

**DOKUMENTASI PROYEK
PEMROGRAMAN BERORIENTASI OBJEK
APLIKASI KONVERSI SUHU**

**Muhammad Hafizh Maulidan – J0304211157
TRK B2**



**PROGRAM STUDI TEKNIK REKAYASA KOMPUTER
SEKOLAH VOKASI
INSTITUT PERTANIAN BOGOR
BOGOR
2022**

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT Yang Maha Esa atas berkat dan rahmat-Nyalah sehingga penulis dapat menyelesaikan Dokumentasi Proyek akhir mata kuliah Pemrograman Berorientasi Objek dengan judul “**APLIKASI KONVERSI SUHU**”.

Adapun tujuan dari penulisan dokumentasi proyek ini adalah untuk menyelesaikan Proyek akhir dari mata kuliah Pemrograman Berorientasi Objek (PBO) pada Sekolah Vokasi Institut Pertanian Bogor dan untuk menyelesaikan mata kuliah PBO pada semester 3 Program Studi Teknologi Rekayasa Komputer.

Pada kesempatan ini, penulis hendak menyampaikan terima kasih kepada semua pihak yang telah memberikan dukungan moral maupun materiil sehingga dokumentasi proyek ini dapat selesai. Banyak ucapan terima kasih ini penulis tujukan kepada:

1. Bayu Widodo S.T., M.T. selaku Dosen yang telah mendidik dan memberikan bimbingan pada mata kuliah Pemrograman Berorientasi Objek (PBO) selama masa Semester 3 ini.
2. Kakak-Kakak Asisten Dosen Mata Kuliah Pemrograman Berorientasi Obyek.
3. Teman-teman anggota kelompok 3 TRK B2.

Meskipun sebaik mungkin penulis mengerjakan laporan ini, penulis tetap menyadari dokumentasi proyek akhir ini masih banyak terdapat kekurangan yang terselip. Maka dari itu, penulis berharap bahwa kritikan dan saran-saran yang diberikan pembaca dapat membangun niat dan semangat untuk menyempurnakan segala yang kurang dari dokumentasi proyek akhir untuk mata kuliah “Pemrograman Berbasis Objek”.

Akhir kata, semoga karya tulis yang diberikan oleh penulis ini menjadi manfaat bagi yang membaca, dan menjadi wawasan tambahan yang dapat memberi manfaat yang sangat baik bagi seluruh pembaca.

Bogor, 13 Desember 2022

Muhammad Hafizh Maulidan

J0304211157

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
DAFTAR GAMBAR.....	iii
I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	1
1.3 Tujuan	1
1.4 Manfaat	2
1.5 Ruang Lingkup Penelitian	2
II PEMBAHASAN	3
2.1 Deskripsi Sistem	3
2.2 Gambaran Arsitektur.....	3
2.3 Komponen Sistem.....	13
2.4 Kode Program	14
2.5 Demo Website.....	18
III KESIMPULAN	21
3.1 Simpulan	21
3.2 Saran	21
DAFTAR PUSTAKA.....	22

DAFTAR GAMBAR

Gambar 1.Menjalankan Aplikasi Django	4
Gambar 2.Menjalankan Base Coding.....	4
Gambar 3.Jalankan Server.....	4
Gambar 4.Menjalankan Proyek Utama	4
Gambar 5.Menambahkan Proyek Utama.....	4
Gambar 6.Membuat sebuah landing page	5
Gambar 7.Melakukan Codingan.....	5
Gambar 8.Memberikan command def Landing Page.....	5
Gambar 9.Menambahkan form dan urlspath	6
Gambar 10.Restart Django	6
Gambar 11.Menguji Landing Page.....	6
Gambar 12.Membuat Web baru	7
Gambar 13.Menambahkan web baru pada urls.py	7
Gambar 14.Menambahkan href pada file html.....	7
Gambar 15.Membuat URL pada website (perhitungan_web).....	7
Gambar 16.Menambahkan fungsi baru pada views.py.....	8
Gambar 17.Mengubah nama html	8
Gambar 18.Header Werb Celcius dan Output	8
Gambar 19.Coding Style Tombol.....	9
Gambar 20.Membuat HTML (perhitungan_fahrenheit.html)	9
Gambar 21.Mengubah nama Web dan label pada form	9
Gambar 22.Tampilan Codingan untuk 4 tombol	10
Gambar 23.Membuat HTML (perhitungan_kelvin.html)	10
Gambar 24.Mengubah nama Web dan Form pada label	11
Gambar 25.Codingan menampilkan 4 tombol.....	11
Gambar 26.Membuat html (perhitungan_reamur.html)	12
Gambar 27.Mengubah nama Web dan label pada form	12
Gambar 28.Codingan 4 Tombol	13
Gambar 29.Pengubahan Struktur Ursl.py	13
Gambar 30.Membuat Urlpatterns	13
Gambar 31.Main Directory Proyekkel3utama.....	14
Gambar 32.Subdirectory proyekkel3.....	14
Gambar 33.Directory Templates/Landing Page	15
Gambar 34.Melakukan Import (untuk render)	15

Gambar 35. Kode Program membuat fungsi (perhitungan_celcius)	17
Gambar 36.Kode Program membuat fungsi (perhitungan_reamur)	18
Gambar 37.Kode Program membuat fungsi (perhitungan_fahrenheit)	20
Gambar 38.Kode Program membuat fungsi (perhitungan_kelvin)	21
Gambar 39.Tampilan Interface Landing Page	22
Gambar 40.Tampilan Interface Main Page Celcius.....	22
Gambar 41.Tampilan Interface Main Page Reamur.....	23
Gambar 42.Tampilan Interface Main Page Fahrenheit	24
Gambar 43. Tampilan Interface Main Page Kelvin.....	25

I PENDAHULUAN

1.1 Latar Belakang

Konversi Suhu adalah untuk menyatakan suhu pada pengukuran satuan skala tertentu agar bisa dibaca menggunakan satuan skala lainnya tanpa harus mengubah ukuran dari suhu tersebut. Terdapat empat skala satuan suhu, yaitu Celcius (C), Reamur (R), Fahrenheit (F), dan Kelvin (K) dengan perbandingan masing-masing skala = 5 : 4 : 9 : 5. Perbandingan ini dapat digunakan untuk rumus Konversi suhu. Khusus untuk satuan skala Fahrenheit harus ditambah atau dikurang 32 dan skala kelvin ditambah atau dikurang 273 (Azly, 2016).

Kemampuan Siswa dalam memahami soal materi konversi suhu sebagian besar masih dibawah standar Kriteria Ketuntasan Minimal (KKM) dengan nilai rata-rata 62,2 dari nilai standar ≥ 64 untuk tuntas KKM. Penyebab kemungkinan terjadi yaitu ketidakmampuan siswa dalam soal hitungan, metode pengajaran yang membosankan, dan belum dimanfaatkannya media yang tepat untuk memahami materi pembelajarannya.(Yusro, 2017)

Sesuai penjelasan yang telah diuraikan diatas, maka penulis tertarik untuk melakukan eksplorasi pada python menggunakan framework Django. Eksplorasi yang dilakukan akan diimplementasikan untuk membuat aplikasi konversi suhu dalam proyek akhir mata kuliah Pemrograman Berorientasi Objek (PBO).

1.2 Perumusan Masalah

Berdasarkan latar belakang dari Tugas proyek akhir semester 3 maka dapat ditetapkan identifikasi masalah sebagai berikut:

1. Bagaimana merancang aplikasi konversi suhu berbasis webberdasarkan Python?
2. Bagaimana implementasi dari aplikasi *web* dengan *framework* Django?
3. Bagaimana membuat sebuah tampilan aplikasi yang menarik danmudah dipahami?
4. Apa saja komponen dari *framework* Django?

1.3 Tujuan

Adapun tujuan dari pembuatan Aplikasi Konversi Suhu, diantara lain:

1. Memahami pembangunan aplikasi *web* berbasis Python.
2. Membangun aplikasi berbasis web yang dapat membantu mengkalkulasikan proses penghitungan konversi skala satuan suhu dengan bahasa Python menggunakan *framework* Django.
3. Memahami komponen dari *framework* Django.

1.4 Manfaat

Manfaat dalam pembuatan Aplikasi Konversi Suhu ini adalah dapat mempermudah dan membantu mengkalkulasikan proses perhitungan konversi skala satuan suhu dengan Bahasa Python menggunakan *Framework Django*.

1.5 Ruang Lingkup Penelitian

Ruang lingkup penelitian ini mencakup proses pembuatan aplikasi dimulai dari instalasi software pendukung hingga software proyeksi yang mencakup yaitu:

- Command Prompt sebagai executor dan pemberi perintah agar Python dan Django dapat berkoordinasi untuk membuat website berbasis Django dan Python.
- Python for Windows Sebagai extension pada windows yang dieksekusi didalam cmd agar dapat menjalankan Instalasi Django.
- Django sebagai extension pada Python sebagai management website yang digunakan untuk Proyek Kali ini.
- Microsoft VisualStudio sebagai File manager berbasis GUI dan editor dari isi-isi file python dan .html yang digunakan sebagai backend pada website agar website dapat bekerja sesuai dengan yang diinginkan.
- Google Chrome Browser for Windows dimana browser ini berfungsi untuk menguji apakah .html dan perhitungan pada python berjalan atau tidak, dan muncul atau tidak pada website nya.

II PEMBAHASAN

2.1 Deskripsi Sistem

Aplikasi konversi suhu untuk mempermudah dalam mengetahui konversi satuan suhu. Setelah mengenal macam dan jenis skala suhu Celcius, Reamur, Kelvin dan Fahrenheit . Didalam konversi suhu itu sendiri merupakan cara untuk menyatakan suhu didalam suatu benda dari 1 skala suhu ke skala suhu lainnya.

Dalam pemilihan tema, praktikan melakukan pertimbangan yang mendorong praktikan untuk memilih tema proyek “APLIKASI KONVERSI SUHU” yaitu

Praktikan melihat salah satu kasus siswa kesulitan dalam memahami soal materi konversi suhu yang sebagian besar masih dibawah standar Kriteria Ketuntasan Minimal (KKM), Penyebab kemungkinan terjadi yaitu ketidakmampuan siswa dalam soal perhitungan konversi suhu, serta kurangnya metode dalam pengajaran yang mungkin membosankan dan juga belum dimanfaatkannya media yang tepat untuk memahami materi pembelajarannya. Maka dari itu, target dalam tujuan pembuatan aplikasi konversi suhu berbasis web ini adalah siswa atau murid sekolah dasar.

Sesuai penjelasan yang telah diuraikan diatas, hal tersebut yang membuat praktikan tertarik untuk melakukan eksplorasi pada *python* menggunakan *Framework Django*. Eksplorasi yang dilakukan akan diimplementasikan untuk membuat aplikasi konversi suhu dalam proyek akhir mata kuliah Pemrograman Berorientasi Objek (PBO).

2.2 Gambaran Arsitektur

Adapun langkah-langkah dalam pembuatan Aplikasi Konversi Suhu berbasis web dengan menggunakan *Django FrameWork*, diantara lain:

1. Jalankan Aplikasi Django nya pada cmd.

```
>. \PBO KEL3 \Scripts \activate
```

Gambar 1. Menjalankan Aplikasi Django

2. Jalankan base coding nya dengan menggunakan command *code*. Maka aplikasi VisualStudio Akan terbuka untuk meng-edit filenya.

```
s \proyekkel3utama>code
```

Gambar 2. Menjalankan Base Coding

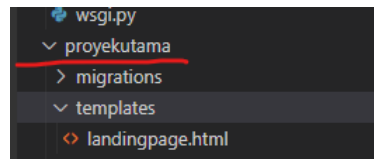
3. Jalankan servernya agar bisa dibuka.

```
py manage.py runserver
```

Gambar 3. Jalankan server

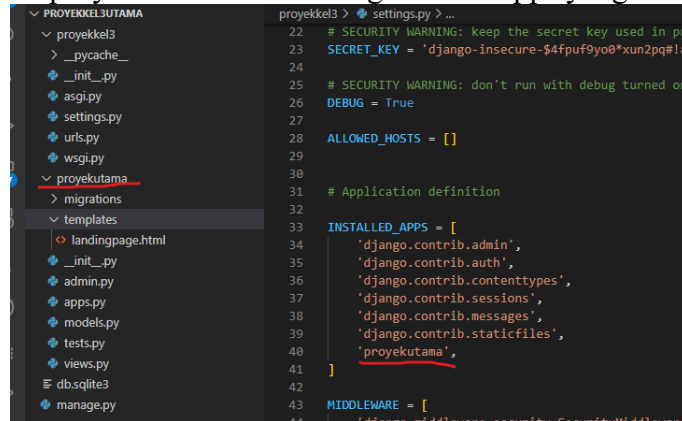
4. Jalankan proyekutama terlebih dahulu. Lalu lihat pada VSCODE apakah ada Utama atau tidak.

```
py manage.py startapp proyekutama
```



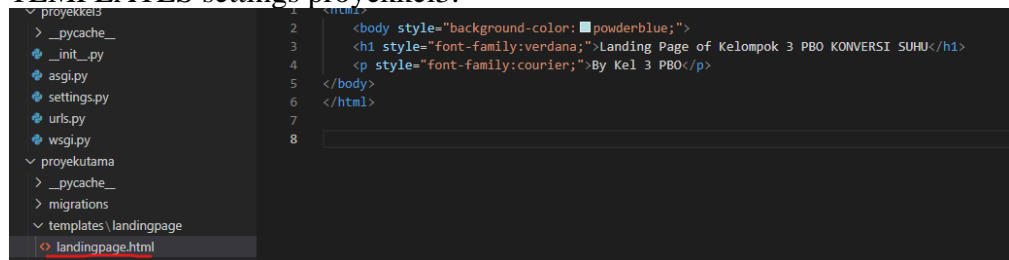
Gambar 4. Menjalankan proyek utama

5. Tambahkan 'proyekutama' sesuai dengan nama apps yang dibuat tadi.



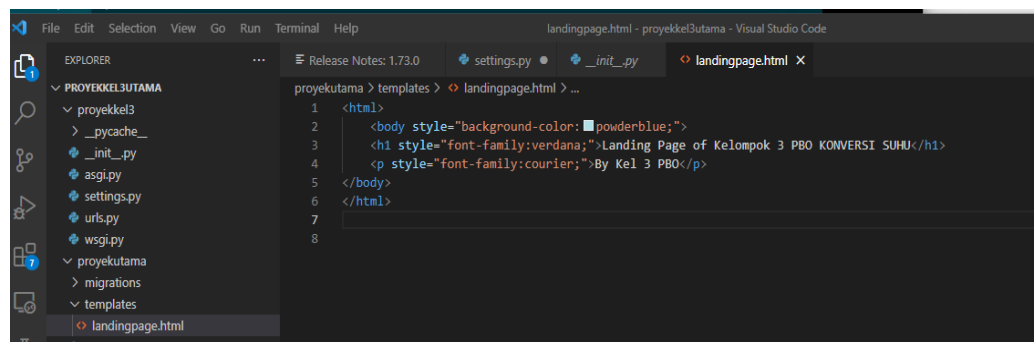
Gambar 5. Menambahkan proyek utama

6. Buatlah sebuah landingpage atau halaman utama. Buat folder pada folder *proyekutama* lalu buat folder *landingpage* didalam folder *templates* dimana folder tersebut akan dimasukkan ke bagian DIRS pada *TEMPLATES* settings proyekkel3.



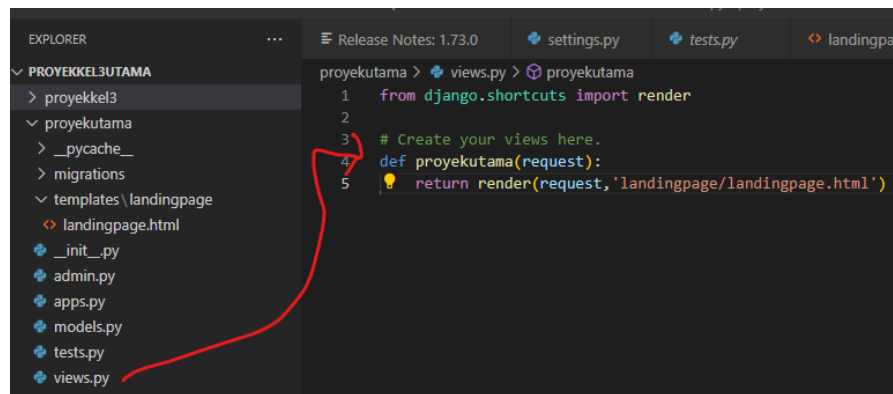
Gambar 6. Membuat sebuah landing page

7. Codingkan website nya. Codingkan landing page nya. Disini menggunakan codingan sederhana. Setelah itu save.



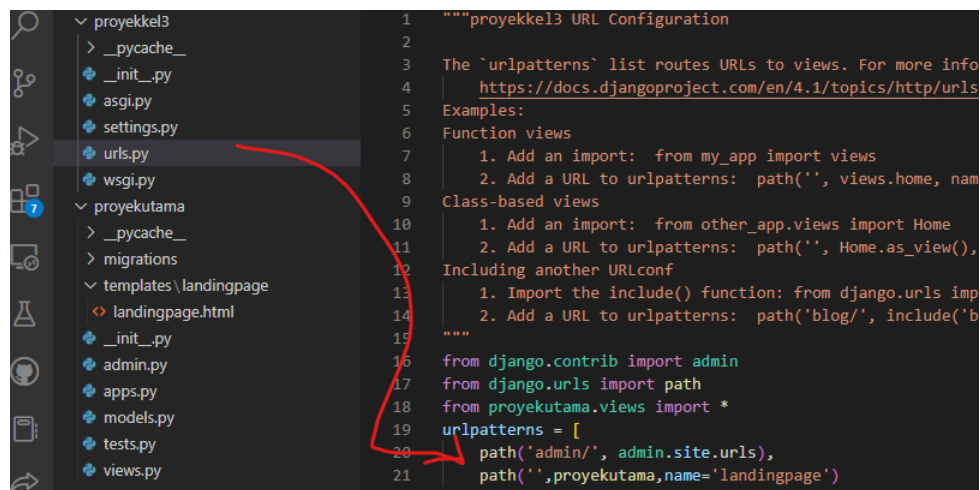
Gambar 7. Melakukan Codingan

8. Arahkan ke proyekutama. Buka views. Lalu berikan command `def` landingpage untuk menyalakan landing page tersebut



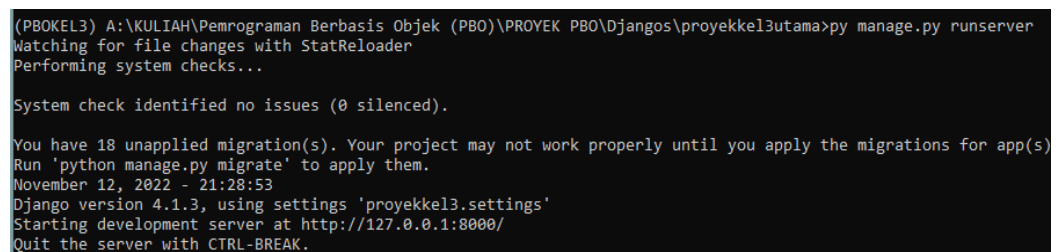
Gambar 8. Memberikan command `def` Landing Page

9. Pada proyekkel3 di urls.py maka tambahkan from untuk proyekutama lalu ke views dan import semuanya. Tambahkan urlpath landingpage. Seperti gambar di bawah. Setelah itu restart.



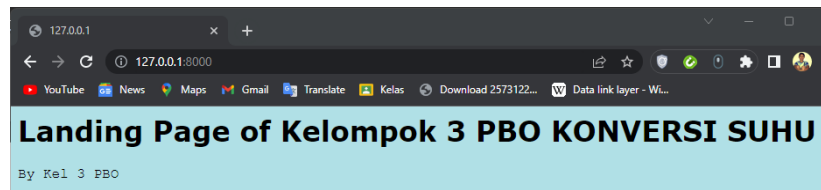
Gambar 9. Menambahkan form serta urlpath

10. Restart DJANGO nya. Lihat apakah ada error. Jika tidak maka seperti gambar di bawah. Buka lagi web nya



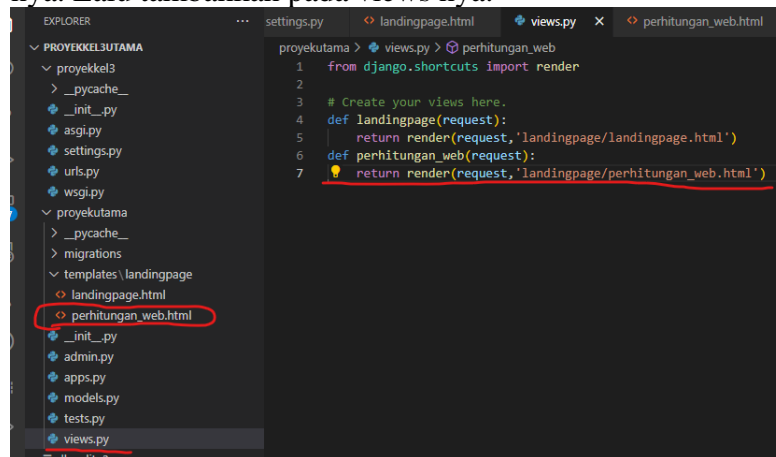
Gambar 10. Restart Django

11. Menguji apakah landing page nya yang dibuat tadi berjalan atau tidak



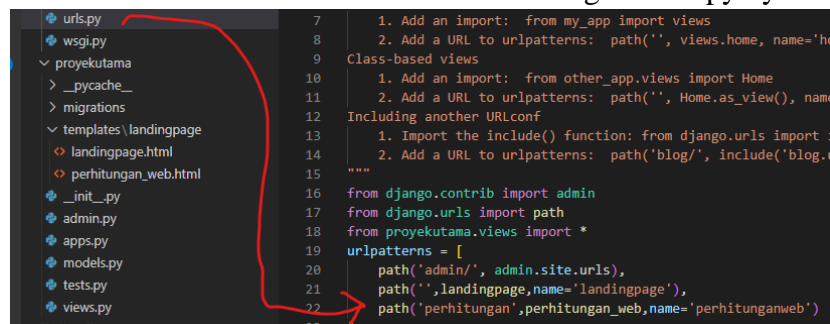
Gambar 11. Menguji Landing Page

12. Untuk membuat Subweb atau membuat web baru lagi. Maka buat lagi file nya. Lalu tambahkan pada views nya.



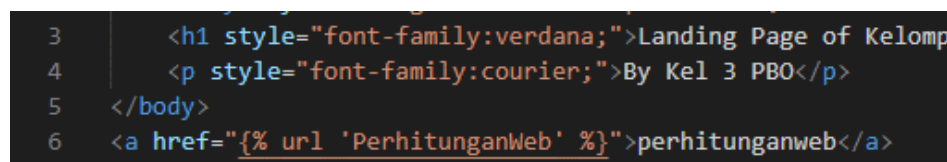
Gambar 12. Membuat Web baru lagi

13. Menambahkan web baru maka tambahkan lagi di urls.py nya.



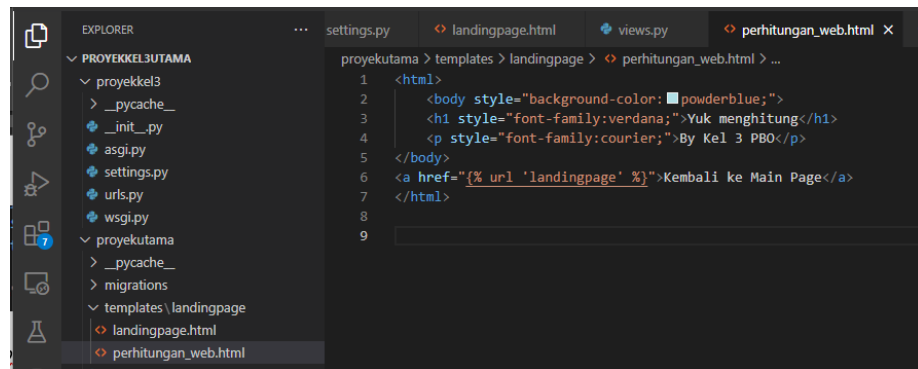
Gambar 13. Menambahkan web baru pada urls.py

14. Tambahkan href pada file html yang diinginkan untuk membuat sebuah tombol redirect lalu arahkan ke `perhitungan_web` atau def mana yang mau di load.



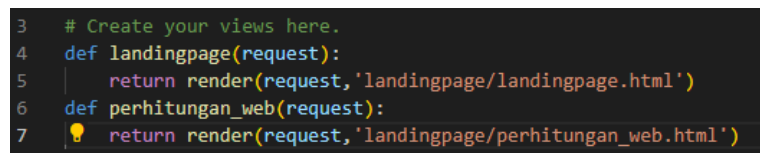
Gambar 14. Menambahkan href pada file html

15. Buat URL nya pada website *perhitungan_web*.



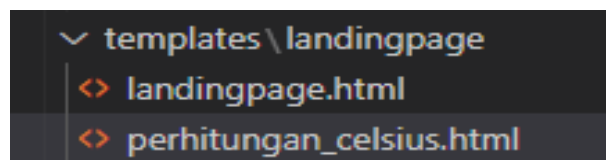
Gambar 15. Membuat URL pada website *perhitungan_web*

16. Tidak lupa pada *views.py* nya ditambahkan fungsi baru yaitu pada *perhitungan_web*.



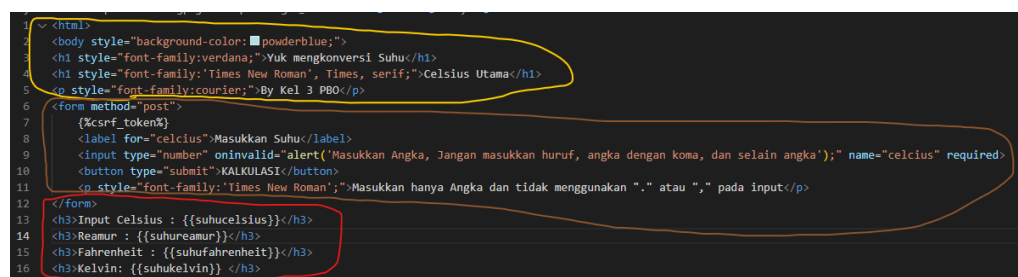
Gambar 16. Menambahkan fungsi baru pada *views.py*

17. Ubah nama html yang baru dibuat tadi. Ganti Namanya menjadi *perhitungan_celsius.html*.



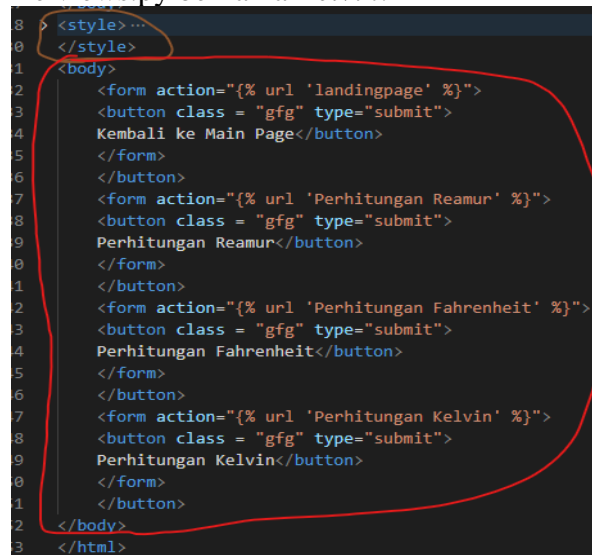
Gambar 17. Mengubah nama html

18. **Pada Garis kuning** merupakan header dari web Celsius. Lalu Garis coklat merupakan codingan untuk memberi Input yang akan ditangkap datanya oleh python. Setelah itu **pada garis merah** merupakan output dari semuanya setelah dikalkulasikan dalam python yaitu dengan 4 output pertama *Input Celsius* yang berisi `{{suhucelsius}}` dimana command tersebut akan memasukkan data yang disamakan formatnya dari perhitungan python.



Gambar 18. Header Web Celcius dan Output

19. Pada Garis coklat merupakan style untuk membuat Tombol. Lalu pada body yaitu membuat tombol. Disini tombol nya memiliki 4 tombol dimana tombol pertama yaitu mengarah ke landingpage, bisa dilihat dalam *form action* nya mengarah ke url *landingpage* yang akan mengarahkan ke views.py bernama *landingpage*, lalu selanjutnya dalam *form action* nya mengarah ke url *Perhitungan Reamur* yang akan mengarahkan ke views.py bernama *Perhitungan Reamur*, lalu selanjutnya dalam *form action* nya mengarah ke url *Perhitungan Fahrenheit* yang akan mengarahkan ke views.py bernama *Fahrenheit*, lalu selanjutnya dalam *form action* nya mengarah ke url *Perhitungan Kelvin* yang akan mengarahkan ke views.py bernama *Kelvin*.



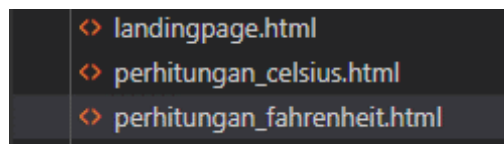
```

8 <style>...
9 </style>
10
11 <body>
12 <form action="{% url 'landingpage' %}">
13 <button class = "gfg" type="submit">
14 Kembali ke Main Page</button>
15 </form>
16 </button>
17 <form action="{% url 'Perhitungan Reamur' %}">
18 <button class = "gfg" type="submit">
19 Perhitungan Reamur</button>
20 </form>
21 </button>
22 <form action="{% url 'Perhitungan Fahrenheit' %}">
23 <button class = "gfg" type="submit">
24 Perhitungan Fahrenheit</button>
25 </form>
26 </button>
27 <form action="{% url 'Perhitungan Kelvin' %}">
28 <button class = "gfg" type="submit">
29 Perhitungan Kelvin</button>
30 </form>
31 </button>
32 </body>
33 </html>

```

Gambar 19. Coding Style Tombol

20. Buat html baru yang bernama *perhitungan_fahrenheit.html*. Duplikat isi filenya sama seperti *perhitungan_celsius.html* untuk mempermudah proses karena hanya ada beberapa perubahan saja.



```

<> landingpage.html
<> perhitungan_celsius.html
<> perhitungan_fahrenheit.html

```

Gambar 20. Membuat html (*Perhitungan_fahrenheit.html*)

21. Ubah pada garis merah yaitu nama web nya bernama "Fahrenheit Utama" lalu pada form ubah labelnya, nama warning input typenya menjadi *fahrenheit* lalu terakhir pada output berikan kata *Input Fahrenheit* yang sebelumnya *Input Celsius* dan berikan {{suhufahrenheit}} sebagai output dari py nya. Sisanya yaitu 3 suhu selain Fahrenheit.

```

<html>
<body style="background-color: powderblue;">
<h1 style="font-family: verdana;">Yuk mengkonversi Suhu</h1>
<h1 style="font-family: 'Times New Roman', Times, serif;">Fahrenheit Utama</h1>
<p style="font-family: courier;">By Kel 3 P80</p>
<form method="post">
  {{csrf_token%}}
  <label for="fahrenheit">Masukkan Suhu</label>
  <input type="number" oninvalid="alert('Masukkan Angka, Jangan masukkan huruf, angka dengan koma, dan selain angka');" name="fahrenheit" required>
  <button type="submit">KALKULASI</button>
  <p style="font-family: 'Times New Roman';">Masukkan hanya Angka dan tidak menggunakan "." atau "," pada input</p>
</form>
<h3>Input Fahrenheit : {{suhufahrenheit}}</h3>
<h3>Celsius : {{suhucelsius}}</h3>
<h3>Reamur : {{suhureamur}}</h3>
<h3>Kelvin: {{suhukelvin}} </h3>

```

Gambar 21. Mengubah nama web dan label pada form

- 22.4 tombolnya disini dimana tombol pertama yaitu mengarah ke landingpage, bisa dilihat dalam *form action* nya mengarah ke url *landingpage* yang akan mengarahkan ke views.py bernama *landingpage*, lalu selanjutnya dalam *form action* nya mengarah ke url *Perhitungan Celsius* yang akan mengarahkan ke views.py bernama *Perhitungan Celsius*, lalu selanjutnya dalam *form action* nya mengarah ke url *Perhitungan Reamur* yang akan mengarahkan ke views.py bernama *Reamur*, lalu selanjutnya dalam *form action* nya mengarah ke url *Perhitungan Kelvin* yang akan mengarahkan ke views.py bernama *Kelvin*.

```

</style>
<body>
  <form action="{% url 'landingpage' %}">
    <button class = "gfg" type="submit">
      Kembali ke Main Page</button>
    </form>
  </button>
  <form action="{% url 'Perhitungan Celsius' %}">
    <button class = "gfg" type="submit">
      Perhitungan Celsius</button>
    </form>
  </button>
  <form action="{% url 'Perhitungan Reamur' %}">
    <button class = "gfg" type="submit">
      Perhitungan Reamur</button>
    </form>
  </button>
  <form action="{% url 'Perhitungan Kelvin' %}">
    <button class = "gfg" type="submit">
      Perhitungan Kelvin</button>
    </form>
  </button>
</body>

```

Gambar 22. Tampilan codingan untuk 4 tombol

23. Buat html baru yang bernama *perhitungan_kelvin.html*. Duplikat isi filenya sama seperti *perhitungan_celsius.html* untuk mempermudah proses karena hanya ada beberapa perubahan saja

```

<> perhitungan_fahrenheit.html
<> perhitungan_kelvin.html

```

Gambar 23. Membuat html (*perhitungan_kelvin.html*)

24. Ubah pada garis merah yaitu nama web nya bernama “Kelvin Utama” lalu pada form ubah labelnya, nama warning input typenya menjadi *kelvin* lalu terakhir pada output berikan kata *Input Kelvin* yang sebelumnya *Input Celsius* dan berikan `{{suhukelvin}}` sebagai output dari py nya. Sisanya yaitu 3 suhu selain Kelvin.

```
<html>
<body style="background-color: powderblue;">
<h1 style="font-family: verdana;">Yuk mengkonversi Suhu</h1>
<h1 style="font-family: 'Times New Roman', Times, serif;">Kelvin Utama</h1>
<p style="font-family: courier;">By Kel 3 P80</p>
<form method="post">
  {{csrf_token}}
  <label for="kelvin">Masukkan Suhu</label>
  <input type="number" oninvalid="alert('Masukkan Angka, Jangan masukkan huruf, angka dengan koma, dan selain angka');" name="kelvin" required>
  <button type="submit">KALKULASI</button>
  <p style="font-family: 'Times New Roman';">Masukkan hanya Angka dan tidak menggunakan "." atau "," pada input</p>
</form>
<h3>Input Kelvin : {{suhukelvin}}</h3>
<h3>Celsius : {{suhucelsius}}</h3>
<h3>Fahrenheit : {{suhufahrenheit}}</h3>
<h3>Reamur : {{suhureamur}}</h3>
</body>
```

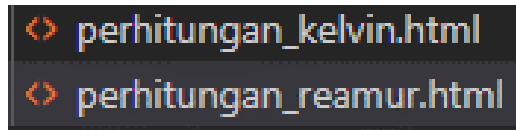
Gambar 24. Mengubah nama web dan form pada label

25. 4 tombolnya disini dimana tombol pertama yaitu mengarah ke landingpage, bisa dilihat dalam *form action* nya mengarah ke url *landingpage* yang akan mengarahkan ke views.py bernama *landingpage*, lalu selanjutnya dalam *form action* nya mengarah ke url *Perhitungan Celsius* yang akan mengarahkan ke views.py bernama *Perhitungan Celsius*, lalu selanjutnya dalam *form action* nya mengarah ke url *Perhitungan Reamur* yang akan mengarahkan ke views.py bernama *Reamur*, lalu selanjutnya dalam *form action* nya mengarah ke url *Perhitungan Fahrenheit* yang akan mengarahkan ke views.py bernama *Fahrenheit*.

```
<body>
  <form action="{% url 'landingpage' %}">
    <button class = "gfg" type="submit">
      Kembali ke Main Page</button>
    </form>
  </button>
  <form action="{% url 'Perhitungan Celsius' %}">
    <button class = "gfg" type="submit">
      Perhitungan Celsius</button>
    </form>
  </button>
  <form action="{% url 'Perhitungan Reamur' %}">
    <button class = "gfg" type="submit">
      Perhitungan Reamur</button>
    </form>
  </button>
  <form action="{% url 'Perhitungan Fahrenheit' %}">
    <button class = "gfg" type="submit">
      Perhitungan Fahrenheit</button>
    </form>
  </button>
```

Gambar 25. Codingan menampilkan 4 tombol

26. Buat html baru yang bernama *perhitungan_reamur.html*. Duplikat isi filenya sama seperti *perhitungan_celsius.html* untuk mempermudah proses karena hanya ada beberapa perubahan saja



Gambar 26. Mmbuat html (*perhitungan_reamur.html*)

27. Ubah pada garis merah yaitu nama web nya bernama “Reamur Utama” lalu pada form ubah labelnya, nama warning input typenya menjadi *kelvin* lalu terakhir pada output berikan kata *Input Kelvin* yang sebelumnya *Input Celsius* dan berikan `{{suhureamur}}` sebagai output dari py nya. Sisanya yaitu 3 suhu selain Reamur.

```
<html>
<body style="background-color: powderblue;">
<h1 style="font-family: verdana;">Yuk mengkonversi Suhu</h1>
<h1 style="font-family: 'Times New Roman', Times, serif;">Reamur Utama</h1>
<p style="font-family: courier;">By Kel 3 PBO</p>
<form method="post">
  {{csrf_token}}
  <label for="reamur">Masukkan Suhu</label>
  <input type="number" oninvalid="alert('Masukkan Angka, Jangan masukkan huruf, angka dengan koma, dan selain angka');" name="reamur" required>
  <button type="submit">KALKULASI</button>
  <p style="font-family: 'Times New Roman';">Masukkan hanya Angka dan tidak menggunakan "." atau "," pada input</p>
</form>
<h3>Input Reamur : {{suhureamur}}</h3>
<h3>Celsius : {{suhucelsius}}</h3>
<h3>Fahrenheit : {{suhufahrenheit}}</h3>
<h3>Kelvin: {{suhukelvin}}</h3>
```

Gambar 27. Mengubah nama web dan label pada form

28. 4 tombolnya disini dimana tombol pertama yaitu mengarah ke landingpage, bisa dilihat dalam *form action* nya mengarah ke url *landingpage* yang akan mengarahkan ke views.py bernama *landingpage*, lalu selanjutnya dalam *form action* nya mengarah ke url *Perhitungan Celsius* yang akan mengarahkan ke views.py bernama *Perhitungan Celsius*, lalu selanjutnya dalam *form action* nya mengarah ke url *Perhitungan Fahrenheit* yang akan mengarahkan ke views.py bernama *Fahrenheit*, lalu selanjutnya dalam *form action* nya mengarah ke url *Perhitungan Kelvin* yang akan mengarahkan ke views.py bernama *Kelvin*.

```

<body>
  <form action="{% url 'landingpage' %}">
    <button class = "gfg" type="submit">
      Kembali ke Main Page</button>
    </form>
  </button>
  <form action="{% url 'Perhitungan Celsius' %}">
    <button class = "gfg" type="submit">
      Perhitungan Celsius</button>
    </form>
  </button>
  <form action="{% url 'Perhitungan Fahrenheit' %}">
    <button class = "gfg" type="submit">
      Perhitungan Fahrenheit</button>
    </form>
  </button>
  <form action="{% url 'Perhitungan Kelvin' %}">
    <button class = "gfg" type="submit">
      Perhitungan Kelvin</button>
    </form>
  </button>

```

Gambar 28. Codingan 4 tombol

29. Pada `urls.py` yang ada di struktur *proyekkel3* maka tambahkan dahulu `from` nya yaitu dari struktur `directory` proyek utama dan untuk file `views.py` nya agar bisa diakses oleh `urls.py` pada struktur *proyekkel3*. Perbolehkan akses ke semua isinya dengan command `import *`.

```
from proyekutama.views import *
```

Gambar 29. pengubahan struktur `urlsl.py`

30. pada baris bawahnya. Buatlah `urlpatterns` selanjutnya. Garis kuning yaitu mengarah ke url dimana nanti pada browser akan ada prefix “(url garis kuning)” dimana nanti url tersebut akan menjalankan fungsi `def perhitungan_celsius` yaitu pada fungsi garis merah, lalu path ini dinamakan pada garis berwarna putih dimana nama nya dapat menjadi tombol untuk referensi dari file `.html` nya.

```

urlpatterns = [
    path('admin/', admin.site.urls),
    path(' ', landingpage, name='landingpage'),
    path('perhitungancelsius', perhitungan_celsius, name='Perhitungan Celsius'),
    path('perhitunganreamur', perhitungan_reamur, name='Perhitungan Reamur'),
    path('perhitunganfahrenheit', perhitungan_fahrenheit, name='Perhitungan Fahrenheit'),
    path('perhitungankelvin', perhitungan_kelvin, name='Perhitungan Kelvin')
]

```

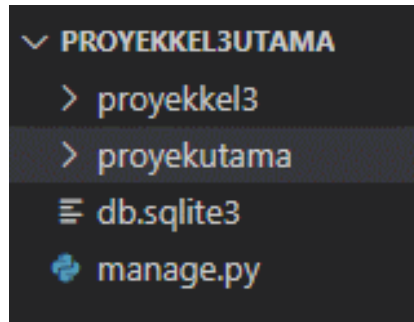
Gambar 30. Membuat `urlpatterns`

31. Untuk isi dari perhitungannya pada `views.py` dan struktur tombol maka terdapat pada bagian IV atau Listing Kode Program Aplikasi Konversi Suhu

2.3 Komponen Sistem

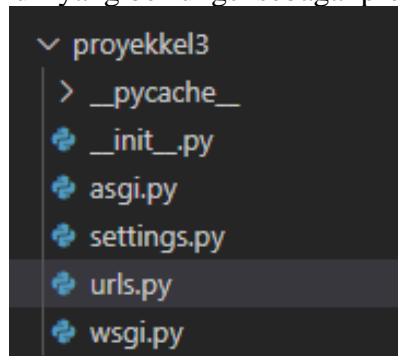
Adapun struktur direktori dalam pembuatan Aplikasi Konversi Suhu, yaitu berikut:

1. Pada main Directory ada *PROYEKKEL3UTAMA* dimana *PROYEKKEL3UTAMA* ini adalah sebuah supermain directory yang menaungi 2 child directory yaitu yang pertama bernama *proyekkel3* lalu *proyekutama*.



Gambar 31. Main Direcorty Proyekkel3utama

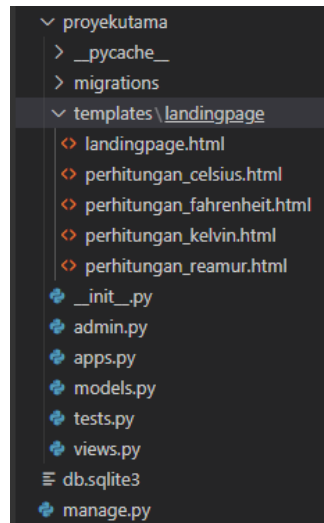
2. Pada subdirectory *proyekkel3* maka ada file-file python. Pada directory ini maka file .py yang diubah yaitu pada *settings.py* dimana pada file ini yaitu ditambahkan 1 apps baru bernama *proyekutama* lalu ada file *urls.py* yang berisi pattern-pattern url yang berfungsi sebagai prefix pada browser.



Gambar 32. Subdirecorty proyekkel3

3. Setelah menjalankan apps baru yaitu bernama *proyekutama* maka subdirectory *proyekutama* dibuat didalam naungan directory *PROYEKKEL3UTAMA*. Setelah itu

maka dibuat lagi Directory bernama *templates \ landingpage* yang berisi file html yang dapat diakses oleh *views.py*



Gambar 33. Direcorty templates/ landingpage

2.4 Kode Program

Dibawah ini berikut penjelasan setiap kode program yang dibuat dan alasan penggunaan metode pengkodean tersebut dalam pembuatan Aplikasi Konversi Suhu, diantara lain:

1. Lakukan import untuk fungsi render yang ada dari library function nya *django.shortcuts*. Setelah itu pada fungsi *landingpage*. Gunakan command *return render* dimana pada url *landingpage* ini hanya memunculkan isi website dari file *landingpage.html*. Maka gunakan command *render* untuk meload isi file html dengan parameter request menuju ke arah folder *landingpage* yaitu *landingpage/landingpage.html* pada subdirectory didalam *proyekutama*

```
1 from django.shortcuts import render
2
3 # Create your views here.
4 def landingpage(request):
5     return render(request, 'landingpage/landingpage.html')
```

Gambar 34. Melakukan import (untuk render)

2. Selanjutnya yaitu membuat fungsi bernama *perhitungan_celsius* dimana memiliki parameter bernama *request*. Lalu ada context berisi kosong yang digunakan untuk mengisi file html. Selanjutnya membuat kode perhitungan dimana jika metode request nya sama dengan 'POST' maka akan melanjutkan untuk perhitungan. Metode 'POST' ini ada didalam file *.html* perhitungannya. Lalu selanjutnya setelah ada input yang masuk menjadi variable *suhu* karena isi dari *variable* tersebut ialah merupakan *request* dari 'POST' nya. Jika angka telah dimasukkan pada website maka akan ditangkap dan dijadikan variable *suhu*. Terakhir pada *return render* dimana setelah dijalankan perhitungannya maka file *.html* pada url nya akan diupdate yang dimana isi context yang ada pada file *.html* telah diisi dengan angka-angka hasil perhitungan tadi.

```
def perhitungan_celsius(request):
    context = {}

    if request.method == 'POST' :
        suhu = int(request.POST['celsius'])
        celciustofahrenheit = (suhu * 9/5) + 32
        celciustoreamur = 0.8 * suhu
        celciustokelvin = suhu + 273.15
        context['suhucelsius'] = round(suhu, 2)
        context['suhufahrenheit'] = round(celciustofahrenheit, 2)
        context['suhureamur'] = round(celciustoreamur, 2)
        context['suhukelvin'] = round(celciustokelvin, 2)

    return render(request, 'landingpage/perhitungan_celsius.html', context)
```

Gambar 35. Kode program membuat fungsi (*perhitungan_celsius*)

3. Selanjutnya yaitu membuat fungsi bernama *perhitungan_reamur* dimana memiliki parameter bernama *request*. Lalu ada context berisi kosong yang digunakan untuk mengisi file html. Selanjutnya membuat kode perhitungan dimana jika metode request nya sama dengan 'POST' maka akan melanjutkan untuk perhitungan. Metode 'POST' ini ada didalam file *.html* perhitungannya. Lalu selanjutnya setelah ada input yang masuk menjadi variable *suhu* karena isi dari *variable* tersebut ialah merupakan *request* dari 'POST' nya. Jika angka telah dimasukkan pada website maka akan ditangkap dan dijadikan variable *suhu*. Selanjutnya diubah ke Celsius menggunakan variabel *kelselsiusdulu* dimana isinya adalah rumus konversi dari reamur ke Celsius adalah suhu dikali 5/4 nya. Lalu buat variabel bernama *reamur* yang berisi sama seperti variabel *kelselsiusdulu* lalu buat variabel lagi bernama *reamurtocelsius* yang berisi variabel reamur.
4. Selanjutnya pembuatan variabel *reamurtofahrenheit*, dan *reamurtokelvin*. Dimana pada variabel *reamurtofahrenheit* berisi rumus untuk menghitung dari suhu reamur yang telah dikonversi ke celsius nya dikali 9 per 5 dan ditambah 32. Untuk *reamurtokelvin* dimana *reamur* ditambahkan 273.15. terakhir maka menambahkan hasil tersebut ke dalam file *.html* nya pada context untuk 'suhureamur' yang ada pada *.html* nya maka masukkan variabel *suhu* yang telah dibulatkan menjadi hanya 2 angka dibelakang koma, lalu pada context untuk 'suhucelsius' yang ada pada *.html* nya maka masukkan variabel *reamurtocelsius* yang telah dibulatkan menjadi hanya 2 angka dibelakang koma, lalu pada context untuk 'suhufahrenheit' yang ada pada *.html* nya maka masukkan variabel *reamurtofahrenheit* yang telah dibulatkan menjadi hanya 2 angka dibelakang koma, dan terakhir pada context untuk 'suhukelvin' yang ada pada *.html* nya maka masukkan variabel *reamurtokelvin* yang telah dibulatkan menjadi hanya 2 angka dibelakang koma

```
def perhitungan_reamur(request):
    context = {

    }
    if request.method == 'POST' :
        suhu = int(request.POST['reamur'])
        kelselsiusdulu = suhu * 5/4
        reamur = kelselsiusdulu
        reamurtocelsius = reamur
        reamurtofahrenheit = (reamur * 9/5) + 32
        reamurtokelvin = reamur + 273.15
        context['suhureamur'] = round(suhu, 2)
        context['suhucelsius'] = round(reamurtocelsius, 2)
        context['suhufahrenheit'] = round(reamurtofahrenheit, 2)
        context['suhukelvin'] = round(reamurtokelvin, 2)

    return render(request,'landingpage/perhitungan_reamur.html',context)
```

Gambar 36. Kode membuat fungsi (perhitungan_reamur)

5. Selanjutnya yaitu membuat fungsi bernama *perhitungan_fahrenheit* dimana memiliki parameter bernama *request*. Lalu ada context berisi kosong yang digunakan untuk mengisi file html. Selanjutnya membuat kode perhitungan dimana jika metode request nya sama dengan 'POST' maka akan melanjutkan untuk perhitungan. Metode 'POST' ini ada didalam file *.html* perhitungannya. Lalu selanjutnya setelah ada input yang masuk menjadi variable *suhu* karena isi dari *variable* tersebut ialah merupakan *request* dari 'POST' nya. Jika angka telah dimasukkan pada website maka akan ditangkap dan dijadikan variable *suhu*. terakhir maka menambahkan hasil tersebut ke dalam file *.html* nya pada context untuk 'suhufahrenheit' yang ada pada *.html* nya maka masukkan variabel *suhu* yang telah dibulatkan menjadi hanya 2 angka dibelakang koma, lalu pada context untuk 'suhucelsius' yang ada pada *.html* nya maka masukkan variabel *fahrenheittocelsius* yang telah dibulatkan menjadi hanya 2 angka dibelakang koma, lalu pada context untuk 'suhureamur' yang ada pada *.html* nya maka masukkan variabel *fahrenheittoreamur* yang telah dibulatkan menjadi hanya 2 angka dibelakang koma, dan terakhir pada context untuk 'suhukelvin' yang ada pada *.html* nya maka masukkan variabel *fahrenheittokelvin* yang telah dibulatkan menjadi hanya 2 angka dibelakang koma.

```
def perhitungan_fahrenheit(request):
    context = {

    }
    if request.method == 'POST' :
        suhu = int(request.POST['fahrenheit'])
        kelselsiusdulu = (suhu - 32) * 5/9
        fahrenheit = kelselsiusdulu
        fahrenheittocelsius = fahrenheit
        fahrenheittoreamur = 0.8 * fahrenheit
        fahrenheittokelvin = fahrenheit + 273.15
        context['suhufahrenheit'] = round(suhu, 2)
        context['suhucelsius'] = round(fahrenheittocelsius, 2)
        context['suhureamur'] = round(fahrenheittoreamur, 2)
        context['suhukelvin'] = round(fahrenheittokelvin, 2)

    return render(request,'landingpage/perhitungan_fahrenheit.html',context)
```

Gambar 37. Kode membuat fungsi (perhitungan_fahrenheit)

6. Selanjutnya yaitu membuat fungsi bernama *perhitungan_kelvin* dimana memiliki parameter bernama *request*. Lalu ada context berisi kosong yang digunakan untuk mengisi file html. Selanjutnya membuat kode perhitungan dimana jika metode request nya sama dengan 'POST' maka

akan melanjutkan untuk perhitungan. Metode 'POST' ini ada didalam file *.html* perhitungannya. Lalu selanjutnya setelah ada input yang masuk menjadi variable *suhu* karena isi dari *variable* tersebut ialah merupakan *request* dari 'POST' nya. Jika angka telah dimasukkan pada website maka akan ditangkap dan dijadikan variable *suhu*. terakhir maka menambahkan hasil tersebut ke dalam file *.html* nya pada context untuk 'suhukelvin' yang ada pada *.html* nya maka masukkan variabel *suhu* yang telah dibulatkan menjadi hanya 2 angka dibelakang koma, lalu pada context untuk 'suhucelsius' yang ada pada *.html* nya maka masukkan variabel *kelvintocelsius* yang telah dibulatkan menjadi hanya 2 angka dibelakang koma, lalu pada context untuk 'suhufahrenheit' yang ada pada *.html* nya maka masukkan variabel *kelvintofahrenheit* yang telah dibulatkan menjadi hanya 2 angka dibelakang koma, dan terakhir pada context untuk 'suhureamur' yang ada pada *.html* nya maka masukkan variabel *kelvintoreamur* yang telah dibulatkan menjadi hanya 2 angka dibelakang koma.

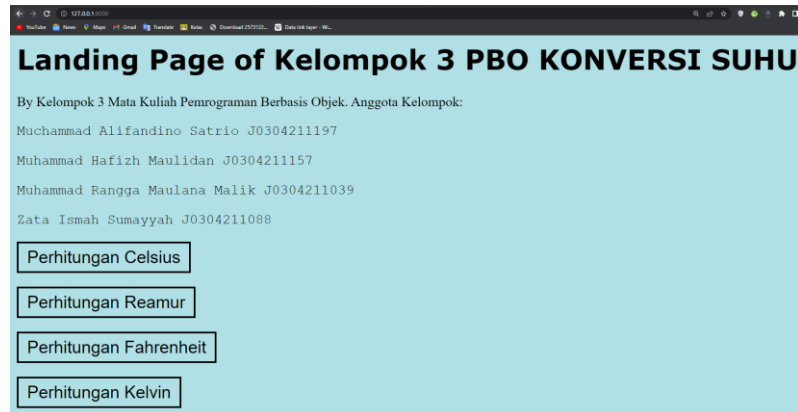
```
def perhitungan_kelvin(request):
    context = {

    }
    if request.method == 'POST':
        suhu = int(request.POST['kelvin'])
        kelselsiusdulu = suhu - 273.15
        kelvin = kelselsiusdulu
        kelvintocelsius = kelvin
        kelvintofahrenheit = (kelvin * 9/5) + 32
        kelvintoreamur = 0.8 * kelvin
        context['suhukelvin'] = round(suhu, 2)
        context['suhucelsius'] = round(kelvintocelsius, 2)
        context['suhufahrenheit'] = round(kelvintofahrenheit, 2)
        context['suhureamur'] = round(kelvintoreamur, 2)

    return render(request, 'landingpage/perhitungan_kelvin.html', context)
```

Gambar 38. Kode membuat fungsi (perhitungan_kelvin)

2.5 Demo Website



Gambar 39.. Tampilan Interface landing page

- Gambar diatas merupakan Landingpage pada url “. Dapat dilihat ada Penjelasan, Anggota kelompok dan Tombol. Pada 4 tombol tersebut yaitu tombol “Perhitungan Celsius”, “Perhitungan Reamur”, “Perhitungan Fahrenheit”, dan “Perhitungan Kelvin”. Maka jika tombol “Perhitungan Celsius” ditekan akan me-redirect ke url *perhitungancelsius* dan mengarah ke html Perhitungan Celsius.



Gambar 40.. Tampilan Interface landing page

- Selanjutnya pada Main Page Celsius. Sebelum dimasukkan Input maka penjelasan output masih kosong. Namun setelah dimasukkan input maka Input tersebut akan mengarah ke “Input Celsius”. Lalu konversi ke “Reamur”, “Fahrenheit”, dan “Kelvin” akan keluar outputnya berdasarkan perhitungan python pada *views.py*. Lalu ada tombol yang sama fungsinya yaitu yang pertama tombol untuk “Kembali ke Main Page”, lalu “Perhitungan Reamur”, lalu “Perhitungan Fahrenheit”, dan “Perhitungan Kelvin”.

Yuk mengkonversi Suhu

Reamur Utama

By Kel 3 PBO

Masukkan Suhu

Masukkan hanya Angka dan tidak menggunakan "." atau "," pada input

Input Reamur :

Celsius :

Fahrenheit :

Kelvin:

Input Reamur : 16787

Celsius : 20983.75

Fahrenheit : 37802.75

Kelvin: 21256.9

Gambar 41.. Tampilan Interface landing page

- Selanjutnya pada Main Page Reamur. Sebelum dimasukkan Input maka penjelasan output masih kosong. Namun setelah dimasukkan input maka Input tersebut akan mengarah ke “Input Reamur”. Lalu konversi ke “Celsius”, “Fahrenheit”, dan “Kelvin” akan keluar outputnya berdasarkan perhitungan python pada *views.py*. Lalu ada tombol yang sama fungsinya yaitu yang pertama tombol untuk “Kembali ke Main Page”, lalu “Perhitungan Celsius”, lalu “Perhitungan Fahrenheit”, dan “Perhitungan Kelvin”.

Yuk mengkonversi Suhu

Fahrenheit Utama

By Kel 3 PBO

Masukkan Suhu

Masukkan hanya Angka dan tidak menggunakan "." atau "," pada input

Input Fahrenheit :

Celsius :

Reamur :

Kelvin:

Input Fahrenheit : 855

Celsius : 457.22

Reamur : 365.78

Kelvin: 730.37

Gambar 42.. Tampilan Interface landing page

- Selanjutnya pada Main Page Fahrenheit. Sebelum dimasukkan Input maka penjelasan output masih kosong. Namun setelah dimasukkan input maka Input tersebut akan mengarah ke “Input Fahrenheit”. Lalu konversi ke “Celsius”, “Reamur”, dan “Kelvin” akan keluar outputnya berdasarkan perhitungan python pada *views.py*. Lalu ada tombol yang sama fungsinya yaitu yang pertama tombol untuk “Kembali ke Main Page”, lalu

“Perhitungan Celsius”, lalu “Perhitungan Reamur”, dan “Perhitungan Kelvin”.



Gambar 43.. Tampilan Interface landing page

- Selanjutnya pada Main Page Kelvin. Sebelum dimasukkan Input maka penjelasan output masih kosong. Namun setelah dimasukkan input maka Input tersebut akan mengarah ke “Input Kelvin”. Lalu konversi ke “Celsius”, “Fahrenheit”, dan “Reamur” akan keluar outputnya berdasarkan perhitungan python pada *views.py*. Lalu ada tombol yang sama fungsinya yaitu yang pertama tombol untuk “Kembali ke Main Page”, lalu “Perhitungan Celsius”, lalu “Perhitungan Reamur”, dan “Perhitungan Fahrenheit”.

III KESIMPULAN

3.1 Simpulan

sehabis dilakukan banyak sekali macam fungsi serta perhitungan. Maka sudah jadi sebuah website menggunakan backend yang terstruktur dimulai berasal menginstall apps baru pada file settings.py, edit urls.py menjadi loader website serta penentu prefix pada browser, lalu urls.py ini menghubungkan file di views.py yang terdapat di subdirectory, selanjutnya views.py ini menjalankan file .html di directory landingpage. Hal ini berjalan secara terstruktur yang dimana website akan me-redirect sesuai menggunakan tombol yang ditekan oleh user website. Semuanya dapat terkoneksi mulai berasal Input dari html yang ditujukan ke perhitungan suhu yang terdapat pada views.py hingga outputnya diberikan dari views.py ke website melalui fungsi yang ada didalam kode-kode nya. Hal ini membagikan bahwa aplikasi konversi suhu ini sudah berjalan dengan baik sesuai dengan konsep awal yang diinginkan oleh penulis. Hal ini berarti bahwa penulis telah mampu dalam mengerjakan hal-hal yang berkaitan dengan object, function, trigger, serta lain sebagainya sampai database dimana hal ini adalah materi yang ada pada mata kuliah Pemrograman Berbasis Objek. Jika praktikan dapat membangun suatu proyek website yang berbasis object. Maka tujuan asal pembelajaran studi pada mata kuliah Pemrograman Berbasis Objek telah terselesaikan.

3.2 Saran

Pada bagian saran maka penulis disini mengharapkan beberapa koreksi pada website dimana pengetahuan penulis masih dianggap kecil. Beberapa saran yang diharapkan yaitu bagaimana caranya agar input pada .html dapat memberikan input angka dengan koma, lalu errorpage jika input yang diberikan kosong atau tidak sesuai dengan input yang diinginkan seperti input huruf yang tidak diinginkan dapat me-redirect ke web errorpage yang dimana hal ini dapat membuat proyek yang dibuat oleh penulis lebih baik dan dapat dipasarkan ke publik dengan baik.

DAFTAR PUSTAKA

Yusro, M. (2017). BINTANG SKALA MENINGKATKAN AKTIVITAS DAN KEMAMPUAN MENYELESAIKAN SOAL KONVERSI SUHU. *Journal Profesi Keguruan*. <https://journal.unnes.ac.id/nju/index.php/jpk/article/view/10482/7033>