigasi. Untuk membuat area tersebut di setiap halaman, kita dapat menambahkan kode HTML yang menghasilkan header seperti itu ke setiap file HTML kita. Namun, hal yang lebih cerdas untuk dilakukan adalah membuat template HTML induk yang kodenya dapat diwarisi dari kedua template anak kita. Kemudian kita cukup menautkan ke tata letak induk dari setiap halaman HTML untuk mewarisi kodenya. Kita perlu membuat halaman HTML baru dan menyebutnya seperti layout.html:

|                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> 1  from microdot_template import render_template 2  from microdot_asyncio import Microdot, Response 3 4  app = Microdot() 5  Response.default_content_type = "text/html" 6 7  @app.route('/') 8  async def index(req): 9      return render_template('index.html') 10 11  @app.route('/about') 12  async def about(req): 13      return render_template('about.html') 14 15  if __name__ == "__main__": 16      app.run(debug=True) </pre> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



### Dukungan Asinkron dengan Asyncio

Microdot dapat diperluas untuk menggunakan model pemrograman asinkron berdasarkan paket asyncio. Saat kelas Microdot diimpor dari paket microdot\_asyncio, server asinkron digunakan, dan penanganan dapat didefinisikan sebagai pola yang sama dalam pengembangan aplikasi coroutine.

Perhatikan contoh kode program berikut ini menggunakan coroutine asyncio untuk konkurensi,

```
1 from microdot_asyncio import Microdot
2 app = Microdot()
3
4 @app.route('/')
5 async def hello(request):
6     return 'Hello, world!'
7
8 if __name__ == "__main__":
9     app.run(debug=True)
```

### HTML-Render Template

Banyak aplikasi web menggunakan template HTML untuk merender konten ke klien. Microdot menyertakan ekstensi untuk merender template dengan paket utemplate di CPython dan MicroPython, dan dengan Jinja hanya di CPython.

#### Menggunakan Engine uTemplate

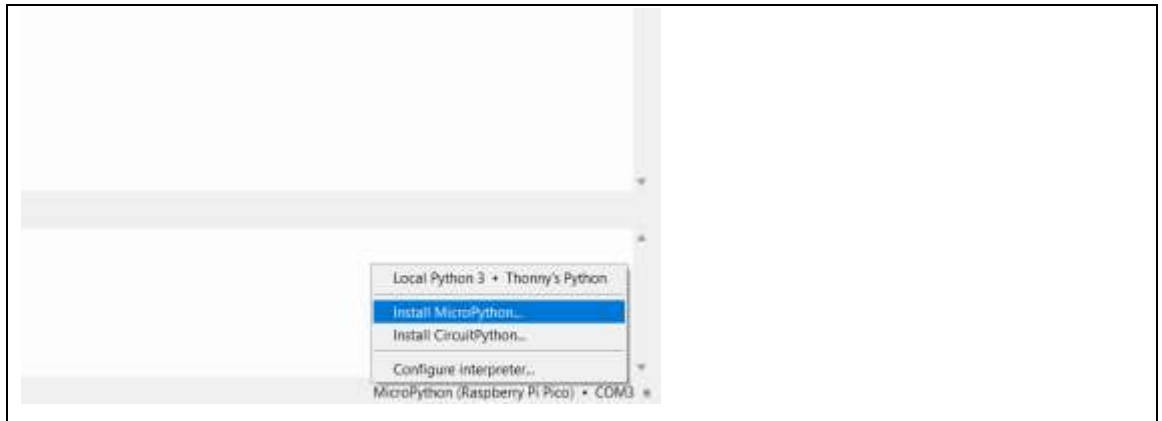
Fungsi render\_template digunakan untuk merender template HTML dengan mesin uTemplate. Argumen pertama adalah nama file template, relatif terhadap direktori templates, yang merupakan template secara default. Argumen tambahan lainnya adalah diteruskan ke engine template untuk digunakan sebagai argument.

#### Menginstall Framework Microdot dan Engine uTemplate

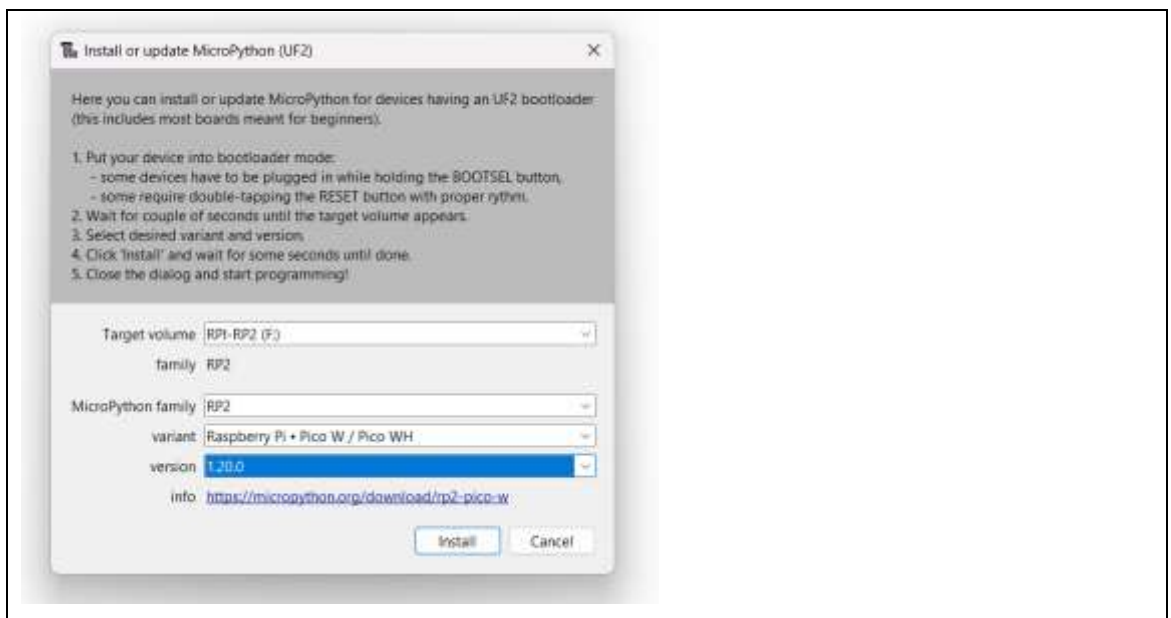
Menginstall framework Microdot dan Engine uTemplate bisa dilakukan dengan menggunakan Thonny dan Pycharm dari windows. Untuk melakukan instalasi nya, bisa dilihat caranya dibawah ini:

##### A. Menginstall Microdot dan uTemplate pada Thonny

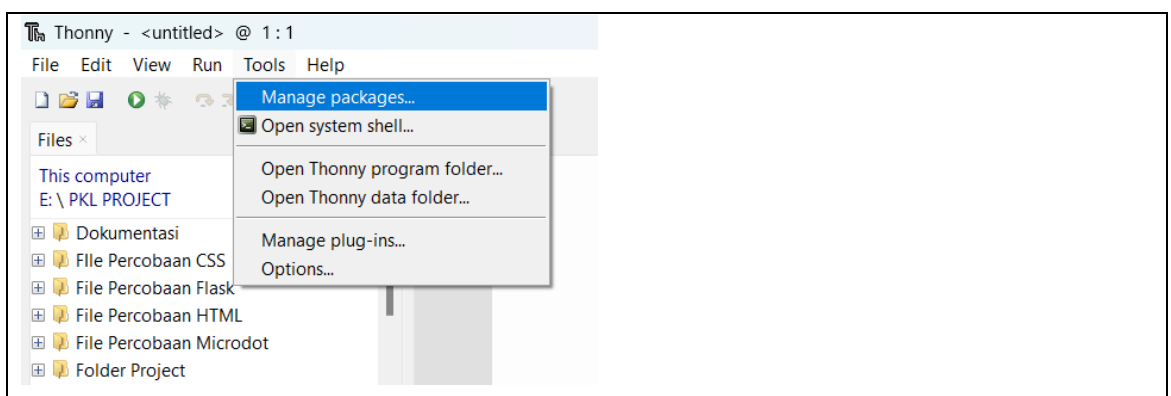
1. Sambungkan Pico-W ke komputer, dan buka Thonny
2. Lihat tampilan kanan bawah pada Thonny, klik dan pilih install MicroPython

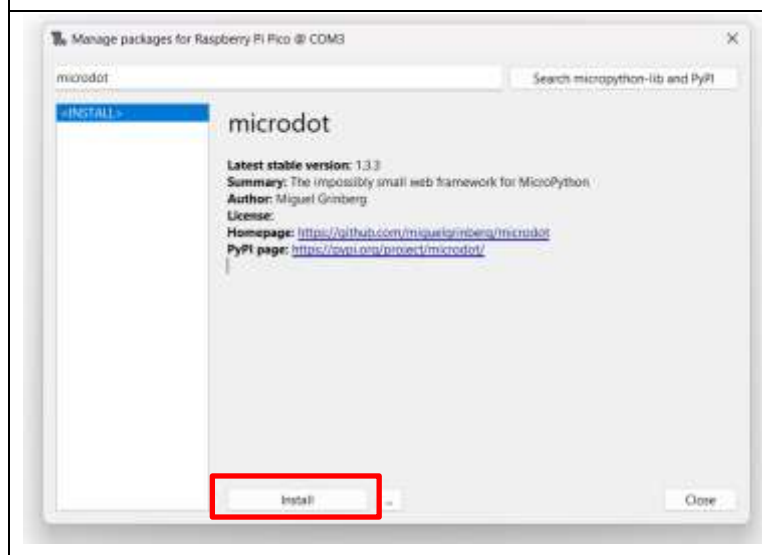
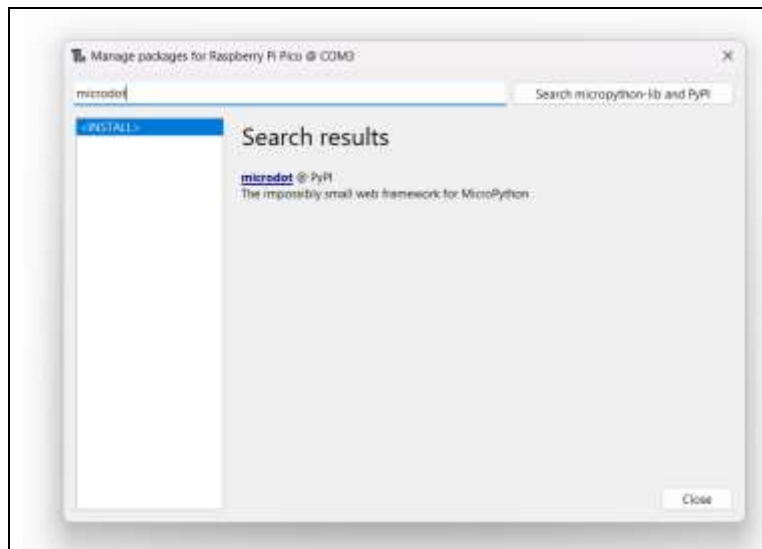


3. Pilih target volume, dan variant untuk Pico-W serta pastikan memilih versi yang terbaru

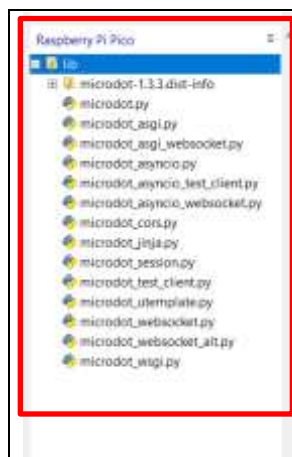


4. Setelah berhasil menginstall MicroPython, selanjutnya lakukan instalasi framework Microdot untuk dijalankan di Pico-W

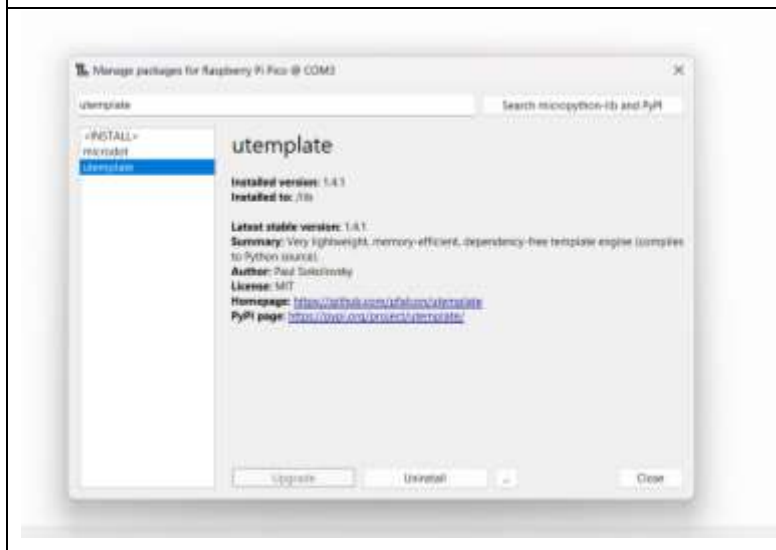
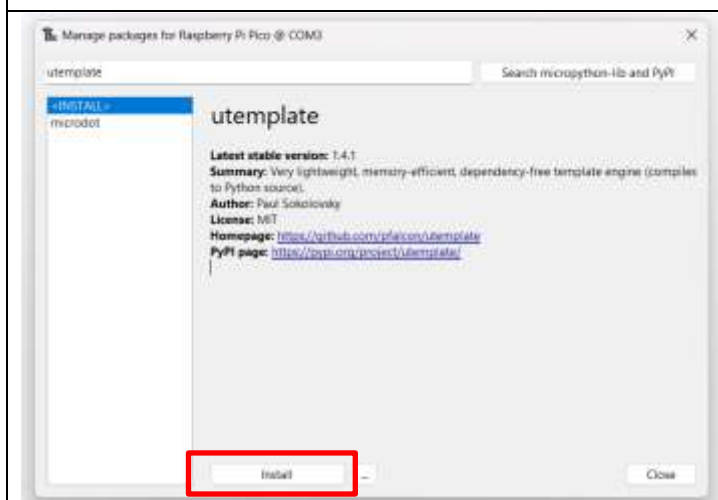
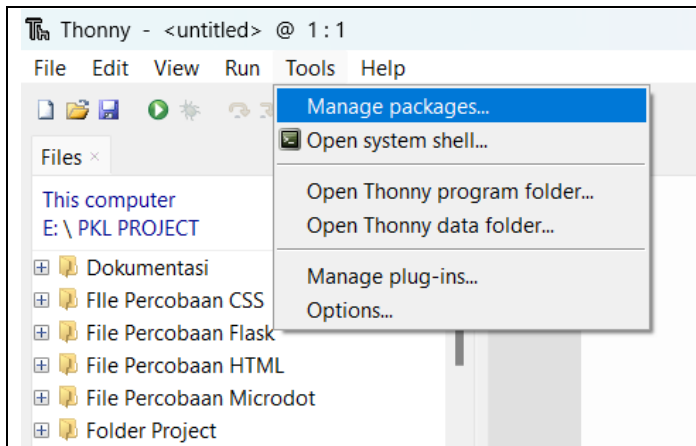




5. Setelah berhasil menginstall Microdot, cek apakah Microdot sudah terinstall di library yang ada di tab kiri



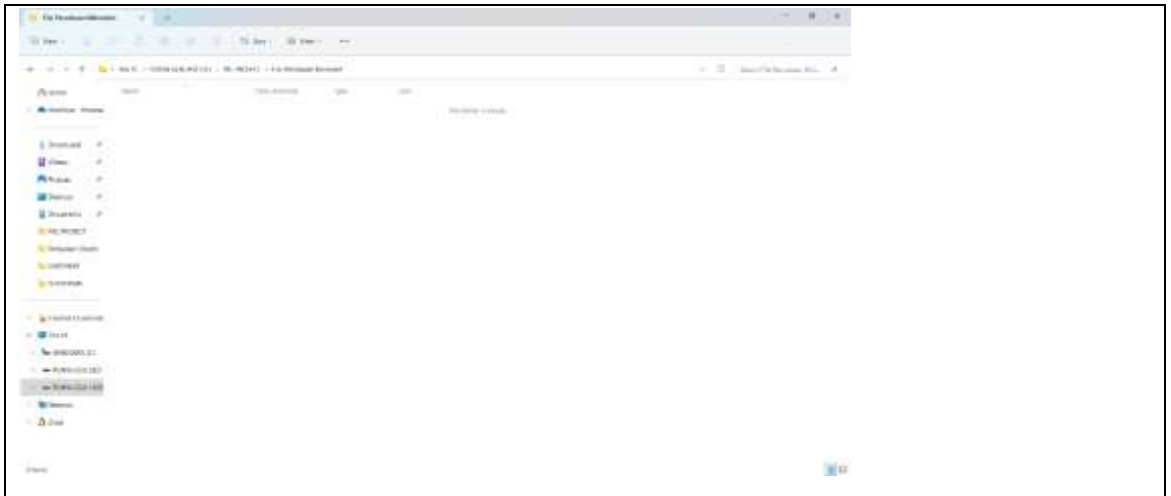
6. Setelah itu dilanjutkan untuk menginstall Engine uTemplate di Pico-W



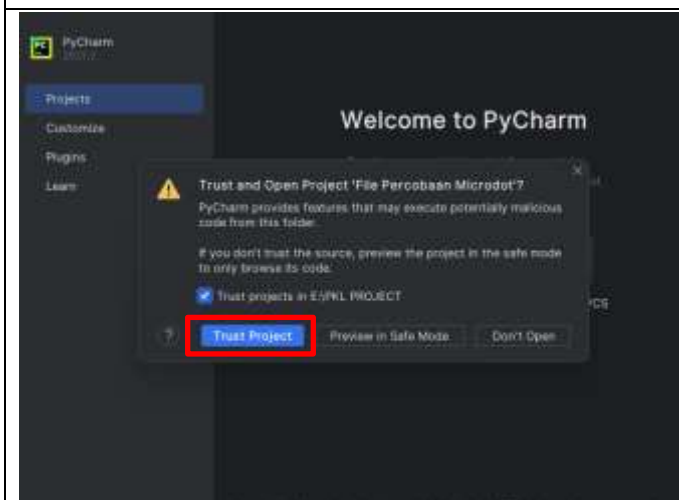
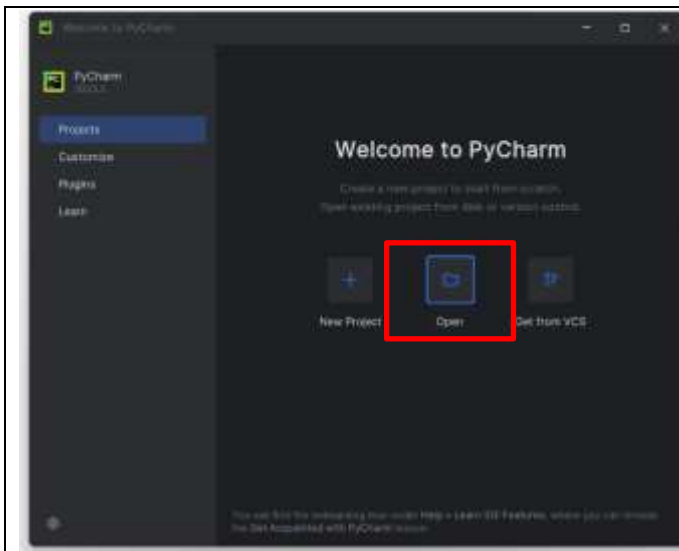


## B. Menginstall uTemplate pada Pycharm

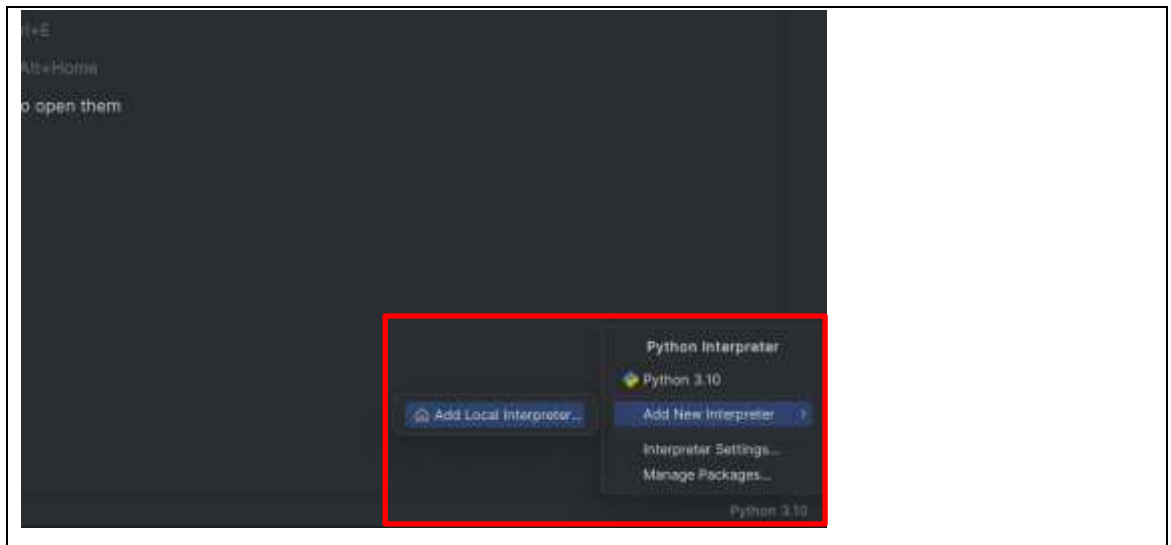
1. Buat folder bebas dan kosong untuk meletakkan projek baru yang akan digunakan



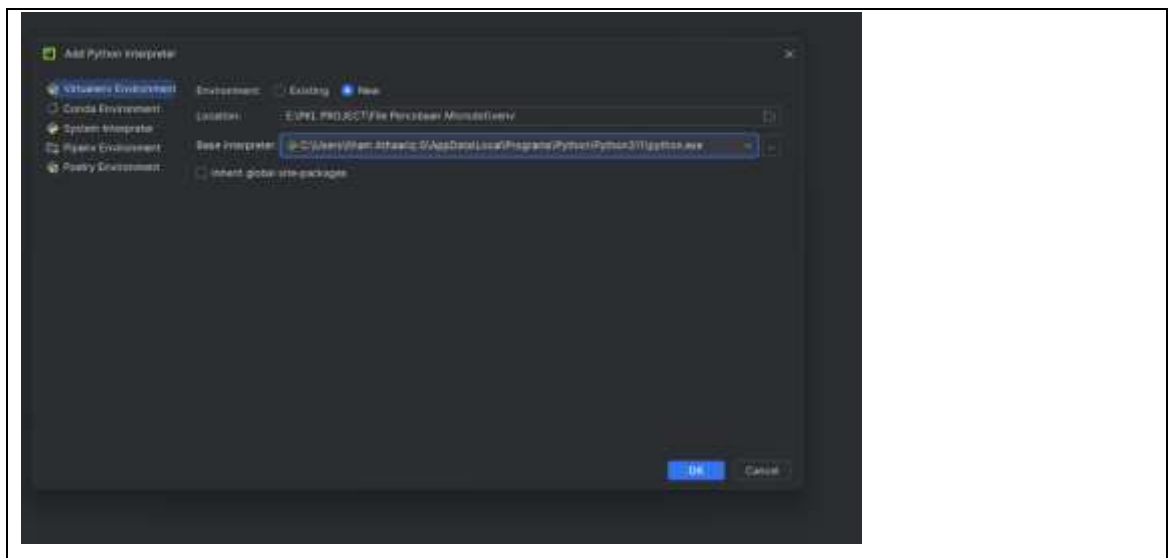
2. Buka Pycharm, dan pilih folder yang dibuat sebelumnya. Lalu berikan izin akses pycharm untuk mengelola folder tersebut



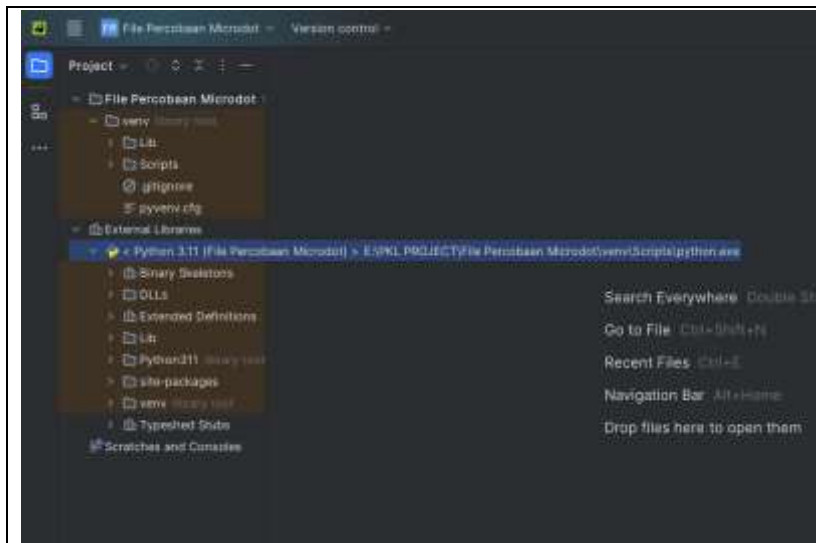
- Setelah halaman utama terbuka, buat terlebih dahulu local python virtual environment untuk dijalankan oleh pycharm dengan menambahkan interpreter baru di kanan bawah



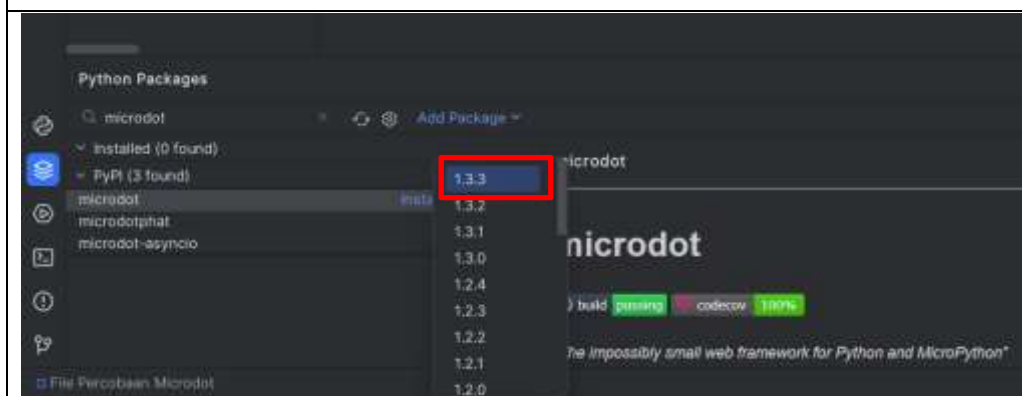
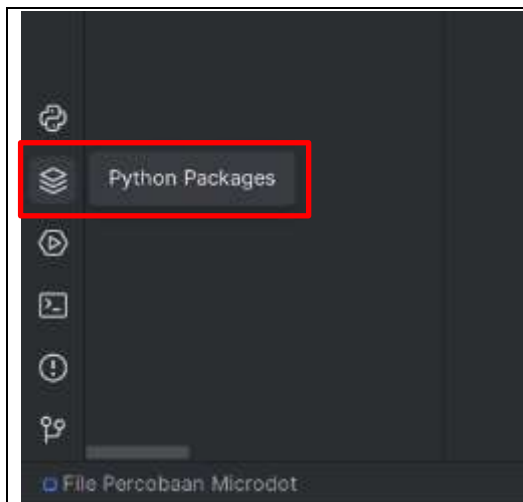
- Buat virtual environment baru, dan pilih base interpreter sesuai dengan python yang terinstall pada windows



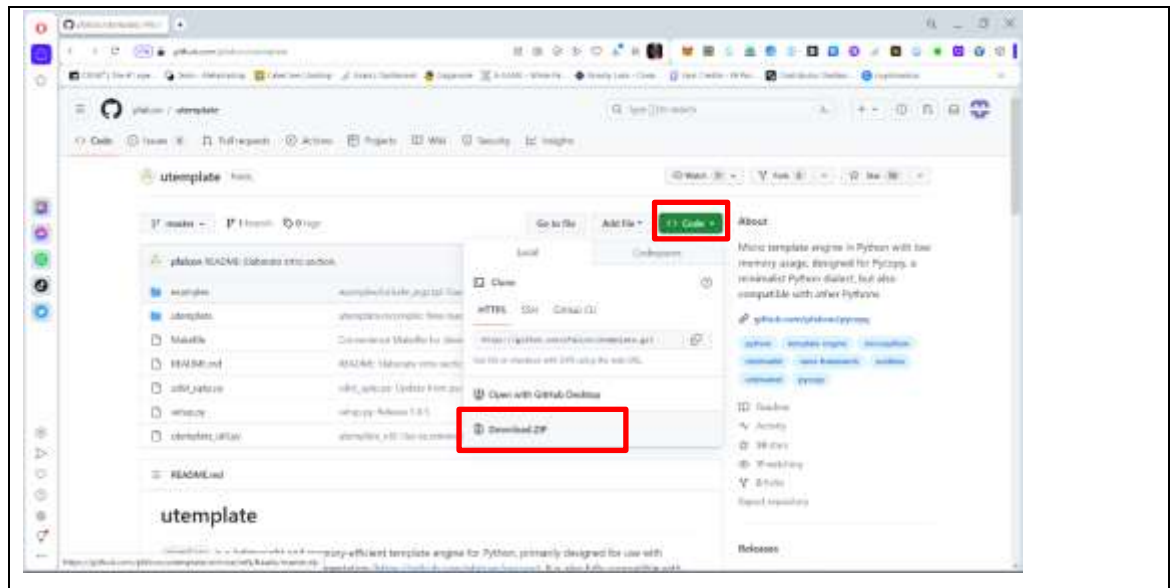
5. Setelah terinstall, maka akan muncul folder venv pada folder yang dibuat tadi, dan pada external library akan terintegrasi dengan folder yang dibuat tadi



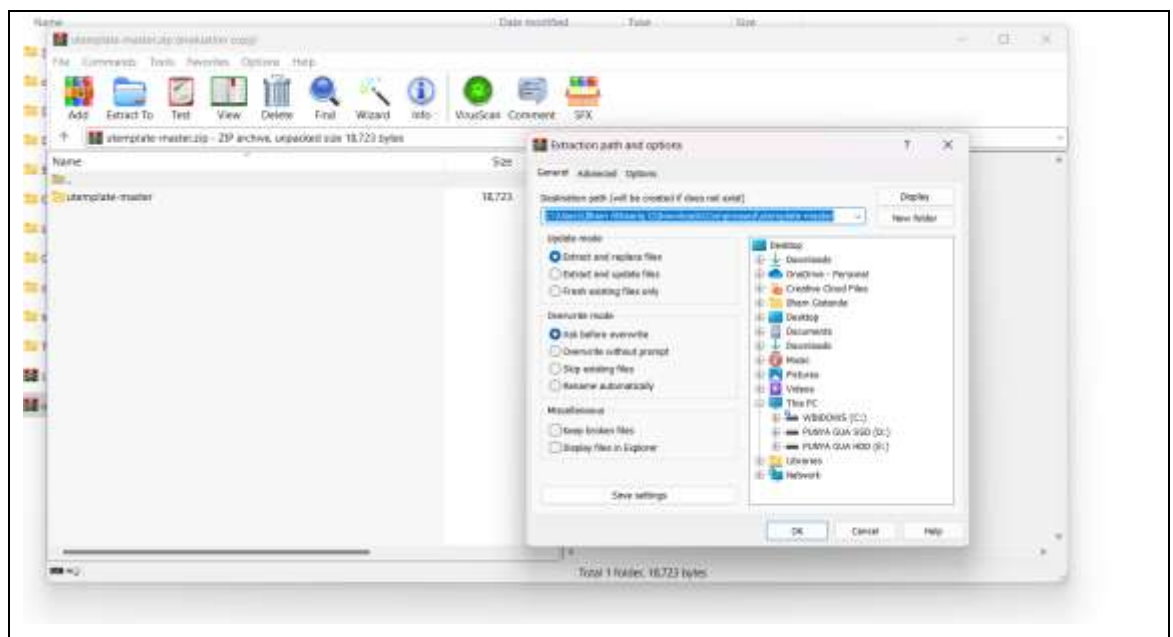
6. Lanjutkan dengan menginstall framework Microdot dari Python Packages yang ada di pycharm dan pilih versi terbaru



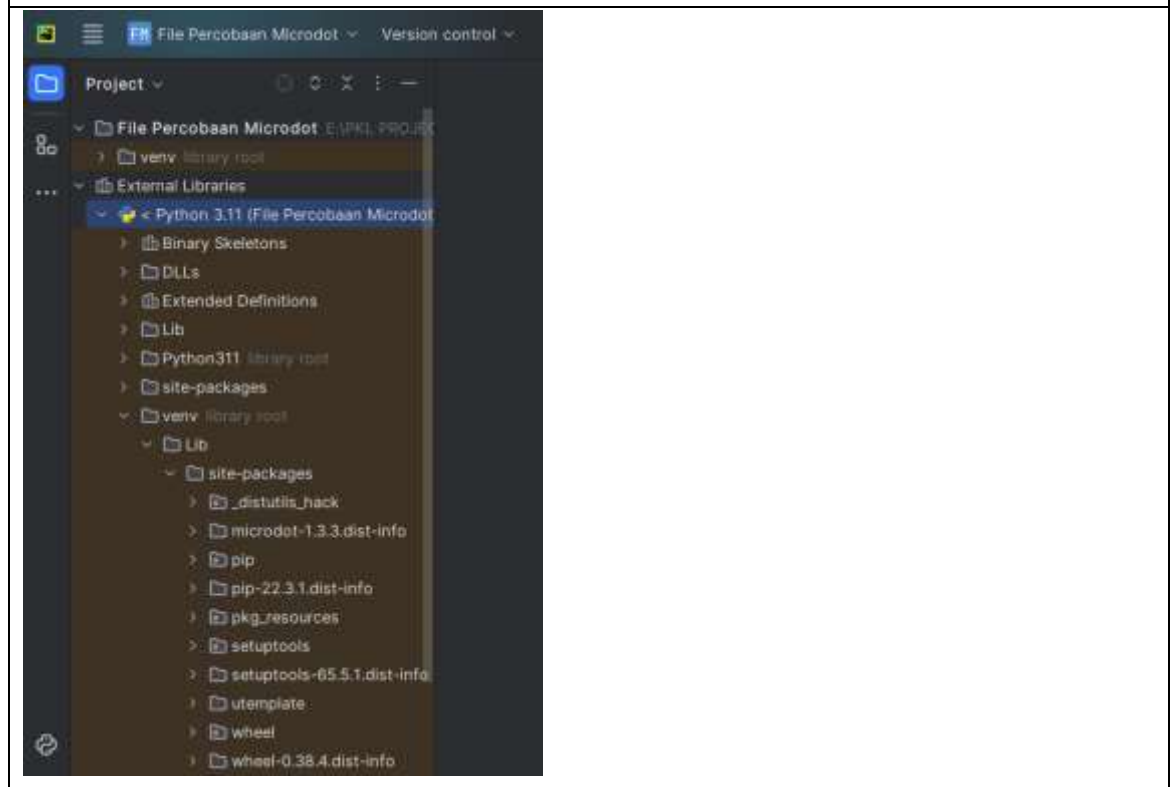
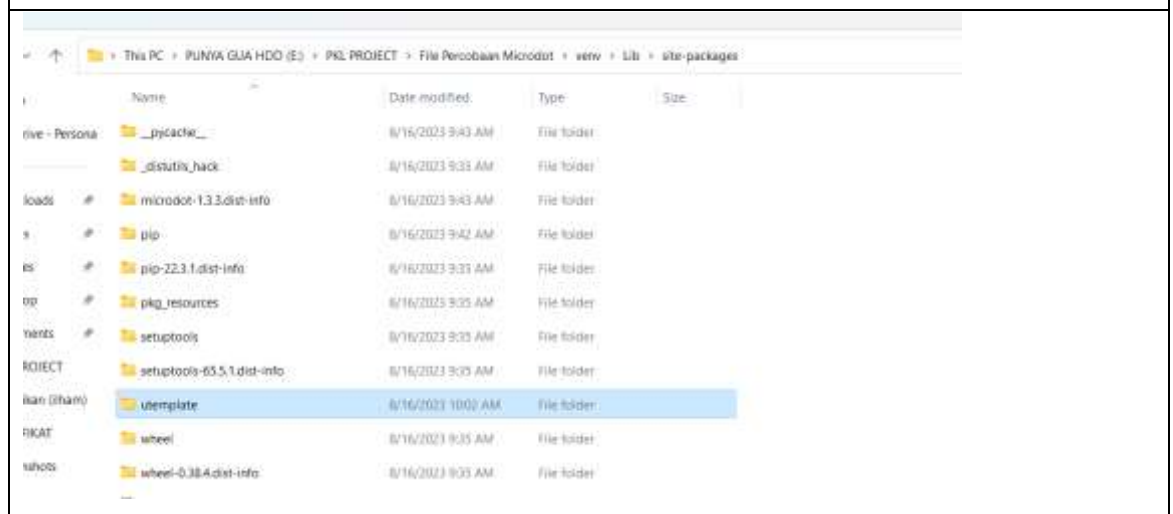
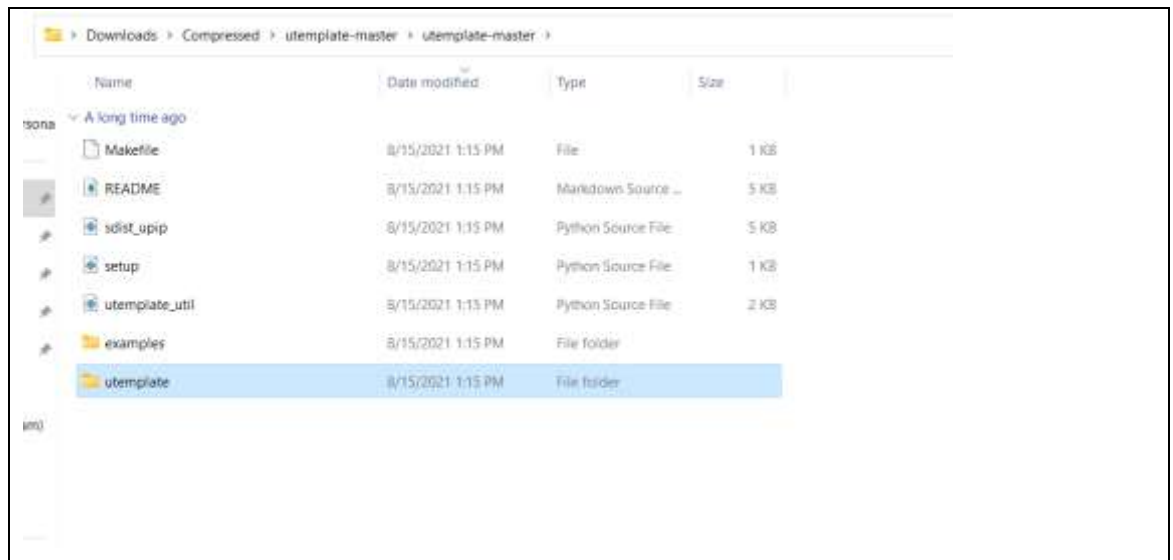
7. Setelah menginstall framework Microdot, lanjutkan dengan menginstall uTemplate. Untuk Engine uTemplate harus diinstal secara manual, download file terlebih dahulu di <https://github.com/pfalcon/utemplate>, dan pilih download ZIP



8. Setelah di download, ekstrak file utemplate-master, buka folder tersebut dan copy folder utemplate ke folder venv > Lib > site-packages di folder proyek yang dibuat







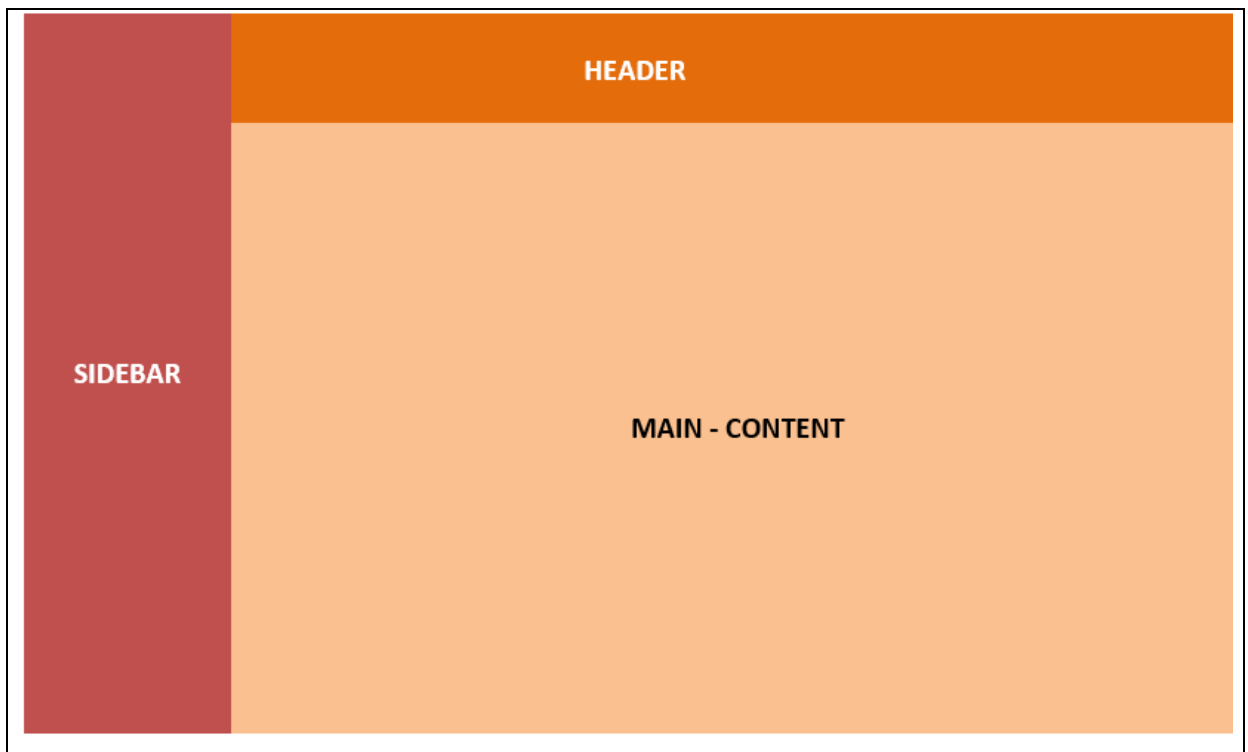
### Membuat Tampilan IoT Web Server Microdot

Membuat Web Server Microdot umumnya menggunakan bahasa pemrograman berbasis HTML (*Hypertext Markup Language*), CSS (*Cascading Style Sheet*), dan JavaScript.

HTML sendiri merupakan bahasa markup standar untuk membuat halaman website. Bahasa ini berfungsi untuk membuat struktur website hingga menyusun format teks dan gambar pada halaman web. Untuk membuat tampilan menjadi menarik perlu menggunakan CSS dalam implementasinya, CSS sendiri berfungsi untuk membantu para web designer agar dapat mengubah dan menambahkan, baik teks, gambar, hingga latar belakang sebuah halaman HTML. Untuk memberikan efek, animasi atau background yang dinamis, HTML dan CSS umumnya diintegrasikan dengan JavaScript.

### Membuat Tampilan IoT Web Server Microdot [HTML-CSS]

Dalam pembuatan Web Server Microdot untuk bagian awal terlebih dahulu menggunakan HTML dan CSS untuk membuat tampilan nya. Dibawah ini akan diberikan contoh layout yang akan dibuat.



Untuk membuat tampilan seperti diatas, diperlukan pembuatan program menggunakan HTML dan CSS, dibawah ini akan diberikan code yang akan menampilkan layout tersebut.

Untuk pembuatan code HTML dan CSS layout diatas, akan dijabarkan dalam beberapa step untuk mempermudah pembuatannya:

1. Step pertama, yaitu membuat tampilan sidebar

HTML:

```
<? index.html = style.css
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport"
6     content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>IoT Control</title>
9
10   <!-- style css -->
11   <link rel="stylesheet" href="/static/style.css">
12 </head>
13
14 <body>
15   <div class="container">
16     <div class="sidebar">
17       <div class="header">
18         <div class="list-item">
19           <span class="header-sidebar">BBPPMPV BOE</span>
20         </div>
21         <div class="sidebar-gambar">
22           
23         </div>
24       </div>
25     </div>
26   </div>
27 </div>
28
29 </body>
30 </html>
```

CSS:

```
<? index.html = style.css
1 /* Menentukan seluruh elemen HTML dan elemen apa yang implementasikan CSS ini */
2 *{
3   font-family: 'Ubuntu';
4   padding: 5px;
5   margin: 0;
6 }
7
8 /* Container berfungsi sebagai wadah utama dari tampilan HTML dan CSS yang dibuat */
9
10 .container {
11   display: flex; /* properti CSS yang digunakan untuk mengatur tata letak elemen-elemen di dalam sebuah container. */
12   min-height: 100vh; /* properti CSS yang digunakan untuk mengatur tinggi minimum dari suatu elemen atau container */
13 }
14
15 /* Sidebar */
16
17 .sidebar {
18   background-color: #888A39;
19   width: 220px;
20   padding: 5px; /* properti CSS yang digunakan untuk memberikan jarak (ruang kosong) di sekitar konten di dalam elemen HTML. */
21   display: flex;
22   box-sizing: border-box; /* properti CSS yang digunakan untuk mengatur tata letak berdasarkan box model padding dan border */
23   flex-direction: column; /* properti CSS yang digunakan untuk mengatur susunan elemen dalam bentuk vertikal */
24 }
25
26 .sidebar .header .list-item {
27   display: flex;
28   flex-direction: row; /* mengatur arah tata letak menjadi horizontal */
29   align-items: center; /* mengatur penyusunan elemen-elemen secara vertikal tengah */
30   padding: 10px 10px; /* memberikan padding 10 piksel pada atas dan bawah, serta 10 piksel pada kanan dan kiri */
31   border-radius: 5px; /* memberikan sudut melengkung pada elemen sebesar 5 piksel */
32   height: 40px; /* menentukan tinggi elemen menjadi 40 piksel */
33   box-sizing: border-box;
34 }
```

```

20
21 .sidebar .header .sidebar-gambar{
22   display: flex;
23   flex-direction: row;
24   align-items: flex-start;
25   padding: 10px;
26   margin: auto;
27   box-sizing: border-box;
28 }
29
30 .sidebar .header .list-item{
31   color: white;
32 }
33
34 /* MAIN CONTENT */
35 .main-content{
36   background-color: #F0F0F0;
37   flex-grow: 1;
38 }

```

### Analisis Code HTML:

```

1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport"
6     content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>IoT Control</title>
9

```

Bagian pertama diisi dengan pengaturan umum dari HTML5 agar web designer dan client memiliki kecocokan pada browser yang digunakan

```

<!-- style css -->
<link rel="stylesheet" href="/static/style.css">
</head>

```

Bagian kedua berisi Code ini berfungsi untuk mengintegrasikan style.css dengan HTML agar code di CSS dapat ditampilkan dalam bentuk web

```

<div class="container">
  <div class="sidebar">
    <div class="header">
      <div class="list-item">
        <span class="header-sidebar">BBPMPV BOE</span>
      </div>
      <div class="sidebar-gambar">
        
      </div>
    </div>
  </div>
</div>

```

Bagian ketiga merupakan pembuatan tampilan pada web.

1. Class Container berfungsi sebagai dasar struktur elemen yang akan digunakan
2. Class Sidebar digunakan untuk membentuk tampilan



|  |                                                                                                                                                                                                                                                                                                                                               |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>sidebar dan meletakkan beberapa Elemen</p> <p>3. Class List-item digunakan untuk mengatur tata letak gambar dan tulisan yang dimasukkan pada class sidebar</p> <p>4. Class header-sidebar digunakan untuk meletakkan tulisan BBPPMPV BOE pada sidebar</p> <p>5. Class sidebar-gambar digunakan untuk meletakkan logo tutwuri handayani</p> |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

#### Analisis Code CSS:

|                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> 1  /* Deklarasi Seluruh Halaman HTML dan Elemen-nya Mengimplementasikan CSS Ini */ 2  +{ 3  font-family: 'Ubuntu'; 4  padding: 0; 5  margin: 0; 6  } </pre>                                                                                                                                                                                              | <p>Pada bagian pertama, terdapat *{...} yang berfungsi untuk mengimplementasikan fungsi CSS pada seluruh halaman dan elemen pada web</p> |
| <pre> 1  /* Container berfungsi sebagai wadah class HTML dan CSS yang dibuat */ 2  + 3  .container { 4  display: flex; /* properti CSS yang digunakan untuk mengatur tata letak elemen-elemen di dalam sebuah container. */ 5  min-height: 100vh; /* properti CSS yang digunakan untuk mengatur tinggi minimum dari suatu elemen atas container */ 6  } </pre> | <p>Pada bagian kedua, pembuatan class container yang berfungsi untuk menentukan ukuran dari halaman web yang dilihat oleh user</p>       |

Pada bagian ketiga, terdapat pembuatan dari class sidebar. Adapun penggunaan dalam contoh (.sidebar .header .list-item) memiliki fungsi untuk mengelompokkan class dan menerapkan gaya dari fungsi yang diterapkan ke dalam kelompok tersebut

- Setelah step pertama selesai, dilanjutkan ke step kedua membuat main-content dan header

HTML:

```

16 <div class="container">
17
18
19 <div class="sidebar">
20 <div class="header">
21 <div class="list-item">
22 <span class="header-sidebar">BBPPMPV BOE</span>
23 </div>
24 <div class="sidebar-gambar">
25 
26 </div>
27 </div>
28 </div>
29
30 <div class="main-content">
31 <div class="header-utama"><h1>DEPARTEMEN TEKNOLOGI INFORMASI</h1></div>
32 </div>
33
34 </div>
35
36 </body>
37
38 </html>

```


CSS:

```

49 /* MAIN CONTENT */
50 .main-content{
51 background-color: #F9E0BB;
52 flex-grow: 1;
53 }
54
55 /* HEADER UTAMA */
56 .header-utama{
57 background-color: #E1D7C6;
58 color: black;
59 padding: 20px;
60 margin: auto;
61 text-align: center;
62 box-shadow: rgba(0, 0, 0, 0.45) 0px 25px 20px -20px;
63 }

```

## Analisis Code HTML:

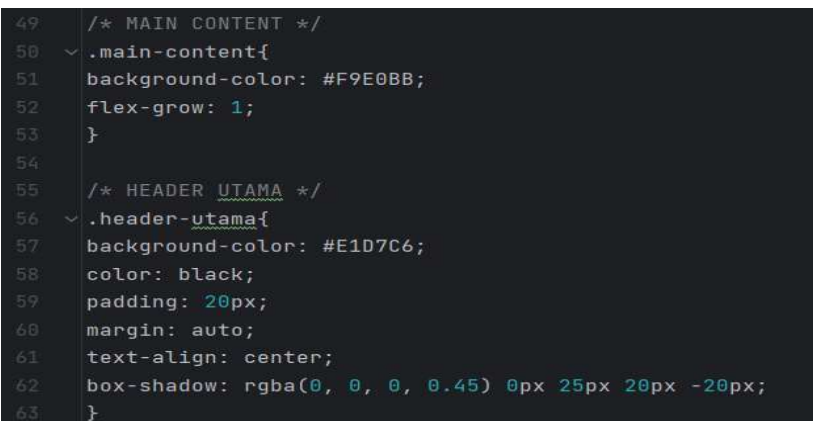


```

14 <div class="container">
15
16   <div class="sidebar">
17     <div class="header">
18       <div class="list-item">
19         <a href="#">BBPMPy BOE</a>
20       </div>
21     </div>
22   </div>
23
24   <div class="main-content">
25     <div class="header-utama">BI-DEPARTEMEN TEKNOLOGI INFORMASI</div>
26   </div>
27 </div>
  
```

Code tersebut dibagi menjadi 3, class container sebagai induk, class sidebar untuk tampilan samping dan class main-content sebagai tempat untuk meletakkan header utama dan elemen yang akan digunakan selanjutnya diposisi tampilan utama

## Analisis Code CSS:



```

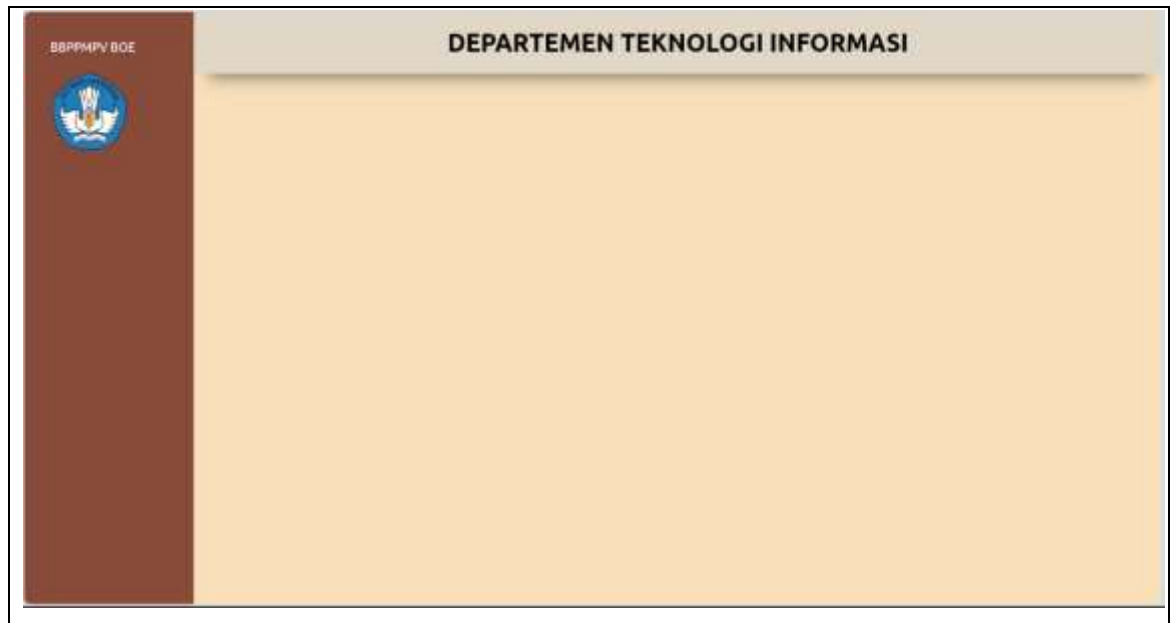
49 /* MAIN CONTENT */
50 .main-content{
51   background-color: #F9E0BB;
52   flex-grow: 1;
53 }
54
55 /* HEADER UTAMA */
56 .header-utama{
57   background-color: #E1D7C6;
58   color: black;
59   padding: 20px;
60   margin: auto;
61   text-align: center;
62   box-shadow: rgba(0, 0, 0, 0.45) 0px 25px 20px -20px;
63 }
  
```

Pada code main-content berisi code yang akan memberikan warna menggunakan Hex Number, dan fungsi (flex-grow: 1;) sebagai fungsi untuk memberikan fleksibilitas sesuai dengan tampilan utama yang diinginkan.

Pada header-utama berisi code yang memberikan posisi yang diinginkan dan memberi sedikit perbaikan tampilan seperti penambahan shadow box



Dengan code yang disusun pada step pertama dan kedua, akan menampilkan layout sebagai berikut:



3. Pada step pertama dan kedua sudah diberikan cara membuat tampilan layout dasar untuk membuat web server, di step ketiga akan diberikan contoh mengisi konten pada layout main-content

HTML:

```

30     <div class="main-content">
31         <div class="header-utama"><h1>DEPARTEMEN TEKNOLOGI INFORMASI</h1></div>
32         <div class="card-container">
33             <div class="main-card">
34                 <h2>INPUT</h2>
35             </div>
36             <div class="main-card">
37                 <h2>OUTPUT</h2>
38             </div>
39         </div>
40     </div>
41 
```

## CSS:

```

65  /* CARD */
66  .card-container{
67    display: grid;
68    position: relative;
69    top: 5%;
70    left: 20%;
71    width: 60%;
72    grid-template-columns: repeat(2, 1fr);
73    gap: 20px;
74    margin-top: 20px;
75  }
76
77  .main-card{
78    background-color: #FFC26F;
79    padding: 15px;
80    border-radius: 20px;
81    margin-left: 10px;
82    margin-right: 10px;
83    height: 150px;
84    box-shadow: rgba(50, 50, 93, 0.25) 0px 6px 12px -2px, rgba(0, 0, 0, 0.3) 0px 3px 7px -3px;
85  }
86
87  .card-container h2{
88    font-size: 20px;
89    margin: auto;
90    text-align: center;
91  }

```

## Analisis Code HTML:

```

60  <div class="main-content">
61    <div class="header-utama"><h1>DEPARTEMEN TEKNOLOGI INFORMASI</h1></div>
62    <div class="card-container">
63      <div class="main-card">
64        <h2>INPUT</h2>
65      </div>
66      <div class="main-card">
67        <h2>OUTPUT</h2>
68      </div>
69    </div>
70  </div>

```

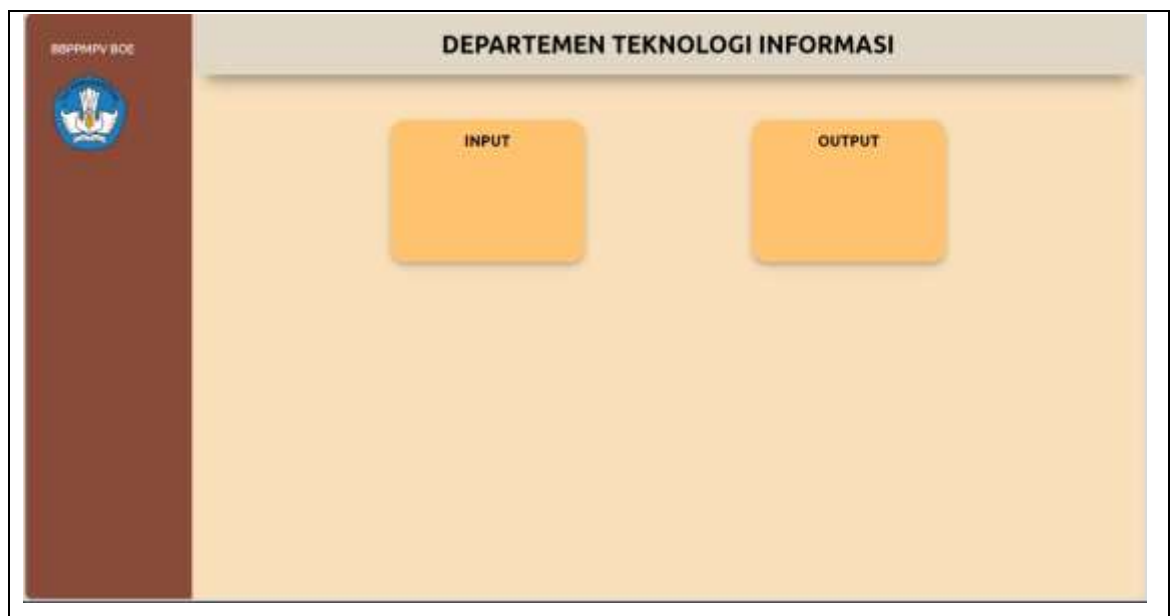
Code tersebut terdiri dari main-content, card-container dan main-card.

1. Main-content sebagai induk tampilan yang akan diisi konten
2. Card-container berfungsi sebagai template untuk meletakkan elemen card
3. Main-card berfungsi untuk membentuk card yang digunakan

### Analisis Code CSS:

|                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> 65  /* CARD */ 66  .card-container{ 67    display: grid; 68    position: relative; 69    top: 5%; 70    left:20%; 71    width: 60%; 72    grid-template-columns: repeat(2,1fr); 73    gap: 200px; 74    margin-top: 20px; 75  } </pre>                                                 | <p>Fungsi dari code tersebut untuk membuat container atau template untuk meletakkan main-card.</p> <p>Penggunaan grid-template-columns dibutuhkan untuk membentuk template yang membagi halaman main-content menjadi 2 kolom secara horizontal</p> |
| <pre> 66  .main-card{ 67    background-color: #FFC26F; 68    padding: 15px; 69    border-radius: 20px; 70    margin-left: 10px; 71    margin-right: 10px; 72    height: 150px; 73    box-shadow: rgba(99, 99, 99, 0.25) 0px 4px 12px -2px, rgba(0, 0, 0, 0.3) 0px 3px 7px -5px; 74  } </pre> | <p>Fungsi code tersebut untuk membentuk card pada template card-container.</p>                                                                                                                                                                     |

Dengan penggabungan code tersebut, dihasilkan tampilan seperti pada gambar:



4. Pada step keempat ini, akan menambahkan elemen button di dalam card input dan memberikan caption ON – OFF di card output

HTML:

```

10 <div class="main-content">
11   <div class="header-utama"><h1>DEPARTEMEN TEKNOLOGI INFORMASI</h1></div>
12   <div class="card-container">
13     <div class="main-card">
14       <h2>INPUT</h2>
15       <div class="button-container">
16         <button type="button" name="button-settings" id="Hijau1" class="button button1" value="on1">ON
17         </button>
18         <button type="button" name="button-settings" id="Merah1" class="button button2" value="off1">
19           OFF
20         </button>
21       </div>
22     </div>
23     <div class="main-card">
24       <h2>OUTPUT</h2>
25       <div class="caption"><h3>ON</h3></div>
26     </div>
27   </div>
28 </div>

```

CSS:

```

93 /* BUTTON */
94 .button-container{
95   padding: 20px;
96   display: grid;
97   z-index: 1;
98 }
99
100 .button {
101   border: none;
102   color: white;
103   padding: 8px 8px;
104   border-radius: 8px;
105   text-align: center;
106   text-decoration: none;
107   font-size: 16px;
108   margin: auto;
109   margin-bottom: 5px;
110   width: 98%;
111   cursor: pointer;
112   z-index: 1;
113 }
114 .button1 {background-color: #4CAF50;} /* Green */
115 .button2 {background-color: #f44336;} /* Merah */
116
117 /* CAPTION ON OFF */
118 .main-content .card-container .main-card h3{
119   font-size: 68px;
120   margin-top: 20px;
121   text-align: center;
122   color: green;
123 }

```

## Analisis Code HTML:

|                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> 1 &lt;div class="main-card"&gt; 2   &lt;h2&gt;INPUT&lt;/h2&gt; 3   &lt;div class="button-container"&gt; 4     &lt;button type="button" name="button-settings" id="Hijau" class="button button1" value="on"&gt;ON 5   &lt;/button&gt; 6     &lt;button type="button" name="button-settings" id="Merah" class="button button2" value="off"&gt; 7       OFF 8     &lt;/button&gt; 9   &lt;/div&gt; 10 &lt;/div&gt; </pre> | <p>Code tersebut memberikan ruang pada main-card untuk meletakkan elemen button ON – OFF yang dibuat dalam class dan memakai function button untuk memberikan perintah click</p> |
| <pre> 1   &lt;div class="main-card"&gt; 2     &lt;h2&gt;OUTPUT&lt;/h2&gt; 3     &lt;div class="caption"&gt;&lt;h3&gt;ON&lt;/h3&gt;&lt;/div&gt; 4   &lt;/div&gt; 5 &lt;/div&gt; </pre>                                                                                                                                                                                                                                        | <p>Code tersebut membuat class caption untuk meletakkan ON sebagai penunjuk pada output</p>                                                                                      |

## Analisis Code CSS:

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> 93  /* BUTTON */ 94  .button-container{ 95    padding: 20px; 96    display: grid; 97    z-index: 1; 98  } 99 100 .button { 101   border: none; 102   color: white; 103   padding: 8px 8px; 104   border-radius: 8px; 105   text-align: center; 106   text-decoration: none; 107   font-size: 16px; 108   margin: auto; 109   margin-bottom: 5px; 110   width: 98%; 111   cursor: pointer; 112   z-index: 1; 113 } 114 .button1 {background-color: #4CAF50;} /* Green */ 115 .button2 {background-color: #f44336;} /* Merah */ </pre> | <p>Code tersebut terdiri dari button-container dan button. Button-container berguna sebagai template untuk meletakkan class button di dalam main-card. Untuk class button dibedakan menjadi 2, yaitu button1 dan button2, dengan template dari class button.</p> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



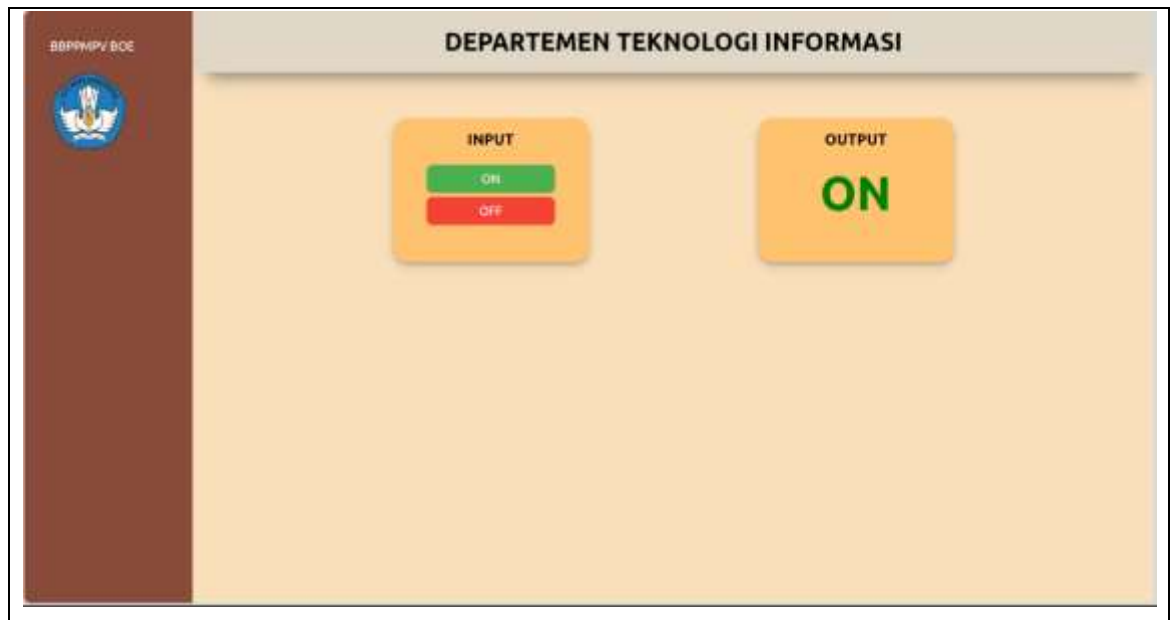
```

117  /* CAPTION ON OFF */
118  .main-content .card-container .main-card .caption h3{
119    font-size: 60px;
120    margin-top: 20px;
121    text-align: center;
122    color: green;
123  }

```

Code tersebut memberikan perintah untuk membentuk tulisan ke class caption sesuai dengan ukuran dan warna yang disiapkan dari code tersebut dengan merubah dari h3 (heading 3)

Dengan penambahan class button dan class caption akan membentuk tampilan seperti gambar berikut:



#### Membuat Tampilan IoT Web Server Microdot [HTML-CSS-JS]

Dalam pembuatan Web Server Microdot selain memerlukan layout dan tampilan pada halaman web, diperlukan juga interaksi antar elemen yang akan digunakan. File JavaScript diletakkan pada folder static. Dibawah ini akan diberikan beberapa contoh penggunaan JavaScript untuk memberikan perintah pada button ON - OFF, mengganti caption sesuai dengan button click yang dipilih, dan menambahkan gauge untuk memberikan informasi yang menarik.

1. Step pertama, membuat button dapat mengeluarkan status ON – OFF pada output

HTML:

```

30 <div class="main-content">
31   <div class="header-wtama"><h1>DEPARTEMEN TEKNOLOGI INFORMASI</h1></div>
32   <div class="card-container">
33     <div class="main-card">
34       <h2>INPUT</h2>
35       <div class="button-container">
36         <button type="button" name="button-settings" id="Hijau1" class="button button1" value="on1"
37           onclick="changeStatus(this.value)">ON
38         </button>
39         <button type="button" name="button-settings" id="Merah1" class="button button2" value="off1"
40           onclick="changeStatus(this.value)">OFF
41         </button>
42       </div>

```

JS:

```

1  /* BUTTON */
2  function changeCaption(value) {
3    var captionElement = document.querySelector('.caption h3');
4    captionElement.textContent = value;
5  }
6  document.getElementById('Hijau1').addEventListener('click', function () {
7    changeCaption('ON');
8  });
9  document.getElementById('Merah1').addEventListener('click', function () {
10   changeCaption('OFF');
11 });

```

Analisis Code HTML:

```

30 <div class="main-content">
31   <div class="header-wtama"><h1>DEPARTEMEN TEKNOLOGI INFORMASI</h1></div>
32   <div class="card-container">
33     <div class="main-card">
34       <h2>INPUT</h2>
35       <div class="button-container">
36         <button type="button" name="button-settings" id="Hijau1" class="button button1" value="on1"
37           onclick="changeStatus(this.value)">ON
38         </button>
39         <button type="button" name="button-settings" id="Merah1" class="button button2" value="off1"
40           onclick="changeStatus(this.value)">OFF
41         </button>
42       </div>

```

Pada code di HTML ini, ada penambahan fungsi onclick="changeStatus(this.value), yang memiliki fungsi untuk memberikan perintah ketika di klik, maka akan mengganti status pada value yang dituju

## Analisis Code JS:

|                                                                                                                                                                                                                             |                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> /* BUTTON */ function changeCaption(value) {     var captionElement = document.querySelector('.caption h3');     captionElement.textContent = value; } </pre>                                                         | <p>Pada bagian JS yang pertama, ada fungsi untuk mengganti value, dengan memanggil atau membaca class dari HTML pada class caption menggunakan document.querySelector, maka JS dapat mengeksekusi pergantian value pada HTML</p> |
| <pre> document.getElementById('Hijau1').addEventListener('click', function () {     changeCaption('ON'); }); document.getElementById('Merah1').addEventListener('click', function () {     changeCaption('OFF'); }); </pre> | <p>Pada bagian kedua, fungsi document.getElementById akan membaca id pada HTML dan menambahkan event click untuk merubah caption ON atau OFF</p>                                                                                 |

Dengan penambahan code JavaScript tersebut, maka akan terjadi perubahan status saat button ditekan, seperti pada gambar:



2. Step kedua, menambahkan temperature gauge meter untuk memberikan informasi tambahan pada web server yang dibuat

HTML:

```
44 <div class="main-card">
45   <h2>OUTPUT</h2>
46   <div class="caption"><h3>ON</h3></div>
47 </div>
48 </div>
49
50 <div class="gauge-container">
51   <div class="gauge-card-1">
52     <h4>TEMPERATURE</h4>
53     <div class="gauge">
54       <div class="gauge__body">
55         <div class="gauge__fill"></div>
56         <div class="gauge__cover"></div>
57       </div>
58     </div>
59   </div>
60
61 </div>
```

CSS:

```

125  /* GAUGE */
126  .gauge-container{
127    display: grid;
128    grid-template-columns: repeat(3,1fr);
129    grid-template-rows: repeat(4,1fr);
130    column-gap: 43px;
131    row-gap: 5px;
132    min-height: 40%;
133    width: 70%;
134    margin-top: 85px;
135  }
136
137  .gauge {
138    width: 100%;
139    position: relative;
140    top: -290px;
141    left: -50px;
142    max-width: 250px;
143    font-size: 35px;
144    color: #004033;
145  }

```

```

147  .gauge__body {
148    width: 100%;
149    height: 0;
150    padding-bottom: 50%;
151    background: #65451F;
152    position: relative;
153    top: 300px;
154    left: 50px;
155    border-top-left-radius: 100% 200%;
156    border-top-right-radius: 100% 200%;
157    overflow: hidden;
158  }
159
160  .gauge__fill {
161    position: absolute;
162    top: 100%;
163    left: 0;
164    width: inherit;
165    height: 100%;
166    background: #FF8551;
167    transform-origin: center top;
168    transform: rotate(0.25turn);
169    transition: transform 0.2s ease-out;
170  }
171

```

```

172 .gauge__cover {
173     width: 75%;
174     height: 150%;
175     background: #CA965C;
176     border-radius: 50%;
177     position: absolute;
178     top: 25%;
179     left: 50%;
180     transform: translateX(-50%);
181
182     /* Text */
183     display: flex;
184     align-items: center;
185     justify-content: center;
186     padding-bottom: 25%;
187     box-sizing: border-box;
188     color: #000000;
189 }
190
191 .main-content h4{
192     font-size: 20px;
193     margin-top: -10px;
194     text-align: center;
195 }

```

```

196 .gauge-card-1{
197     background-color: #CA965C;
198     grid-column: 1;
199     grid-row: 1 / span 2;
200     padding: 20px;
201     margin-left: 10px;
202     margin-right: 10px;
203     width: 100%;
204     box-shadow: rgba(17, 17, 26, 0.1) 0px 4px 16px, rgba(17, 17, 26, 0.1) 0px 8px 24px, rgba(17, 17, 26, 0.1) 0px 16px 56px;
205 }

```

JS:

```

13 /* GAUGE */
14 const gaugeElement = document.querySelector(".gauge");
15 function setGaugeValue(gauge, value) {
16     if (value < 0 || value > 1) {
17         return;
18     }
19     gauge.querySelector(".gauge__fill").style.transform = `rotate(${
20         value / 2
21     }turn)`;
22     gauge.querySelector(".gauge__cover").textContent = `${Math.round(
23         value * 100
24     )}%`;
25 }
26 setGaugeValue(gaugeElement, 0.3);

```



### Analisis Code HTML:

```

50 <div class="gauge-container">
51   <div class="gauge-card-1">
52     <h4>TEMPERATURE</h4>
53     <div class="gauge">
54       <div class="gauge__body">
55         <div class="gauge__fill"></div>
56         <div class="gauge__cover"></div>
57       </div>
58     </div>
59   </div>
60 </div>
61
62

```

Code disamping memberikan template untuk membentuk card dan gauge yang diletakkan di class gauge-container, dan untuk membentuk gauge nya, menggunakan 4 class yaitu gauge, gauge\_\_body, gauge\_\_fill, gauge\_\_cover

### Analisis Code CSS:

```

125 /* GAUGE */
126 .gauge-container{
127   display: grid;
128   grid-template-columns: repeat(3,1fr);
129   grid-template-rows: repeat(4,1fr);
130   column-gap: 43px;
131   row-gap: 5px;
132   min-height: 40%;
133   width: 70%;
134   margin-top: 85px;
135 }

```

Di code css ini, dibagi menjadi 3 section, section pertama membuat gauge-container sebagai template untuk diletakkan di main-content dan sebagai template untuk meletakkan card gauge

```

137  ✓ .gauge {
138      width: 100%;
139      position: relative;
140      top: -290px;
141      left: -50px;
142      max-width: 250px;
143      font-size: 35px;
144      color: #004033;
145  }

```

```

147  ✓ .gauge__body {
148      width: 100%;
149      height: 0;
150      padding-bottom: 50%;
151      background: #65451F;
152      position: relative;
153      top: 300px;
154      left: 50px;
155      border-top-left-radius: 100% 200%;
156      border-top-right-radius: 100% 200%;
157      overflow: hidden;
158  }
159
160  ✓ .gauge__fill {
161      position: absolute;
162      top: 100%;
163      left: 0;
164      width: inherit;
165      height: 100%;
166      background: #FF8551;
167      transform-origin: center top;
168      transform: rotate(0.25turn);
169      transition: transform 0.2s ease-out;
170  }
171

```

Dari code tersebut, yaitu membuat bentuk gauge nya secara manual.

Class gauge digunakan untuk mengatur tulisan / angka pada gauge meter.

Class gauge\_\_body, gauge\_\_fill dan gauge\_\_cover digunakan dalam membentuk setengah lingkaran dan memberikan radius yang dibagi pada sisi kanan dan kiri

```

172 .gauge__cover {
173     width: 75%;
174     height: 150%;
175     background: #CA965C;
176     border-radius: 50%;
177     position: absolute;
178     top: 25%;
179     left: 50%;
180     transform: translateX(-50%);
181
182     /* Text */
183     display: flex;
184     align-items: center;
185     justify-content: center;
186     padding-bottom: 25%;
187     box-sizing: border-box;
188     color: #000000;
189 }
190
191 .main-content h4{
192     font-size: 20px;
193     margin-top: -10px;
194     text-align: center;
195 }

```

```

197 .gauge-card-1{
198     background-color: #CA965C;
199     grid-column: 1;
200     grid-row: 1 / span 1;
201     padding: 10px;
202     margin-left: 10px;
203     margin-right: 10px;
204     width: 100%;
205     height: 100px;
206     border-radius: 10px;
207 }

```

Class gauge-card-1 merupakan class untuk meletakkan posisi card berdasarkan grid yang digunakan untuk meletakkan gauge temperature.

#### Analisis Code JS:

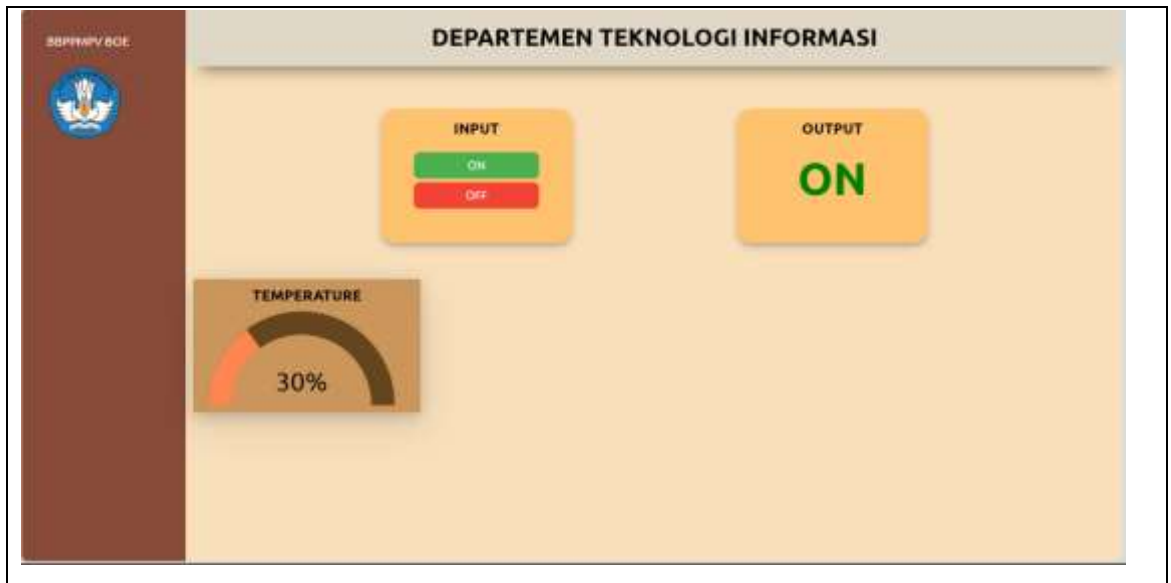
```

13 /* GAUGE */
14 const gaugeElement = document.querySelector(".gauge");
15 function setGaugeValue(gauge, value) {
16     if (value < 0 || value > 1) {
17         return;
18     }
19     gauge.querySelector(".gauge__fill").style.transform = `rotate(${
20         value / 2
21     }turn)`;
22     gauge.querySelector(".gauge__cover").textContent = `${Math.round(
23         value * 100
24     )}%`;
25 }
26 setGaugeValue(gaugeElement, 0.3);

```

Code JS disamping merupakan code fix untuk membentuk posisi radius pada gauge. Untuk merubah angka maksimal pada radius gauge, dapat merubah angka 100 pada code tersebut dengan angka yang diinginkan.

Dengan penambahan code JavaScript tersebut, maka akan terjadi penambahan card gauge seperti gambar dibawah ini:



3. Dengan menggabungkan beberapa code diatas, dapat membuat beberapa card untuk temperature, humidity dan altitude

HTML:

```
<div class="gauge-container">
  <div class="gauge-card-1">
    <h4>TEMPERATURE</h4>
    <div class="gauge">
      <div class="gauge__body">
        <div class="gauge__fill"></div>
        <div class="gauge__cover"></div>
      </div>
    </div>
  </div>
  <div class="gauge-card-2">
    <h4>HUMIDITY</h4>
    <div class="gauge">
      <div class="gauge__body">
        <div class="gauge__fill"></div>
        <div class="gauge__cover"></div>
      </div>
    </div>
  </div>
  <div class="gauge-card-3">
    <h4>ALTITUDE</h4>
    <div class="gauge">
      <div class="gauge__body">
        <div class="gauge__fill"></div>
        <div class="gauge__cover"></div>
      </div>
    </div>
  </div>
</div>
```

## CSS:

```
.gauge-card-1{
background-color: #CA965C;
grid-column: 1;
grid-row: 1 / span 2;
padding: 20px;
margin-left: 10px;
margin-right: 10px;
width: 100%;
box-shadow: rgba(17, 17, 26, 0.1) 0px 4px 16px, rgba(17, 17, 26, 0.1) 0px 8px 24px, rgba(17, 17, 26, 0.1) 0px 16px 56px;
}

.gauge-card-2{
background-color: #CA965C;
grid-column: 2;
grid-row: 1 / span 2;
padding: 20px;
margin-left: 10px;
margin-right: 10px;
width: 100%;
box-shadow: rgba(17, 17, 26, 0.1) 0px 4px 16px, rgba(17, 17, 26, 0.1) 0px 8px 24px, rgba(17, 17, 26, 0.1) 0px 16px 56px;
}

.gauge-card-3{
background-color: #CA965C;
grid-column: 3;
grid-row: 1 / span 2;
padding: 20px;
margin-left: 10px;
margin-right: 10px;
width: 100%;
box-shadow: rgba(17, 17, 26, 0.1) 0px 4px 16px, rgba(17, 17, 26, 0.1) 0px 8px 24px, rgba(17, 17, 26, 0.1) 0px 16px 56px;
}
```

## JS:

```
/* GAUGE */
const gaugeTemperatureElement = document.querySelector(".gauge-card-1 .gauge");
const gaugeHumidityElement = document.querySelector(".gauge-card-2 .gauge");
const gaugeAltitudeElement = document.querySelector(".gauge-card-3 .gauge");

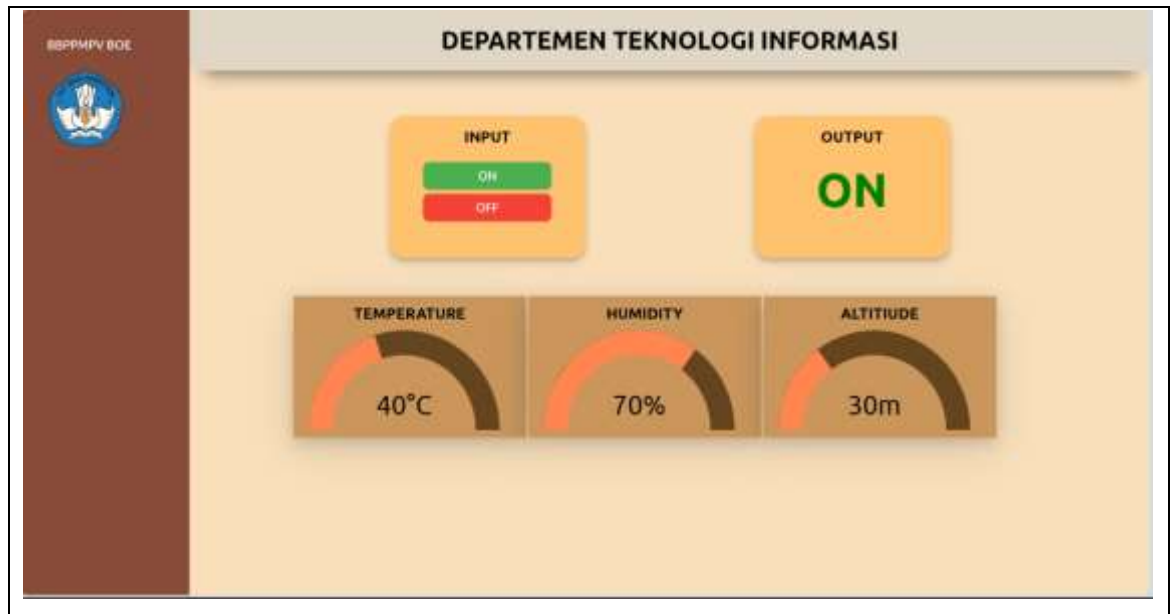
function setGaugeValue(gauge, value, unit) {
  if (value < 0 || value > 1) {
    return;
  }
  gauge.querySelector(".gauge__fill").style.transform = `rotate(${
    value / 2
  }turn)`;
  gauge.querySelector(".gauge__cover").textContent = `${Math.round(
    value * 100
  )}%${unit}`;
}

// Set value for temperature gauge (between 0 and 1)
const temperatureValue = 0.4; // Change this value to the desired temperature value
setGaugeValue(gaugeTemperatureElement, temperatureValue, '°C');

// Set value for humidity gauge (between 0 and 1)
const humidityValue = 0.7; // Change this value to the desired humidity value
setGaugeValue(gaugeHumidityElement, humidityValue, '%');

// Set value for altitude gauge (between 0 and 1)
const altitudeValue = 0.3; // Change this value to the desired altitude value
setGaugeValue(gaugeAltitudeElement, altitudeValue, 'm');
```

Dengan penambahan code JavaScript tersebut, maka akan terjadi penambahan card gauge seperti gambar dibawah ini:



#### Membuat Tampilan IoT Web Server Microdot [HTML-CSS-JS-PYTHON]

Dalam pembuatan Web Server Microdot selain ditambahkan JavaScript untuk memberikan event pada web, diperlukan adanya sambungan antara web dengan servernya. Disini, akan ada penambahan script python yang berisi code untuk menjembatani web dengan server. Untuk koneksi akan menggunakan websocket karena untuk saat ini penggunaan websocket merupakan cara paling stabil dan mudah untuk diimplementasikan. Untuk menggunakan script python ini, file python wajib diletakkan di folder proyek.

1. Untuk membuat percobaan sederhana dengan melakukan koneksi antar script python ke web, bisa dengan melakukan random generate dari gauge meter yang telah dibuat



JS:

```

/* TEMPERATURE GAUGE */

// Fungsi untuk mengambil data aktual temperature dari server
function fetchTemperatureData() {
  return fetch('/updateData')
    .then((response) => response.json())
    .then(data => {
      return data.readingTemp; // Mengambil nilai temperatur dari response server
    })
    .catch(error => {
      console.error("Error fetching temperature data:", error);
      return 0; // Jika terjadi kesalahan, kembalikan nilai default (0)
    });
}

// Fungsi untuk mengubah nilai pada gauge temperature
function updateTemperatureGauge() {
  const gaugeElement = document.querySelector(".gauge-card-1 .gauge");

  // Ambil data aktual temperature dari server menggunakan fungsi fetchTemperatureData()
  fetchTemperatureData()
    .then(temperatureInCelsius => {
      // Normalisasi nilai temperature menjadi range 0 hingga 1 (asumsi maksimal 100 derajat Celsius)
      const normalizedTemperature = temperatureInCelsius / 100;

      // Memperbarui nilai pada gauge temperature
      setTemperatureGaugeValue(gaugeElement, normalizedTemperature);

      // Mengulangi pembaruan setiap 3 detik
      setTimeout(updateTemperatureGauge, 3000);
    });
}

```

```

// Fungsi untuk mengatur nilai pada gauge temperature
function setTemperatureGaugeValue(gauge, value) {
  if (value < 0 || value > 1) {
    return;
  }

  const angle = value * 180; // Convert normalized value to an angle (0 to 180 degrees)
  gauge.querySelector(".gauge__fill").style.transform = `rotate(${angle}deg)`;
  gauge.querySelector(".gauge__cover").textContent = `${Math.round(value * 100)}°C`;
  // Menampilkan nilai temperature dalam derajat Celsius
}

// Memulai pembaruan gauge temperature setiap 3 detik
updateTemperatureGauge();

```

# PYTHON:

```

1  from microdot_asyncio import Microdot, Response, send_file
2  from microdot_uteplate import render_template
3  import ujson
4  import random
5
6
7  app = Microdot()
8  Response.default_content_type = 'text/html'
9
10
11 @app.route('/')
12 async def index(request):
13     return render_template('index.html')
14
15
16 @app.route('/updateData')
17 async def get_sensor_data(request):
18     print("Receive get data request!")
19     sensor_reads_temp = random.uniform(a: 0, b: 100)
20     return ujson.dumps({"readingTemp": sensor_reads_temp,})
21
22
23
24 @app.route('/shutdown')
25 async def shutdown(request):
26     request.app.shutdown()
27     return 'The server is shutting down...'
28
29
30 @app.route('/static/<path:path>')
31 def static(request, path):
32     if '..' in path:
33         # directory traversal is not allowed
34         return 'Not found', 404
35     return send_file('static/' + path)
36
37
38 if __name__ == "__main__":
39     try:
40         app.run(debug=True)
41     except KeyboardInterrupt:
42         pass

```

## Analisis Code JS:

```

/* TEMPERATURE GAUGE */

// Fungsi untuk mengambil data aktual temperature dari server
function fetchTemperatureData() {
  return fetch('/updateData')
    .then((response) => response.json())
    .then(data => {
      return data.readingTemp; // Mengambil nilai temperature dari response server
    })
    .catch(error => {
      console.error("Error fetching temperature data:", error);
      return 0; // Jika terjadi kesalahan, kembalikan nilai default (0)
    });
}

```

Untuk melakukan random generate dari script python, rubah script JS bagian gauge.

Perubahan untuk bagian pertama ini yaitu melakukan koneksi data antara web dengan micropython menggunakan fungsi fetch

```

// Fungsi untuk mengubah nilai pada gauge temperature
function updateTemperatureGauge() {
  const gaugeElement = document.querySelector(".gauge-card-1 .gauge");

  // Ambil data aktual temperature dari server menggunakan fungsi fetchTemperatureData()
  fetchTemperatureData()
    .then(temperatureInCelsius => {
      // Normalisasi nilai temperature menjadi range 0 hingga 1 (skala maksimal 100 derajat Celsius)
      const normalizedTemperature = temperatureInCelsius / 100;

      // Memperbarui nilai pada gauge temperature
      setTemperatureGaugeValue(gaugeElement, normalizedTemperature);

      // Mengulang pembaruan setiap 3 detik
      setTimeout(updateTemperatureGauge, 3000);
    });
}

```

Bagian kedua melakukan pengambilan data gauge dari html untuk class gauge-card-1 dan .gauge

Bagian ketiga mengambil data dari generate random python yang dibuat

```

// Fungsi untuk mengatur nilai pada gauge temperature
function setTemperatureGaugeValue(gauge, value) {
  if (value < 0 || value > 1) {
    return;
  }

  const angle = value * 180; // Convert normalized value to an angle (0 to 180 degrees)
  gauge.querySelector(".gauge__fill").style.transform = `rotate(${angle}deg)`;
  gauge.querySelector(".gauge__cover").textContent = `${Math.round(value * 100)}°C`;
  // Menampilkan nilai temperature dalam derajat Celsius
}

// Memulai pembaruan gauge temperature setiap 3 detik
updateTemperatureGauge();

```

Bagian keempat yaitu fungsi untuk mengatur nilai radius dari gauge meter nya sesuai dengan data random generate yang dipakai.

Dan yang terakhir data akan di update secara dinamis setiap 3 detik sekali

## Analisis Code PYTHON:

|                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                              |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> 1  from microdot_asyncio import Microdot, Response, send_file 2  from microdot_utemplate import render_template 3  import ujson 4  import __random 5 </pre>                                                                                      | <p>Bagian pertama merupakan pemanggilan library yang diperlukan untuk melakukan koneksi.</p>                                                                                                                                                                                                                 |
| <pre> 7  app = Microdot() 8  Response.default_content_type = 'text/html' 9 10 11  @app.route('/') 12  async def index(request): 13      return render_template('index.html') 14 </pre>                                                                 | <p>Bagian kedua membuat objek microdot() menjadi app dan melakukan routing URL menggunakan @app.route untuk request ke index.html</p>                                                                                                                                                                        |
| <pre> 16  @app.route('/updateData') 17  async def get_sensor_data(request): 18      print("Receive get data request!") 19      sensor_reads_temp = random.uniform(a=0, b=100) 20      return ujson.dumps({"readingTemp": sensor_reads_temp}) 21 </pre> | <p>Bagian ketiga, membuat routing URL untuk meminta data melalui /updateData.</p> <p>Di dalam routing ini dimasukkan fungsi random.uniform yang berfungsi untuk mengenerate data antara 0 – 100.</p> <p>Setelah itu data akan disimpan dalam JSON, dan dibaca oleh JS untuk dirubah radius dan value nya</p> |
| <pre> 24  @app.route('/shutdown') 25  async def shutdown(request): 26      request.app.shutdown() 27      return 'The server is shutting down...' </pre>                                                                                               | <p>Fungsi ini adalah endpoint yang digunakan untuk menghentikan server secara aman.</p> <p>Ketika user mengakses URL /shutdown, fungsi</p>                                                                                                                                                                   |

|                                                                                                                                                                                                                                               |                                                                                                                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                                                                               | shutdown akan dijalankan.                                                                                                                                                                 |
| <pre> 29 30 @app.route('/static/&lt;path:path&gt;') 31 def static(request, path): 32     if '..' in path: 33         # directory traversal is not allowed 34         return 'Not found', 404 35     return send_file('static/' + path) </pre> | Ini adalah rute yang menangani permintaan untuk file statis. Routing ini digunakan untuk mengambil dan mengirim file-file seperti gambar, stylesheet, atau skrip JavaScript kepada klien. |
| <pre> 37 38 ▶ if __name__ == "__main__": 39     try: 40         app.run(debug = True) 41     except KeyboardInterrupt: 42         pass </pre>                                                                                                 | Code tersebut merupakan code umum untuk melakukan run script python secara localhost.                                                                                                     |

Dengan perubahan code JavaScript tersebut, dan menambahkan koneksi ke script python, maka akan terjadi random data di gauge meter temperature seperti gambar dibawah ini:



Jika ingin melakukan generate random data, lakukan hal yang sama pada script JS dan python dengan melakukan copy-paste pada temperature gauge dan melakukan perubahan yang sesuai dengan data yang diinginkan. Dan pada script python tambahkan variabel baru untuk membaca data humidity dan altitude di dalam routing /updateData



2. Setelah berhasil melakukan koneksi script python dengan web, pada percobaan kedua akan membuat percobaan sederhana dengan menyambungkan koneksi antara script microdot dan web ke aktuator untuk menyalakan dan mematikan menggunakan button. Koneksi ini akan menggunakan websocket sebagai jembatan koneksi nya.

A. Buat terlebih dahulu script melalui pycharm

JS:

```

/* AKTUATOR WEBSOCKET */
|
var targetUrl = 'ws://$location.host/ws';
var websocket;
const button = document.querySelector("button");

window.addEventListener("load", onLoad);
function onLoad() {
    initializeSocket();
    setDefaultSpeed();
}

function initializeSocket() {
    console.log("Opening WebSocket connection to ESP32 MicroPython Server...");
    websocket = new WebSocket(targetUrl);
    websocket.onopen = onOpen;
    websocket.onclose = onClose;
    websocket.onmessage = onMessage;
}

function onOpen(event) {
    console.log("Starting connection to WebSocket server..");
}

function onClose(event) {
    console.log("Closing connection to server..");
    setTimeout(initializeSocket, 2000);
}

function onMessage(event) {
    console.log("WebSocket message received:", event);
}

function sendMessage(message) {
    websocket.send(message);
}

var buttonSettings = document.querySelectorAll('button[type=button][name="button-settings"]');
buttonSettings.forEach((button) => {
    button.addEventListener("click", () => {
        var btn = button.value;
        console.log("Button Settings :: " + btn);
        sendMessage(btn);
    });
});

function setDefaultSpeed() {
    console.log("Setting default speed to normal..");
    let idM1 = document.getElementById("Merah1");
    let idS1 = document.getElementById("Hijau1");
}

```

#### MAIN.PY:

```
from microdot_asyncio import Microdot, Response, send_file
from microdot_uTEMPLATE import render_template
from microdot_asyncio_websocket import with_websocket
import ujson
import __random
from aktuator import AKTUATOR

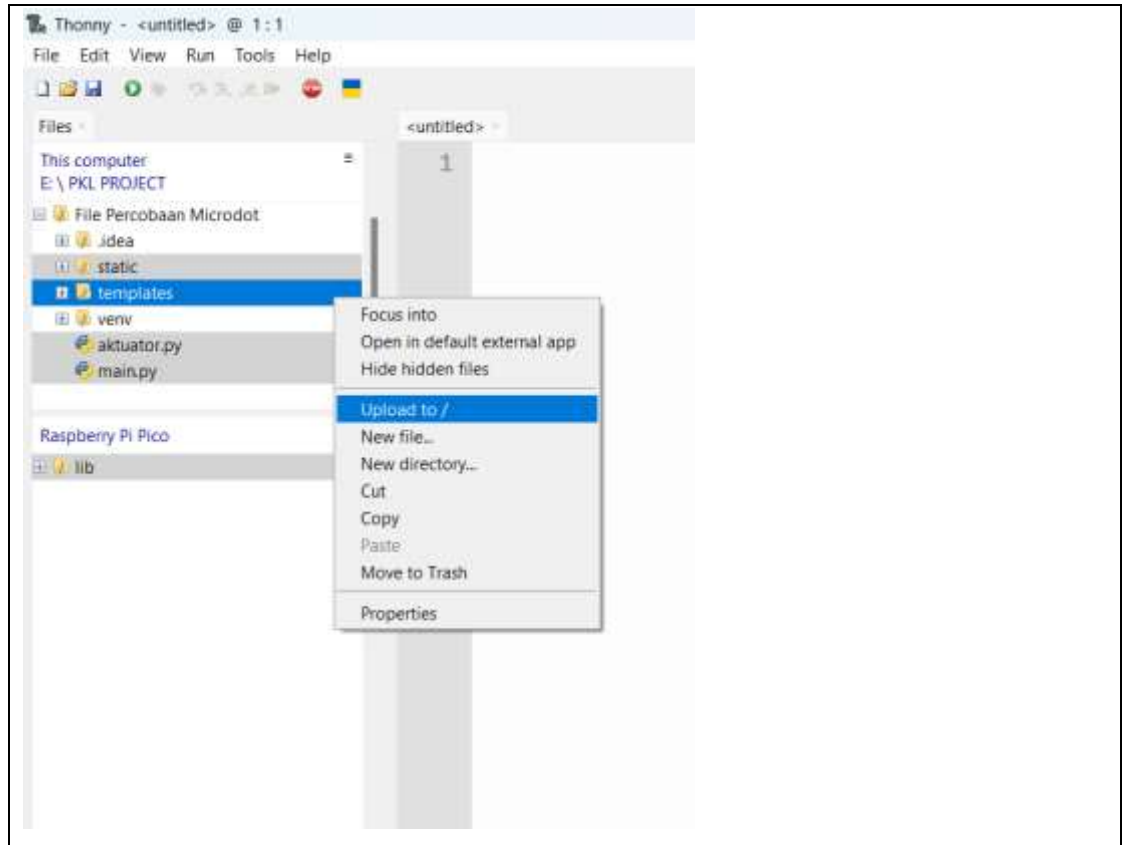
aktuator = AKTUATOR(pin_1=4)
```

```
@app.route("/ws")
@with_websocket
async def kontrolButton(request, ws):
    while True:
        data = await ws.receive()
        if data == 'on1':
            aktuator.on_1()
        if data == 'off1':
            aktuator.off_1()
            status = "OFF"
        await ws.send("OK")
```

#### AKTUATOR.PY:

```
1  from machine import Pin
2
3  2 usages
4  class AKTUATOR:
5      def __init__(self, pin_1):
6          self.pin_1 = Pin(pin_1, Pin.OUT)
7
8      1 usage
9      def on_1(self):
10         self.pin_1.value(1)
11
12         1 usage
13         def off_1(self):
14             self.pin_1.value(0)
```

- B. Setelah memperbarui script, lakukan upload folder static, templates, dan script python yang ada di dalam folder proyek ke pico-W yang telah tersambung di komputer pengguna



- C. Setelah terupload, bikin file baru di dalam thonny yang bernama boot.py, masukkan ssid dan password wifi yang digunakan oleh Pico-W

```

1 # boot.py -- run on boot-up
2 import network, utime
3
4 # Replace the following with your WIFI Credentials
5 SSID = "BOE-"
6 SSI_PASSWORD = "1"
7
8 def do_connect():
9     import network
10    sta_if = network.WLAN(network.STA_IF)
11    if not sta_if.isconnected():
12        print('connecting to network...')
13        sta_if.active(True)
14        sta_if.connect(SSID, SSI_PASSWORD)
15        while not sta_if.isconnected():
16            pass
17        print('Connected! Network config:', sta_if.ifconfig())
18
19 print("Connecting to your wifi...")
20 do_connect()
21
22
  
```

- D. Run script boot.py dan cek pada log status dibawah, lalu masukkan IP yang terdeteksi ke main.py

```

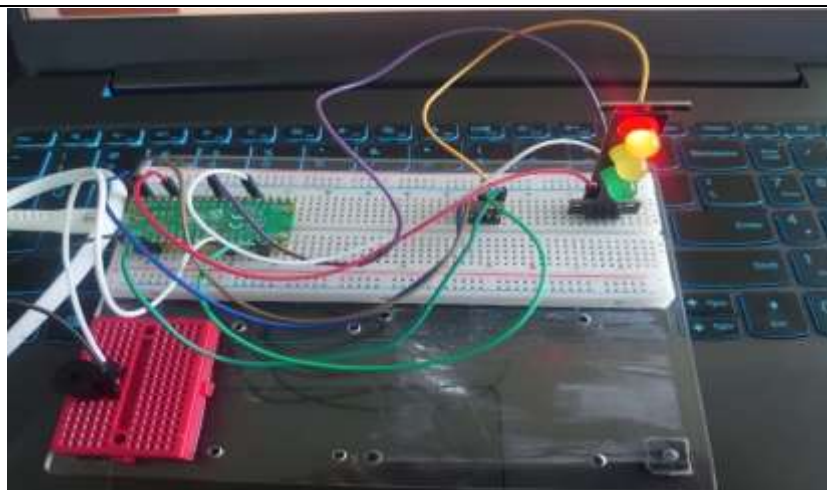
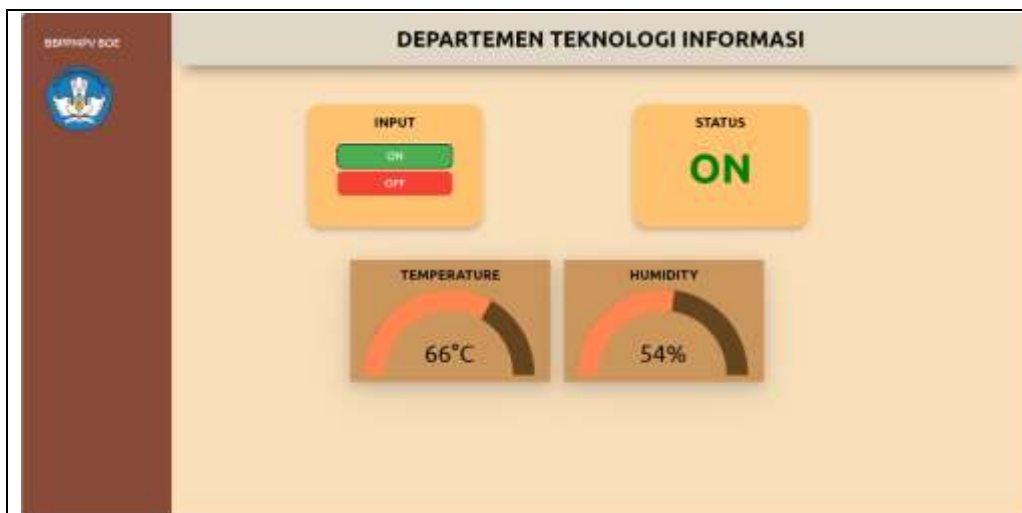
Shell :
$ python3 boot.py
Connecting to your wifi...
connecting to network...
Connected! Network config: ('192.168.56.88', '255.255.255.0', '192.168.56.1', '192.168.56.1')
Connecting to your wifi...
Connected! Network config: ('192.168.56.88', '255.255.255.0', '192.168.56.1', '192.168.56.1')
>>>

60 if __name__ == "__main__":
61     try:
62         app.run(debug = True, host = '192.168.56.88')
63     except KeyboardInterrupt:
64         pass
>>>

Shell :
$ python3 main.py
Connecting to your wifi...
connecting to network...
Connected! Network config: ('192.168.56.88', '255.255.255.0', '192.168.56.1', '192.168.56.1')
Connecting to your wifi...
Connected! Network config: ('192.168.56.88', '255.255.255.0', '192.168.56.1', '192.168.56.1')
>>>

```

- E. Jalankan script main.py dari thonny



## Analisis Code JavaScript:

```
var targetUrl = `ws://${location.host}/ws`;
var websocket;
const button = document.querySelector("button");
```

Di sini, program mendefinisikan tiga variabel. targetUrl adalah URL WebSocket yang akan digunakan untuk terhubung ke perangkat Pico-W. websocket akan menyimpan objek WebSocket yang akan digunakan untuk berkomunikasi. button adalah elemen button pertama yang ditemukan di dokumen.

```
window.addEventListener("load", onLoad);
function onLoad() {
  initializeSocket();
  setDefaultSpeed();
}
```

1. Window.addEventListener yang akan memanggil fungsi onLoad saat halaman web selesai dimuat.
2. Fungsi onLoad dipanggil saat halaman selesai dimuat. Ia memanggil fungsi initializeSocket untuk membuka koneksi WebSocket dan setDefaultSpeed untuk mengatur kecepatan default.

```
function initializeSocket() {
  console.log("Opening WebSocket connection to ESP32 MicroPython Server...");
  websocket = new WebSocket(targetUrl);
  websocket.onopen = onOpen;
  websocket.onclose = onClose;
  websocket.onmessage = onMessage;
}
```

Fungsi ini digunakan untuk menginisialisasi koneksi WebSocket ke server Pico-W. server akan mencetak pesan ke konsol untuk menunjukkan bahwa koneksi sedang dibuka. Selanjutnya, server menetapkan fungsi callback untuk event onopen, onclose, dan onmessage dari WebSocket.

```
function onOpen(event) {
  console.log("Starting connection to WebSocket server..");
}

function onClose(event) {
  console.log("Closing connection to server..");
  setTimeout(initializeSocket, 2000);
}

function onMessage(event) {
  console.log("WebSocket message received:", event);
}

function sendMessage(message) {
  websocket.send(message);
}
```

1. Fungsi onOpen dipanggil saat koneksi WebSocket berhasil dibuka.
2. Fungsi onClose dipanggil saat koneksi WebSocket ditutup.
3. Fungsi onMessage dipanggil saat pesan diterima melalui koneksi WebSocket.
4. Fungsi sendMessage digunakan untuk mengirim pesan melalui koneksi WebSocket ke server.

```
var buttonSettings = document.querySelectorAll("button[type=button][name='button-settings']");
buttonSettings.forEach((button) => {
  button.addEventListener("click", () => {
    var btn = button.value;
    console.log("Button Settings :: " + btn);
    sendMessage(btn);
  });
});

function setDefaultSpeed() {
  console.log("Setting default speed to normal..");
  let idM1 = document.getElementById("Merah1");
  let idM2 = document.getElementById("Hijau1");
}
```

1. Var buttonSettings = ini menangani klik pada tombol-tombol dengan atribut type=button dan name="button-settings". Ketika salah satu tombol diklik, program mencetak pesan ke konsol untuk menunjukkan tombol mana yang diklik dan mengirim pesan yang sesuai melalui WebSocket.
2. Fungsi setDefaultSpeed memiliki fungsi untuk variabel membaca id dari HTML yang dituju



### Analisis Code main.py:

|                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>from microdot_asyncio import Microdot, Response, send_file from microdot_uteplate import render_template from microdot_asyncio_websocket import with_websocket import ujson import random from aktuator import AKTUATOR</pre>                                                                        | <p>Dari code tersebut terlihat adanya library baru yang ditambahkan, yaitu memanggil aktuator.py dengan class AKTUATOR untuk dieksekusi</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <pre>aktuator = AKTUATOR(pin_1=4)</pre>                                                                                                                                                                                                                                                                   | <p>Code tersebut menginisialisasi pin_1 dari class AKTUATOR menjadi variabel aktuator yang dipasang ke GPI.O 4 pada Pico-W</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <pre>@app.route("/ws") @with_websocket async def kontrolButton(request, ws):     while True:         data = await ws.receive()         if data == 'on1':             aktuator.on_1()         if data == 'off1':             aktuator.off_1()             status = "OFF"         await ws.send("OK")</pre> | <ol style="list-style-type: none"> <li>1. App.route mengatur URL mana yang dieksekusi oleh web, di code tersebut /ws merupakan URL yang digunakan</li> <li>2. @with_websocket merupakan decorator custom yang berasal dari library microdot_asyncio</li> <li>3. Async def ... merupakan fungsi yang mengatur koneksi websocket. Dengan meminta permintaan HTTP dari web ke server dengan websocket</li> <li>4. While true merupakan fungsi looping</li> <li>5. Data = await ws.receive() merupakan fungsi untuk menerima data dari web.</li> <li>6. If data == '....' Merupakan kondisi server melakukan checking data yang diterima, jika ON maka mengaktifkan aktuator.on_1 dan jika OFF maka mengaktifkan aktuator.off_1</li> </ol> |

### Analisis Code aktuator.py:

```

1  from machine import Pin
2
3  2 usages
4  class AKTUATOR:
5      def __init__(self, pin_1):
6          self.pin_1 = Pin(pin_1, Pin.OUT)
7
8      1 usage
9      def on_1(self):
10         self.pin_1.value(1)
11
12     1 usage
13     def off_1(self):
14         self.pin_1.value(0)

```

1. Mengambil data dari module machine dan mengimport Pin dari GPI.O Pico-W
2. Membuat class AKTUATOR dan mendefinisikan pin\_1 sebagai variabel GPI.O dari Pico-W yang akan diisi pin output
3. Def on\_1 dan off\_1 merupakan variabel yang dibuat untuk menentukan value dari eksekusi, jika pin\_1.value(1) maka aktif, dan pin\_1.value(0) maka nonaktif

### Membaca Sensor Suhu Pada Pico-W Dengan WebServer

Pico-W memiliki sensor suhu internal yang sudah terintegrasikan dengan salah satu pin ADC (*Analog Digital Converter*) nya. Untuk membaca sensor suhu internal ini, akan dijabarkan code yang harus diterapkan.

1. Tambahkan code berikut ke main.py

#### Code Membaca Suhu:

```

1.  import machine
2.  import utime
3.
4.  sensor_temp = machine.ADC(4)
5.  conversion_factor = 3.3 / (65535)
6.
7.  while True:
8.      reading = sensor_temp.read_u16() * conversion_factor
9.      temperature = 27 - (reading - 0.706)/0.001721
10.     print(temperature)
11.     utime.sleep(2)

```

Setelah dimasukkan di main.py:

```
1  from microdot_asyncio import Microdot, Response, send_file
2  from microdot_utemplate import render_template
3  from microdot_asyncio_websocket import with_websocket
4  import ujson
5  from aktuator import AKTUATOR
6  import machine
7  import utime
```

```
11
12  sensor_temp = machine.ADC(4)
13  conversion_factor = 3.3 / (65535)
```

```
37  @app.route('/updateData')
38  async def get_sensor_data(request):
39      print("Receive get data request!")
40      reading = sensor_temp.read_u16() * conversion_factor
41      temperature = 27 - (reading - 0.706)/0.001721
42      fahrenheit = (temperature * 9 / 5) + 32
43      sensor_reads_cel = temperature
44      sensor_reads_fah = fahrenheit
45      return ujson.dumps(
46          {"readingCel": sensor_reads_cel, "readingFah": sensor_reads_fah})
```

## 2. Rubah code pada JavaScript

### Gauge Temperature Celcius:

```

61  /* CELCIUS GAUGE */
62
63  // Fungsi untuk mengambil data aktual temperature celcius dari server
64  function fetchCelciusData() {
65      return fetch('/updateData')
66          .then((response) => response.json())
67          .then(data => {
68              return data.readingCel; // Mengambil nilai Celcius dari respons server
69          })
70          .catch(error => {
71              console.error("Error fetching Celcius data:", error);
72              return 0; // Jika terjadi kesalahan, kembalikan nilai default (0)
73          });
74  }
75
76  // Fungsi untuk mengubah nilai pada gauge celcius
77  function updateCelciusGauge() {
78      const gaugeElement = document.querySelector(".gauge-card-1 .gauge");
79
80      // Ambil data aktual Celcius dari server menggunakan Fungsi fetchCelciusData()
81      fetchCelciusData()
82          .then(temperatureInCelsius => {
83              // Normalisasi nilai temperature menjadi range 0 hingga 1 (sangat maksimal 100 derajat Celsius)
84              const normalizedTemperature = temperatureInCelsius / 100;
85
86              // Memperbarui nilai pada gauge temperature
87              setCelciusGaugeValue(gaugeElement, normalizedTemperature);
88
89              // Mengulangi pembaruan setiap 5 detik
90              setTimeout(updateCelciusGauge, 3000);
91          });
92  }

```

```

93  // Fungsi untuk mengatur nilai pada gauge Celcius
94  function setCelciusGaugeValue(gauge, value) {
95      if (value < 0 || value > 1) {
96          return;
97      }
98
99      const angle = value * 180; // Convert normalized value to an angle (0 to 180 degrees)
100      gauge.querySelector(".gauge__fill").style.transform = `rotate(${angle}deg)`;
101      gauge.querySelector(".gauge__cover").textContent = `${Math.round(value * 100)}°C`;
102      // Menampilkan nilai temperature dalam derajat Celsius
103  }
104
105  // Memulai pembaruan gauge temperature setiap 3 detik
106  updateCelciusGauge();

```

### Gauge Temperature Fahrenheit:

```

109  /* FAHRENHEIT GAUGE */
110
111  // Fungsi untuk mengambil data aktual humidity dari server
112  function fetchFahrenheitData() {
113    return fetch('/updateData')
114      .then((response) => response.json())
115      .then(data => {
116        return data.readingFah; // Mengambil nilai Fahrenheit dari response server
117      })
118      .catch(error => {
119        console.error("Error fetching Fahrenheit data:", error);
120        return 0; // Jika terjadi kesalahan, kembalikan nilai default (0)
121      });
122  }
123
124  // Fungsi untuk mengubah nilai pada gauge Fahrenheit
125  function updateFahrenheitGauge() {
126    const gaugeElement = document.querySelector(".gauge-card-2 .gauge");
127
128    // Ambil data aktual Fahrenheit dari server menggunakan fungsi fetchFahrenheitData()
129    fetchFahrenheitData()
130      .then(temperatureInFahrenheit => {
131        // Normalisasi nilai Fahrenheit menjadi range 0 hingga 1 (asumsi maksimal 212 °F)
132        const normalizedFahrenheit = temperatureInFahrenheit / 212;
133
134        // Memperbarui nilai pada gauge Fahrenheit
135        setFahrenheitGaugeValue(gaugeElement, normalizedFahrenheit);
136
137        // Mengulang pembaruan setiap 3 detik
138        setTimeout(updateFahrenheitGauge, 3000);
139      });
140  }
141
142  // Fungsi untuk mengatur nilai pada gauge Fahrenheit
143  function setFahrenheitGaugeValue(gauge, value) {
144    if (value < 0 || value > 1) {
145      return;
146    }
147
148    const angle = value * 180; // Convert normalized value to an angle (0 to 180 degrees)
149    gauge.querySelector(".gauge__fill").style.transform = `rotate(${angle}deg)`;
150    gauge.querySelector(".gauge__cover").textContent = `${Math.round(value * 212)}°F`; // Menampilkan nilai Fahrenheit
151  }
152
153  // Mulai pembaruan gauge humidity setiap 3 detik
154  updateFahrenheitGauge();

```

### Analisis Code main.py:

|  |                                                                                                                                                                                      |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <ol style="list-style-type: none"> <li>1. Import module machine dan module utime</li> <li>2. Baris 4 membuat variabel sensor_temp dengan mengambil input dari pin ADC (4)</li> </ol> |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```

1: import machine
2: import utime
3:
4: sensor_temp = machine.ADC(4)
5: conversion_factor = 3.3 / (65535)
6:
7:
8: reading = sensor_temp.read_u16() * conversion_factor
9: temperature = 27 - (reading - 0.706)/0.001721

```

3. Baris 5 merupakan rumus matematika untuk membaca suhu dari internal
4. Baris 8 membuat variabel reading dan membaca data dari sensor\_temp x conversion\_factor
5. Baris 9 merupakan rumus untuk membuat °C

```

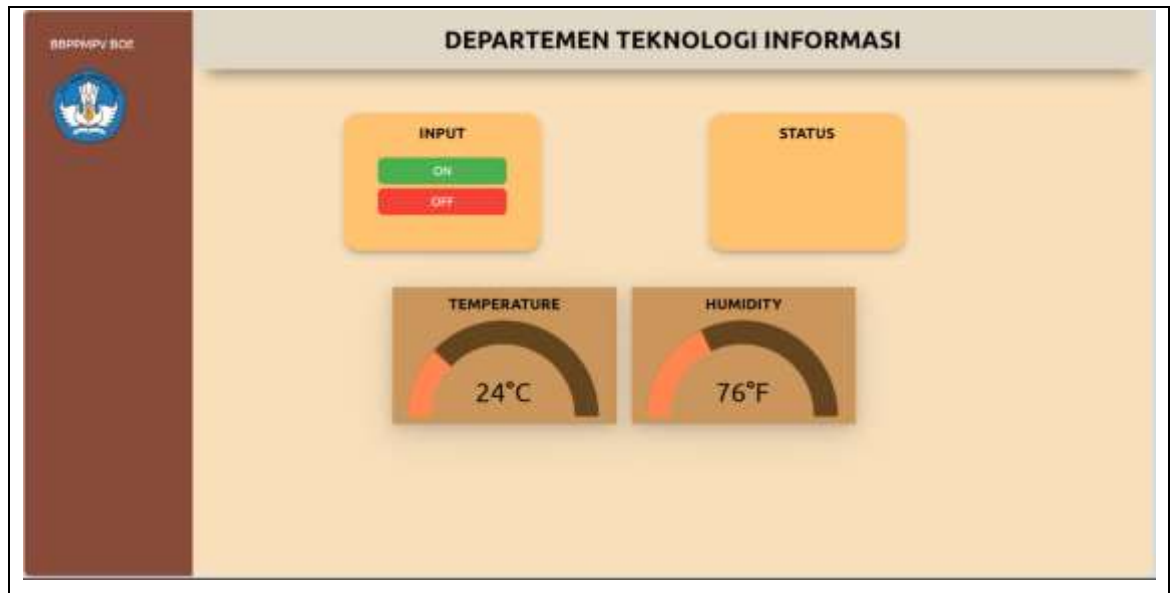
37 @app.route('/updateData')
38 async def get_sensor_data(request):
39     print("Receive get data request!")
40     reading = sensor_temp.read_u16() * conversion_factor
41     temperature = 27 - (reading - 0.706)/0.001721
42     fahrenheit = (temperature * 9 / 5) + 32
43     sensor_reads_cel = temperature
44     sensor_reads_fah = fahrenheit
45     return jsonify({
46         "readingCel": sensor_reads_cel, "readingFah": sensor_reads_fah})

```

Code disamping merupakan routing /updateData yang digunakan untuk merubah value pada gauge meter. Code disamping telah dirubah dari sebelumnya yang merupakan random data, menjadi data aktual dari sensor, dengan penambahan rumus fahrenheit untuk ditampilkan dalam gauge fahrenheit



Setelah merubah dan menambahkan code tersebut, maka sensor sudah dapat dibaca oleh framework Microdot dan ditampilkan dalam bentuk gauge meter celcius dan fahrenheit di webserver IoT yang telah dibuat. Untuk gambar nya bisa dilihat dibawah ini:



Untuk mempermudah pembelajaran secara langsung, file yang telah dibuat dapat diunduh di:

[https://drive.google.com/drive/folders/1-Xb1nUmnD6uEnF5qGrpiWf28Tw-T6Ql5?usp=drive\\_link](https://drive.google.com/drive/folders/1-Xb1nUmnD6uEnF5qGrpiWf28Tw-T6Ql5?usp=drive_link)

## DAFTAR PUSTAKA

- [1] Gareth Halfacree, Ben Everard (2021), Get started with MicroPython on Raspberry Pi Pico, Raspberry Pi Trading Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4
- [2] Richard Blum (2014), Python Programming for Raspberry Pi in 24 Hours, Sams Teach Yourself.
- [3] <https://www.raspberrypi.org/documentation/usage/python/>
- [4] <https://www.programiz.com/python-programming>
- [5] <https://how2electronics.com/>
- [6] <https://carakami.com/sistem-sensor/>
- [7] <https://www.niagahoster.co.id/blog/html-adalah/>
- [8] <https://www.biznetgio.com/news/apa-itu-css>
- [9] <https://www.hostinger.co.id/tutorial/apa-itu-javascript#:~:text=Web%20developer%20umumnya%20menggunakan%20JavaScript,serta%20mengatur%20layout%20halaman%20website.>
- [10] [https://www.youtube.com/watch?v=FnUkVcQ\\_3CQ](https://www.youtube.com/watch?v=FnUkVcQ_3CQ)
- [11] <https://codepen.io/dcode-software/pen/zYGVXyX>
- [12] <https://iotprojectsideas.com/read-internal-temperature-sensor-value-from-raspberry-pi-pico/#:~:text=The%20internal%20temperature%20sensor%20that,ADC%20pins%20support%2012%20bits.>
- [13] <https://how2electronics.com/read-temperature-sensor-value-from-raspberry-pi-pico/>

## PENULIS



Eko Subiyantoro adalah widyaiswara di BBPPMPV BOE-VEDC Malang di Departemen Teknologi Informasi. Lahir di Banyuwangi pada tahun 1975. Pendidikan SD, SMP, dan STM (sekarang SMK) diselesaikan di kota kelahirannya. Selanjutnya ia menyelesaikan S1 di Program Studi Pendidikan Teknik Elektronika FPTK IKIP Yogyakarta (sekarang UNY) pada tahun 1998. Sarjana Sains Terapan (SST) diperolehnya melalui program beasiswa Dikmenjur (sekarang Direktorat Pembinaan Sekolah Menengah Kejuruan) di PENS ITS Surabaya pada tahun 2002 pada Prodi Teknologi Informasi dengan predikat cum laude. Magister Teknik diperolehnya melalui program beasiswa unggulan BPKLN di ITS Surabaya Program Studi Teknik Elektro konsentrasi Jaringan Cerdas Multimedia pada tahun 2009 juga dengan predikat cum laude. Doktor diperolehnya juga melalui beasiswa unggulan bagi pegawai Kemdikbud di Prodi S3 Ilmu Komputer FMIPA Universitas Gadjah Mada.

Pengalaman luar negeri diperolehnya melalui program Inwent dari pemerintah Jerman pada tahun 2003 dalam program Advance Automation Training. Selain itu, pada tahun 2006 mengikuti Expert Program Maintenance and Repair di Abha College Kingdom of Saudi Arabia (KSA). Selanjutnya pada tahun 2013 melalui IGI kerja sama Indonesia-Jerman diikutkan dalam program Modern Management School di Kassel Jerman.

Penulis memiliki pengalaman kerja di PPPPTK BOE/VEDC Malang sebagai Kepala Departemen Teknologi Informasi selama 2 periode 2010–2013 dan 2013–2015. Selain itu juga sebagai tim pengembang Java Education Network Indonesia (JENI), sebuah sistem yang diinisiasi BPKLN Kemdikbud dalam kurun waktu tahun 2007–2009. Sertifikasi yang diperoleh dalam bidang pemrograman adalah Sun Certified Java Programmer (SCJP) pada tahun 2008.


Eko Subiyantoro dapat dihubungi melalui alamat e-mail [ekovedc@gmail.com](mailto:ekovedc@gmail.com)



Ilham Athaariq Gistanda adalah seorang mahasiswa PKL (Praktik Kerja Lapangan) di departemen Teknologi Informasi BBPPMPV BOE - VEDC Malang. Lahir di Denpasar pada tahun 2000. Pernah mengenyam pendidikan dari SD, SMP, dan SMA yang diselesaikan di Sidoarjo dan Malang, Jawa Timur.

Sampai saat ini ditulis pada tahun 2023, ia sedang menempuh pendidikan D4 di perguruan tinggi vokasi Politeknik Negeri Malang dari jurusan Teknik Elektro program studi Jaringan Telekomunikasi Digital pada semester 7.

Ilham Athaariq Gistanda dapat dihubungi melalui alamat e-mail [iathaariq10@gmail.com](mailto:iathaariq10@gmail.com)

|                                                                                     |                                              |                                         |                            |                                        |    |    |    |                |
|-------------------------------------------------------------------------------------|----------------------------------------------|-----------------------------------------|----------------------------|----------------------------------------|----|----|----|----------------|
|  | Bidang Keahlian:<br>Teknologi Informasi      | Laboratorium:<br>RPL-403                | Tanggal:<br>Juli. 2023     | 70                                     | 01 | 02 | VT | Hal<br>61 - 61 |
|                                                                                     | Program Keahlian:<br>Pengembangan PL dan Gim | Komputasi Physical<br>Rapsberry Pi Pico | Rev.Tanggal:<br>27/06/2023 | Dibuat oleh:<br><i>Eko Subiyantoro</i> |    |    |    |                |