

Hasil Test

user: Muhammad Athallah 2006527481 - muhammad.athallah01
 test: **[UPDATE] Latihan Pra-UAS PBP S1 Reguler 2021 - 1**
 Platform Based Programming
 waktu mulai: 2021-12-11 22:56:04
 waktu selesai: 2021-12-11 23:14:38
 waktu test: 00:18:34
 poin: 35.000 / 48.000 (73%)
 benar: 17 / 26 (65%)
 salah: 9 / 26 (35%)
 tidak dijawab: 0 / 26 (0%)
 tidak tampil: 0 / 26 (0%)
 tanpa rating: 0 / 26 (0%)
 komentar:

1. **[1.000] (IP:281473568538625 | 22:56:04 | 22:56:22 | 00:18 | 17.561)**

Berikut ini adalah yang **tidak** termasuk server side scripting languages adalah:

- ☒ ☐ HTML
- ☐ ☐ PHP
- ☐ ☐ Phyton
- ☐ ☐ Ruby
- ☐ ☐ Java

2. **[3.000] (IP:281473568538625 | 22:56:22 | 22:56:41 | 00:19 | 18.008)**

Perhatikan model Makanan yang merepresentasikan makanan yang dijual di kantin berikut ini:

- class Makanan(models.Model):
- nama = models.CharField(max_length=50, 3. blank=False)
- harga = models.IntegerField()
- created_date = models.DateTimeField(auto_now_add=True)

Berikut ini yang dimaksud dengan kode baris ke-5 dari potongan kode di atas adalah:

- ☐ ☐ Nilai dari nama, harga, dan created_date akan diisi secara otomatis saat pertama kali database dibuat
- ☐ ☐ Object Makanan akan dibuat secara otomatis
- ☐ ☐ Nilai dari attribute created_date akan diisi secara otomatis dengan tanggal dan waktu setiap kali ada perubahan data didalam database
- ☒ ☐ Nilai dari attribute created_date akan diisi secara otomatis dengan tanggal dan waktu saat data pertama kali dibuat
- ☐ ☐ Nilai dari attribute created_date harus diisi dengan tanggal dan waktu yang sesuai dengan timezone yang digunakan saat ini

3. **[1.000] (IP:281473568538625 | 22:56:41 | 22:56:54 | 00:13 | 11.676)**

Berikut ini salah satu aktivitas dari penerapan TDD yang benar.

- ☐ ☐ memperbaiki layout penulisan kode
- ☐ ☐ menerapkan continuous integration dan continuos deployment.
- ☐ ☐ Meminta programmer lain membuat code-coverage dari project kita 100%
- ☐ ☐ membuat exception handler untuk setiap kemungkinan error
- ☒ ☐ Unit testing dilakukan sebelum mengimplementasikan fitur

4. **[0.000] (IP:281473568538625 | 22:56:54 | 22:58:41 | 01:47 | 106.066)**

Pada saat mengerjakan story, halaman web yang sudah dibuat dan di-*deploy* ke Heroku ternyata tidak sesuai dengan *styling* yang sudah diatur di file css. Hal ini terjadi karena...

- ☐ ☐ Terdapat folder 'static' di folder root project
- ☐ ☐ Tidak ada fungsi style di views.py
- ☐ ☐ Browser yang digunakan tidak support dengan versi CSS yang digunakan
- ☐ ☐ Belum menambahkan whitenoise di settings.py
- ☒ ☐ Salah melakukan setting path css eksternal

5. **[0.000] (IP:281473568538625 | 22:58:41 | 23:00:32 | 01:51 | 108.652)**

Pernyataan berikut yang **SALAH** terkait dengan HTML Form dan beberapa komponen Django:

- ☐ ☐ Atribut pada Class FORMS.py harus sama dengan atribut pada MODELS.py
- ☐ ☐ Pada file HTML, tampilan FORM dapat digenerate berdasarkan informasi dari VIEWS
- ☒ ☐ Penyimpanan data FORM ke MODELS, dilakukan dengan membuat fungsi pada VIEWS.
- ☐ ☐ Action pada FORM mengacu pada URLS yang telah didefinisikan
- ☐ ☐ Salah satu fungsi pada VIEWS bertugas membuat objek untuk kelas FORMS

6. **[0.000] (IP:281473568538625 | 23:00:32 | 23:02:55 | 02:23 | 141.778)**

Terdapat potongan kode seperti berikut :

```
<head>
<style>
  body {
    color: pink;
  }
  div {
```

```
        color: black;
    }
    h1 {
        color: white;
        text-align: center;
    }
    .row .what-color {
        color: blue;
    }
    .what-color {
        color: green;
    }
</style>
</head>
<body>
<div class="row">
<h1 class="what-color" style="color:inherit;"> Warna PPW bagi saya ..... </h1>
</div>
```

Apakah warna dari tulisan "Warna PPW bagi saya"

1.

☒

☐

hijau
2.

☐

☐

putih
3.

☐

☒

hitam
4.

☐

☐

merah muda
5.

☐

☐

biru

7. [1.000] (IP:281473568538625 | 23:02:55 | 23:03:17 | 00:22 | 20.01)

Flutter memiliki mekanisme stack dalam navigatornya. Jika kita menekan tombol back, artinya terjadi pemanggilan ..

1.

☐

☐

Navigator.pages
2.

☒

☒

Navigator.pop
3.

☐

☐

Navigator.onPressed
4.

☐

☐

Navigator.push

8. [0.000] (IP:281473568538625 | 23:03:17 | 23:03:27 | 00:10 | 9.627)

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('First Route'),
    ),
    body: Center(
      child: ElevatedButton(
        child: const Text('Open route'),
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => const SecondRoute()),
          );
        },
      ),
    ),
  );
}
```

Pada cuplikan kode di atas ini, atribut / class / fungsi apa yang menyebabkan terjadinya perpindahan layout ke halaman lain?

1.

☐

☒

Navigator.push
2.

☒

☐

MaterialPageRoute
3.

☐

☐

return Scaffold
4.

☐

☐

Text('First Route')

9. [3.000] (IP:281473568538625 | 23:03:27 | 23:03:47 | 00:20 | 19.058)

Berikut ini adalah manfaat menerapkan external CSS, pernyataan yang paling tepat adalah...

1.

☐

☐

Agar semua halaman web bisa seragam.
2.

☐

☐

Agar halaman web selalu responsif
3.

☒

☒

Agar pembaharuan layout/styling halaman web di kemudian hari menjadi lebih mudah
4.

☐

☐

Agar pengaturan halaman web jadi lebih bagus
5.

☐

☐

Agar halaman web bisa diatur ukuran huruf-nya

10. [0.000] (IP:281473568538625 | 23:03:47 | 23:03:53 | 00:06 | 6.029)

Perhatikan potongan kode views.py berikut:

```
1 def index(request):
2     ...
3     to_do = {'name': 'Budi'}
4     return render(request, 'index.html', to_do)
```

| | | | |
|----|---|---|------------|
| 1. | | | %name% |
| 2. | | | { name } |
| 3. | | ® | {{ name }} |
| 4. | | | name |
| 5. | x | | {% name %} |

```
Future<void> fetchData() async {
  const url = 'http://127.0.0.1:8000/view_anggota';
  try {
    final response = await http.get(Uri.parse(url));
    print(response.body);
    Map<String, dynamic> extractedData = jsonDecode(response.body);
    extractedData.forEach((key, val) {
      print(val);
    });
  } catch (error) {
    print(error);
  }
}
```

akan di-return hasil pemanggilan webservice ke pemanggil fungsi fetchData

| | | | |
|----|---|---|------------|
| 1. | | ® | {{ name }} |
| 2. | x | | {% name %} |
| 3. | | | %name% |
| 4. | | | name |
| 5. | | | { name } |

| | | | |
|----|-------------------------------------|-------------------------------------|---|
| 1. | <input type="checkbox"/> | <input checked="" type="checkbox"/> | Data yang dimasukkan dapat disimpan kedalam <i>bookmark</i> |
| 2. | <input type="checkbox"/> | <input type="checkbox"/> | Data form dapat dirahasiakan |
| 3. | <input type="checkbox"/> | <input type="checkbox"/> | Data yang dimasukkan tidak akan terlihat di alamat URL |
| 4. | <input type="checkbox"/> | <input type="checkbox"/> | Data form dengan method GET tidak akan bisa dimasukkan ke <i>database</i> |
| 5. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Data yang dimasukkan akan ditampilkan pada <i>webpage</i> |

```
<p> Saya </p>
<div>
<span><p> Kamu </p></span>
<p> Kita </p>
</div>
<div> Fasilkom </div>
```

Jika ditambahkan kode berikut pada bagian css,

p {
color: blue;
}

Tulisan mana yang akan berwarna biru?

1. ☐ ☐ Fasikom
2. ☐ ☐ Saya
3. ☐ ☐ Kamu
4. ☒ ☒ Kamu, Kita
5. ☐ ☐ Saya, Kamu, Fasikom

15. [1.000] (IP:281473568538625 | 23:06:38 | 23:07:10 | 00:32 | 31.576)

Apa yang harus kita lakukan agar Unit Test yang sudah dibuat dapat lolos?

1. ☐ ☐ Tambah test baru pada file tests.py yang dapat mengatasi masalah tersebut
2. ☐ ☐ Hapus fungsi pada views.py yang tidak dijalankan saat test
3. ☐ ☐ Hapus baris pada file tests.py
4. ☐ ☐ Perhatikan method pada file tests.py, perbaiki method yang membuat test coverage berkurang
5. ☒ ☒ Perbaiki fungsi pada views.py agar sesuai dengan ekspektasi saat Unit Test dilakukan

16. [1.000] (IP:281473568538625 | 23:07:10 | 23:07:20 | 00:10 | 8.117)

Berikut adalah pernyataan yang benar mengenai Flutter, kecuali ..

1. ☐ ☐ Dapat dijalankan dengan perintah flutter run
2. ☐ ☐ Struktur UI-nya memiliki representasi logis dalam bentuk Widget Tree
3. ☐ ☐ Perintah flutter doctor digunakan dalam pengecekan konfigurasi
4. ☒ ☒ Flutter mengenali text direction langsung secara implisit

17. [1.000] (IP:281473568538625 | 23:07:20 | 23:08:03 | 00:43 | 42.59)

Berikut adalah hal - hal yang biasanya dilakukan di berkas views.py, **kecuali**:

1. ☐ ☐ Menambahkan data kedalam database
2. ☐ ☐ Membuat fungsi penambahan dua bilangan
3. ☐ ☐ Menentukan berkas template yang akan ditampilkan
4. ☒ ☒ Membuat konfigurasi URL
5. ☐ ☐ Merender templates

18. [3.000] (IP:281473568538625 | 23:08:03 | 23:08:16 | 00:13 | 11.769)

Manakah salah satu langkah yang relevan untuk membuat sebuah model pada Django Framework:

1. ☐ ☐ Menjalankan perintah python manage.py test
2. ☐ ☐ Membuat berkas deployment.sh
3. ☐ ☐ Menjalankan perintah python manage.py showmigration
4. ☐ ☐ Menjalankan perintah python manage.py migration
5. ☒ ☒ Menjalankan perintah python manage.py migrate

19. [5.000] (IP:281473568538625 | 23:08:16 | 23:09:26 | 01:10 | 69.365)

Perhatikan potongan kode di bawah ini untuk menambahkan data pada model Makanan:

```
>>> q1 = Makanan.objects.create(nama='Ayam Cah Cabe Kering', harga=16000)
>>> q1 = Makanan.objects.create(nama='Sate Ayam', harga=16000)
>>> q2 = Makanan.objects.create(nama='Nasi Rawon', harga=15000)
>>> q3 = Makanan.objects.create(nama='Gado-gado', harga=11000)
```

Jika ingin **hanya** menghapus makanan 'Ayam Cah Cabe Kering' dari database maka menggunakan perintah:

1. ☐ ☐ Makanan.objects.delete(nama='Ayam Cah Cabe Kering')
2. ☐ ☐ Makanan.objects.get(harga=16000).delete()
3. ☐ ☐ Makanan.objects.all().delete()
4. ☐ ☐ Makanan.objects.obtain(nama='Ayam Cah Cabe Kering').delete()
5. ☒ ☒ Makanan.objects.get(nama='Ayam Cah Cabe Kering').delete()

20. [0.000] (IP:281473568538625 | 23:09:26 | 23:10:01 | 00:35 | 34.271)

Pernyataan yang benar terkait dengan potongan kode berikut ini adalah:

1. <form id="form" method="POST" action="{% url 'add_message' %}">
2. {% csrf_token %}
3. {{ message_form }}
4.

5. <button type="submit">Kirim</button>
6. </form>
1. ☐ ☐ Method POST tidak dapat diganti dengan method GET
2. ☐ ☒ message_form merupakan atribut yang berasal dari VIEWS.py
3. ☐ ☐ CSRF token menghasilkan atribut baru pada FORM yang dapat dilihat oleh pengguna pada halaman web
4. ☒ ☐ add_message harus dibuat sebagai nama fungsi pada views
5. ☐ ☐ Nama button KIRIM pada baris ke 5 tidak dapat diganti dengan nama lain.

21. [1.000] (IP:281473568538625 | 23:10:01 | 23:10:25 | 00:24 | 22.537)

Berikut ini adalah pernyataan yang SALAH atas pemanggilan "python manage.py startapp aplikasi_baru":

- ☐ ☐ Aplikasi web dengan nama "aplikasi_baru" belum dapat diakses karena belum dilakukan konfigurasi pada urls.py
- ☒ ☒ Perintah "python manage.py startapp aplikasi_baru" adalah untuk menjalankan Aplikasi webserver Django
- ☐ ☐ Di dalam direktori aplikasi_baru terdapat berkas views.py, models.py, dan tests.py
- ☐ ☐ Sebuah aplikasi web bernama "aplikasi_baru" secara otomatis dibuat, namun belum secara otomatis terdaftar di dalam Framework Django
- ☐ ☐ Berkas-berkas views.py, models.py, dan tests.py masih kosong dan perlu dilakukan implementasi

22. **[5.000] (IP:281473568538625 | 23:10:25 | 23:12:02 | 01:37 | 96.602)**

Perhatikan dua berkas berikut:

tarungcss.css:

```
.tarung {  
  color: pink !important  
  background: red  
}
```

tarunghtml.html:

```
<html>  
<head>  
<link href="tarungcss.css" rel="stylesheet" />  
<style>  
  .tarung {  
    color: red  
    background : blue  
  }  
</style>  
</head>  
<body>  
<p class="tarung" style="color: blue">PPW adalah bekal awal saya untuk membuat sesuatu yang bermanfaat</p>  
<p>Saya cinta PPW</p>  
</body>  
</html>
```

Bagaimanakah tampilan pada tag

<p>

di atas?

- ☐ ☐ background berwarna merah muda
- ☒ ☒ kalimat "PPW adalah bekal awal saya untuk membuat sesuatu yang bermanfaat" berwarna biru
- ☐ ☐ kalimat "PPW adalah bekal awal saya untuk membuat sesuatu yang bermanfaat" berwarna merah muda dan background berwarna biru
- ☐ ☐ semua tulisan berwarna biru
- ☐ ☐ kalimat "PPW adalah bekal awal saya untuk membuat sesuatu yang bermanfaat" berwarna merah dan background berwarna merah muda

23. **[0.000] (IP:281473568538625 | 23:12:02 | 23:12:29 | 00:27 | 25.945)**

Ada permintaan bahwa ketika mengakses www.hostname.com/login akan mengembalikan halaman login. Berikut ini adalah deskripsi unit test yang relevan untuk permintaan tersebut adalah:

- ☐ ☐ Ketika URL diakses, maka harus ada tulisan "Login"
- ☐ ☐ Ketika URL diakses, maka akan menggunakan fungsi login index
- ☐ ☒ Ketika URL diakses, maka berkas login.css juga harus diberikan
- ☒ ☐ Ketika URL diakses, maka harus memberikan success response code 200
- ☐ ☐ Ketika URL diakses, maka header.html dan footer.html harus ditampilkan

24. **[1.000] (IP:281473568538625 | 23:12:29 | 23:12:39 | 00:10 | 9.633)**

Apa bedanya antara Stateful and Stateless widgets di Flutter?

- ☒ ☒ StatefulWidget memiliki informasi status dan bersifat dinamis karena dapat mengubah data selama masa pakai widget.
- ☐ ☐ StatefulWidget memiliki build() method.
- ☐ ☐ StatelessWidget memiliki createState() method
- ☐ ☐ Widget yang memungkinkan kita untuk me-refresh layar dapat dilakukan pada StatelessWidget .

25. **[3.000] (IP:281473568538625 | 23:12:39 | 23:13:15 | 00:36 | 35.586)**

Pada beberapa pernyataan mengenai *Client Side Scripting* vs *Server Side Scripting* di bawah ini yang **sesuai** adalah:

- ☐ ☐ Server side scripting bergantung pada merk & kemampuan Browser
- ☐ ☐ Client side scripting tidak bergantung pada merk & kemampuan Browser
- ☐ ☐ User dapat melihat source code berupa Python dari Server Side scripting dengan cara mengarahkan browser pada alamat website
- ☐ ☐ Kita dapat menggabungkan dua bahasa server side scripting untuk mengembangkan satu web aplikasi
- ☒ ☒ User dapat melihat source code berupa Java Script dari Client Side scripting dengan cara View Source pada browser

26. **[3.000] (IP:281473568538625 | 23:13:15 | 23:14:38 | 01:23 | 80.274)**

Ketika seseorang membuat "unit test", sebenarnya apakah yang dibuatnya?

- ☐ ☐ menghasilkan sebuah unit kerja (bagian) dalam sebuah software house yang menguji hasil kerja programmer.
- ☐ ☐ sebuah project django yang melakukan testing terhadap project django lainnya.
- ☐ ☐ menghasilkan sebuah pipeline di gitlab bisa berjalan.
- ☐ ☐ program untuk memeriksa keseragaman kerja sesama programmer
- ☒ ☒ fungsi untuk memeriksa kesesuaian program yang dibuat

statistik

statistik [semua + modul]

| M# | S# | Q# | A# | saat ini | poin | waktu | benar | salah | tidak dijawab | tidak tampil | tanpa rating |
|-------|----|----|----|---------------|------|-------|-----------|----------|---------------|--------------|--------------|
| semua | | | | 26 (100%) | 65% | 00:42 | 17 (65%) | 9 (35%) | 0 (0%) | 0 (0%) | 0 (0%) |
| M1 | | | | 21 (81%) | 62% | 00:48 | 13 (62%) | 8 (38%) | 0 (0%) | 0 (0%) | 0 (0%) |
| | | | | PPW - Django | | | | | | | |
| M2 | | | | 5 (19%) | 80% | 00:18 | 4 (80%) | 1 (20%) | 0 (0%) | 0 (0%) | 0 (0%) |
| | | | | PBP - Flutter | | | | | | | |

PDF

< Beranda

Halaman ini menampilkan detail test untuk user yang dipilih.

user: muhammad.athallah01 [logout](#)

[AR](#) | [AZ](#) | [BG](#) | [BR](#) | [CN](#) | [DE](#) | [EL](#) | [EN](#) | [ES](#) | [FA](#) | [FR](#) | [HE](#) | [HI](#) | [HU](#) | [ID](#) | [IT](#) | [JP](#) | [MR](#) | [MS](#) | [NL](#) | [PL](#) | [RO](#) | [RU](#) | [TR](#) | [UR](#) | [VN](#)

TCExam ver. 14.8.1 - Copyright © 2004-2020 Nicola Asuni - [Tecnick.com LTD](#)

W3C XHTML 1.0

W3C CSS 2.0

W3C WAI-AAA