

Nama aplikasi

Orplant (Ornament Plant)

Nama kelompok

Kelompok 8

Anggota

1. Ananda Thalia (G74170048)
2. Irsyad M. Andre (G74170077)
3. Ibrahim Dano P. (G74170080)

Latar belakang

Masyarakat masih merasa kesulitan dalam mencari informasi mengenai tanaman hias seperti cara merawat tanaman secara tepat dan mencari lokasi untuk membelinya. Komunitas tanaman hias juga belum memiliki wadah khusus untuk melakukan transaksi, selama ini mereka hanya bertukar informasi melalui media sosial.

Era globalisasi memudahkan manusia untuk mengakses informasi melalui internet. Berbagai macam informasi dapat diperoleh, salah satunya mengenai tanaman hias. Tanaman hias memiliki daya tarik tersendiri bagi siapa saja yang melihatnya. Tanaman hias menjadi salah satu hobi yang diminati masyarakat Indonesia, sehingga terbentuknya komunitas tanaman hias di Indonesia semakin banyak. Platform ini akan memudahkan bagi orang-orang untuk melakukan transaksi jual beli tanaman hias yang diinginkan tanpa khawatir untuk mencari lokasi dan harga tanaman hias yang akan dibeli.

Tujuan

Website ini bertujuan untuk memudahkan masyarakat untuk menjangkau sarana transaksi tanaman hias lebih mudah dan menjadi sarana pengetahuan terkait tanaman hias.

Ruang Lingkup

1. Perangkat Keras
 - processor : Intel Core i3
 - Memory : 4 GB
 - Graphics Card : NVIDIA 920MX
 - Storage : 1 TB
2. Perangkat Lunak
 - Framework : Django, Bootstrap
 - Database : SQLite
 - Text editor : Visual Studio Code

3. Lainnya

- Development Time : 2 bulan

Deskripsi perangkat lunak

Software yang dibuat merupakan platform website. Website ini akan menjadi wadah bagi para pecinta tanaman hias ataupun orang yang tertarik akan tanaman hias untuk melakukan transaksi jual-beli dan sebagai sarana informatif terkait tanaman hias.

Analisis user (user profile)

Target untuk pengguna platform ini yaitu masyarakat berusia 30 tahun keatas dikarenakan dengan analisis yang diperoleh bahwa pengoleksi tanaman hias merupakan seorang yang telah memiliki penghasilan atau telah menginjak rata rata umur 30 tahun keatas. Tetapi tidak menutup kemungkinan bagi kalangan muda sebagai target pemasaran produk disini.

User story

As a	Visit	So
Guest	Ingin mengunjungi website Orplant	Saya bisa menggunakan mesin pencari untuk mencari website Orplant
	Ingin melihat barang yang dijual	Saya bisa mengunjungi laman store pada website Orplant
	Ingin mencari informasi terkait tanaman hias	Saya bisa mengunjungi laman artikel pada website Orplant

As a	Sign up	So
Guest	Ingin memasukkan data nama lengkap	Saya bisa mengisi data nama lengkap
	Ingin memasukkan username baru	Saya bisa mengisi username baru
	Ingin memasukkan password baru	Saya bisa mengisi password baru
	Ingin memasukkan alamat email	Saya bisa mengisi alamat email
	Ingin memasukkan No. HP	Saya bisa mengisi No. HP
	Ingin memasukkan nama provinsi	Saya bisa mengisi nama provinsi

	Ingin memasukkan nama kabupaten	Saya bisa mengisi nama kabupaten
	Ingin memasukkan nama kecamatan	Saya bisa mengisi nama kecamatan
	Ingin memasukkan alamat lengkap tempat tinggal	Saya bisa mengisi alamat lengkap tempat tinggal
	Ingin mengupload foto KTP	Saya bisa mengupload foto KTP
	Ingin menekan tombol submit	Akun saya terdaftar sebagai member Orplant

As a	Sign in	So
Member	Ingin memasukkan username/alamat email	Saya bisa mengisi email
	Ingin memasukkan password	Saya bisa mengisi password
	Bisa menekan forgot password	Saya bisa mereset password saat saya lupa password yang telah saya buat
	Ingin menekan tombol login	Saya bisa masuk ke aplikasi dengan akun yang sudah saya buat

As a	Artikel	So
Guest, member	Ingin mencari informasi terkait tanaman hias	Saya bisa membuka laman artikel pada website Orplant
Member	Ingin memposting artikel terkait tanaman hias	Saya bisa menambahkan artikel pada laman website Orplant
	Ingin menghapus postingan yang telah saya buat dalam artikel	Saya bisa menghapus postingan yang telah saya buat
Admin	Ingin memantau laman artikel berjalan dengan semestinya	Saya bisa melihat kondisi laman artikel untuk memastikan berjalan sesuai dengan tujuan
	Ingin menghapus laman artikel	Saya bisa menghapus artikel yang tidak sesuai dengan ketentuan/tidak berkaitan dengan tanaman hias

As a	Menjual	So
Member	Ingin memposting produk yang akan dijual	Saya bisa menjual produk tanaman hias dengan memposting pada website Orplant
	Ingin menambahkan keterangan pada produk yang akan dijual	Saya bisa menambahkan keterangan terkait spesifikasi produk
	Ingin menambahkan foto dari produk yang akan dijual	Saya bisa menambahkan foto untuk lebih menarik pembeli
	Ingin menambahkan nominal harga produk	Saya bisa menambahkan nominal harga pada produk yang dijual
	Ingin menghapus postingan jika terjadi suatu kendala pada produk yang dijual	Saya menekan tombol delete pada postingan yang telah saya buat

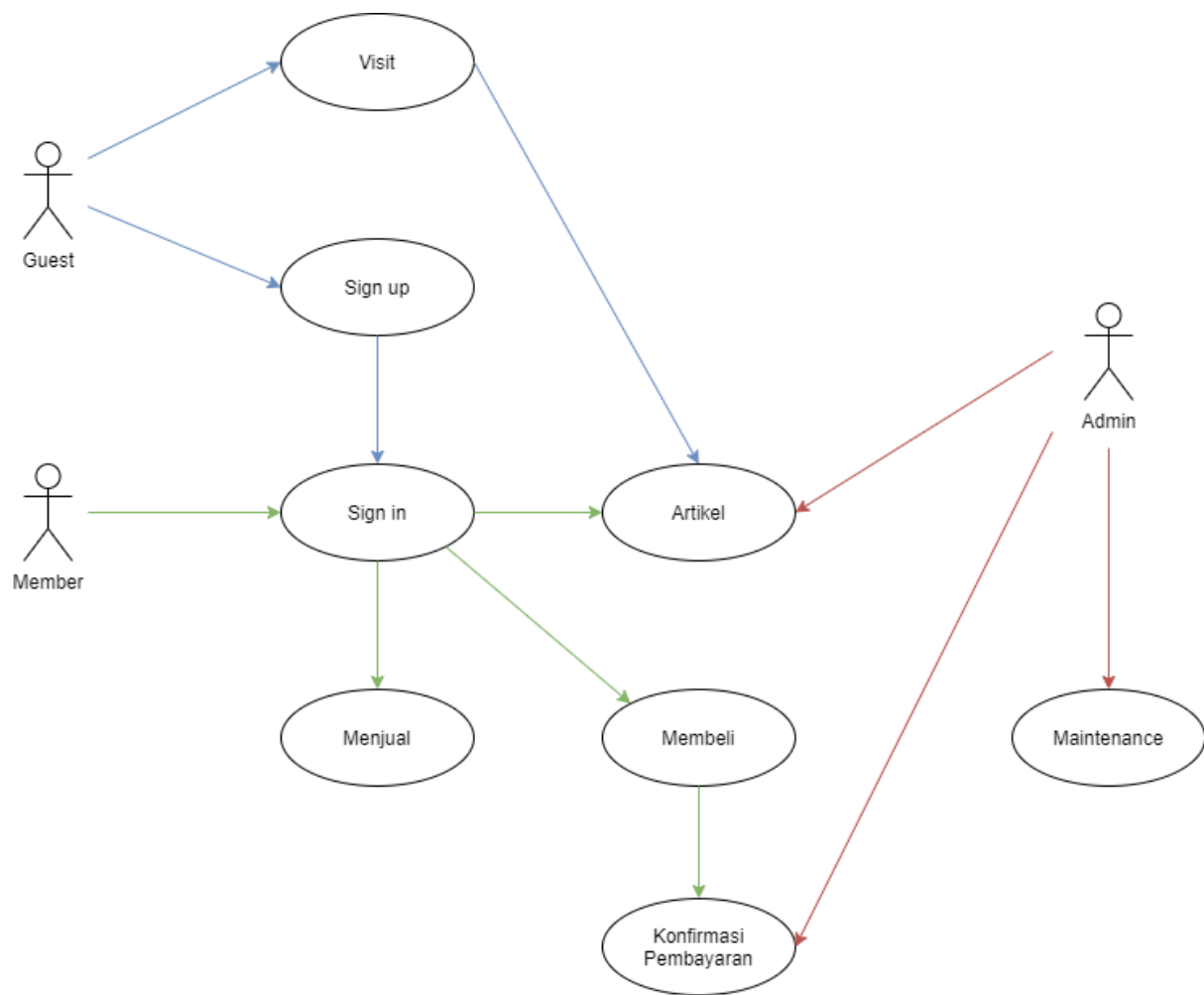
As a	Membeli	So
Member	Ingin melihat produk yang dijual	Saya bisa melihat produk yang dijual dengan menekan tools produk dihalaman beranda website Orplant
	Ingin membeli produk yang dijual	Saya bisa memilih terlebih dahulu barang yang ingin dibeli
	Ingin menambahkan jumlah produk kedalam cart	Saya bisa menekan tombol + untuk menambahkan produk yang akan dibeli
	Ingin mengurangi jumlah produk dalam cart	Saya bisa menekan tombol - untuk menambahkan produk yang akan dibeli
	Ingin menghapus/membatalkan produk yang akan dibeli	Saya bisa menekan tombol x untuk membatalkan pembelian produk dalam cart
	Ingin memilih metode pembayaran	Saya bisa menekan tools yang tertera dan memilih beberapa pilihan metode pembayaran

	Ingin mengupload bukti pembayaran	Saya bisa menekan tombol upload foto yang tertera pada laman yang tersedia
	Ingin mendapat konfirmasi validasi dari bukti pembayaran yang telah diupload	Saya bisa mengecek konfirmasi dengan menekan tools notifikasi

As a	Konfirmasi pembayaran	So
Member	Ingin mengupload bukti pembayaran produk yang dibeli	Saya bisa menekan tombol upload foto yang tertera pada laman yang tersedia
	Ingin mendapatkan balasan konfirmasi dari foto yang telah diupload	Saya bisa mengecek konfirmasi dengan menekan tools notifikasi
Admin	Ingin mengkonfirmasi pembayaran dari pembeli	Saya bisa mengkonfirmasi bukti pembayaran jika valid
	Ingin membatalkan konfirmasi pembayaran dari pembeli	Saya bisa membatalkan konfirmasi jika bukti pembayaran tidak valid

As a	Maintenance	So
Admin	Ingin melakukan pembaharuan terhadap sistem	Saya bisa mengupdate sistem menjadi lebih baik dan berguna bagi user
	Ingin melakukan perbaikan terhadap sistem	Saya bisa memulihkan permasalahan terhadap sistem yang terkena bug
	Ingin meningkatkan ketahanan sistem terhadap masalah yang memungkinkan terjadi	Saya bisa meningkatkan ketahanan sistem agar user lebih merasa aman dalam melakukan transaksi jual beli dan merasa aman terhadap informasi pribadi user

Use case diagram



Use case description

Nama use case	UC1: Visit
Aktor	Guest
Deskripsi	Mengunjungi laman website Orplant
Successful completion	<ol style="list-style-type: none"> 1. Guest mencari laman website Orplant melalui laman pencarian internet 2. Memasuki laman website Orplant 3. Sistem menampilkan laman home Orplant kepada guest 4. Guest melihat laman beranda dan dapat mengklik beberapa tombol yang ditampilkan 5. Guest dapat melihat produk yang dijual tetapi tidak dapat mengakses untuk membeli 6. Guest dapat melihat isi dari artikel tetapi tidak memiliki akses untuk menambahkan artikel
Alternatif	<ol style="list-style-type: none"> 1. Guest mengklik link laman website dari copy link 2. Memasuki laman website Orplant 3. Sistem menampilkan laman home Orplant kepada guest

	<ol style="list-style-type: none"> Guest melihat laman Home dan dapat mengklik beberapa tombol yang ditampilkan Guest dapat melihat produk yang dijual tetapi tidak dapat mengakses untuk membeli Guest dapat melihat isi dari artikel tetapi tidak memiliki akses untuk menambahkan artikel
<i>Precondition</i>	Guest mengunjungi laman website Orplant
<i>Postcondition</i>	Tidak ada
Asumsi	Guest mengunjungi laman website Orplant untuk melihat atau mencari informasi terkait tanaman hias melalui laman forum.

Nama use case	UC2: Sign up
Aktor	Guest
Deskripsi	Mendaftarkan diri untuk menjadi member website Orplant
<i>Successfull completion</i>	<ol style="list-style-type: none"> Guest memasuki laman website Orplant Menekan “My Profile” Sistem menampilkan laman dengan pilihan Sign up dan Sign in Guest menekan Sign up Wajib mengisi data yang dibutuhkan serta memberikan foto KTP Menginput email dan password Menekan tombol submit Guest menerima email sebagai verifikasi Menekan link yang ada pada email yang telah diterima Guest memasuki laman beranda Orplant Guest berhasil mendaftarkan diri sebagai member
Alternatif	<ol style="list-style-type: none"> Guest memasuki laman website Orplant Menekan “My Profile” Sistem menampilkan laman dengan pilihan Sign up dan Sign in Guest menekan Sign up Wajib mengisi data yang dibutuhkan serta memberikan foto KTP Menginput nomor handphone dan password Menekan tombol submit Guest menerima sms sebagai verifikasi Menekan link yang ada pada email yang telah diterima Guest memasuki laman beranda Orplant Guest berhasil mendaftarkan diri sebagai member
<i>Precondition</i>	Belum mendaftarkan diri sebagai member
<i>Postcondition</i>	Konfirmasi dari admin dan verifikasi data
Asumsi	Guest ingin mendaftarkan diri sebagai member

Nama use case	UC3: Sign in
Aktor	Member
Deskripsi	Memasuki laman website Orplant dengan memiliki akses lebih dari pada Guest
<i>Successful completion</i>	<ol style="list-style-type: none"> 1. Membuka laman website Orplant 2. Menekan “My Profile” 3. Sistem menampilkan laman dengan pilihan Sign up dan Sign in 4. Menekan Sign in 5. Memasukkan email/no handphone yang telah terdaftar 6. Memasukkan password 7. Member akan memasuki laman beranda Orplant 8. Member dapat memiliki akses untuk menjual dan membeli sesuatu produk, dapat memposting pada laman artikel
Alternatif	<ol style="list-style-type: none"> 1. Membuka laman website Orplant 2. Menekan “My Profile” 3. Sistem menampilkan laman dengan pilihan Sign up dan Sign in 4. Menekan Sign in 5. Memasukkan email/no handphone yang telah terdaftar 6. Menekan tombol forgot password 7. Mengecek email/sms untuk mendapat kode verifikasi 8. Memasukkan kode tersebut pada laman website Orplant 9. Mengganti password baru 10. Memasukkan email/no hp dan password yang baru 11. Member akan memasuki laman beranda Orplant 12. Member dapat memiliki akses untuk menjual dan membeli sesuatu produk, dapat memposting pada laman artikel
<i>Precondition</i>	Sudah mendaftarkan diri menjadi member
<i>Postcondition</i>	Tidak ada
Asumsi	Guest telah memiliki akun sebagai member Orplant

Nama use case	UC4: Artikel
Aktor	Member dan admin
Deskripsi	Memberikan atau memperoleh informasi terkait tanaman hias
<i>Successful completion</i>	<ol style="list-style-type: none"> 1. Telah melakukan Sign in 2. Memasuki laman forum 3. Memposting sebuah artikel terkait tanaman hias 4. Dapat menghapus artikel yang telah dibuat
Alternatif	Tidak ada

<i>Precondition</i>	Mencari informasi atau memberikan informasi terkait tanaman hias
<i>Postcondition</i>	Tidak ada
Asumsi	Member mencari informasi atau memberikan informasi tanaman hias sedangkan admin melakukan pemantauan terhadap artikel yang ada pada laman artikel

Nama use case	UC5: Menjual
Aktor	Member
Deskripsi	Menambahkan produk untuk dijual
<i>Successful completion</i>	<ol style="list-style-type: none"> 1. Telah melakukan Sign in 2. Memasuki laman “My Account” atau profil 3. Menekan tombol My Shop 4. Menambahkan produk tanaman hias yang akan dijual 5. Sertakan bukti foto dan keterangan terkait produk yang dijual 6. Barang diposting pada laman tersebut 7. Menunggu adanya pembeli yang tertarik pada produk yang dijual
Alternatif	Tidak ada
<i>Precondition</i>	Memiliki produk untuk dijual dan sudah menjadi member
<i>Postcondition</i>	Menunggu adanya pembeli
Asumsi	Menjual barang berupa tanaman hias atau kebutuhan tanaman hias untuk menambah kebutuhan ekonomi

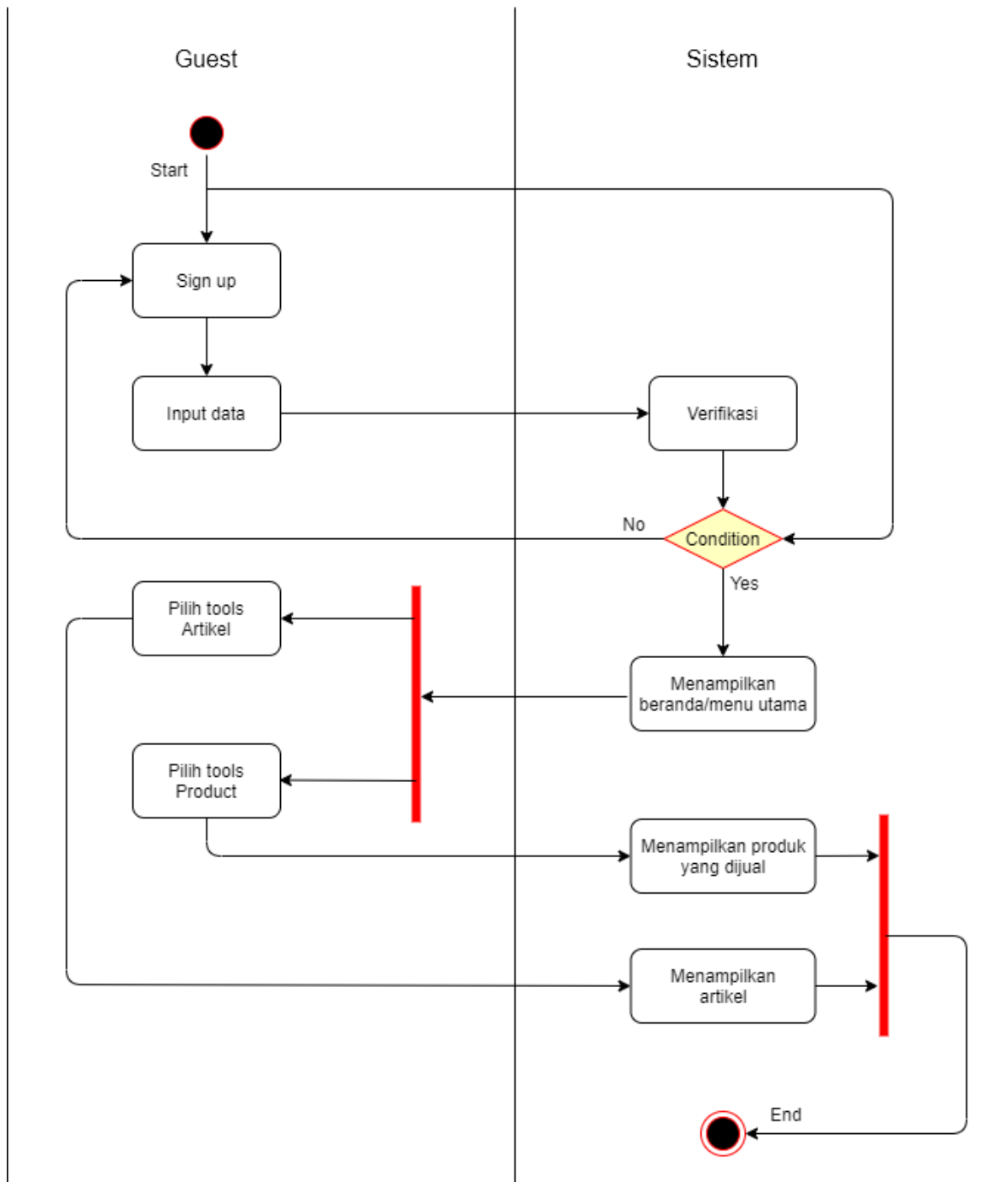
Nama use case	UC6: Membeli
Aktor	Member
Deskripsi	Membeli tanaman hias atau membeli kebutuhan terkait tanaman hias
<i>Successful completion</i>	<ol style="list-style-type: none"> 1. Telah melakukan Sign in 2. Memasuki laman beranda 3. Menekan tombol Store 4. Memilih produk yang akan dibeli 5. Menambah produk kedalam cart 6. Melakukan proses pembayaran 7. Memfoto bukti pembayaran 8. Upload foto bukti pembayaran 9. Menunggu konfirmasi dari admin 10. Setelah disetujui, barang akan mulai diproses untuk dikirim 11. Barang sampai pada pembeli
Alternatif	Tidak ada
<i>Precondition</i>	Sudah menjadi member
<i>Postcondition</i>	Menunggu konfirmasi pembayaran dari admin
Asumsi	Tertarik pada produk yang dijual

Nama use case	UC7: Konfirmasi pembayaran
Aktor	Admin
Deskripsi	Mengkonfirmasi pembayaran dengan mengecek bukti pembayaran dari member yang membeli produk
<i>Successful completion</i>	<ol style="list-style-type: none"> 1. Admin menerima pemberitahuan dari sistem 2. Mengecek bukti pembayaran 3. Melakukan konfirmasi pada bukti pembayaran 4. Penjual dapat melanjutkan proses pengemasan untuk mengirimkan barang pada pembeli
Alternatif	<ol style="list-style-type: none"> 1. Admin menerima pemberitahuan dari sistem 2. Mengecek bukti pembayaran 3. Melakukan konfirmasi pada bukti pembayaran 4. Bukti pembayaran tidak valid 5. Admin menolak permintaan dari pembeli 6. Penjual tidak melanjutkan proses pengemasan barang
<i>Precondition</i>	Pembeli telah melakukan pembayaran dan mengupload bukti pembayaran
<i>Postcondition</i>	Menunggu adanya pembeli
Asumsi	Pembeli telah melakukan pembayaran sehingga admin dapat melakukan konfirmasi terhadap bukti pembayaran yang telah diupload

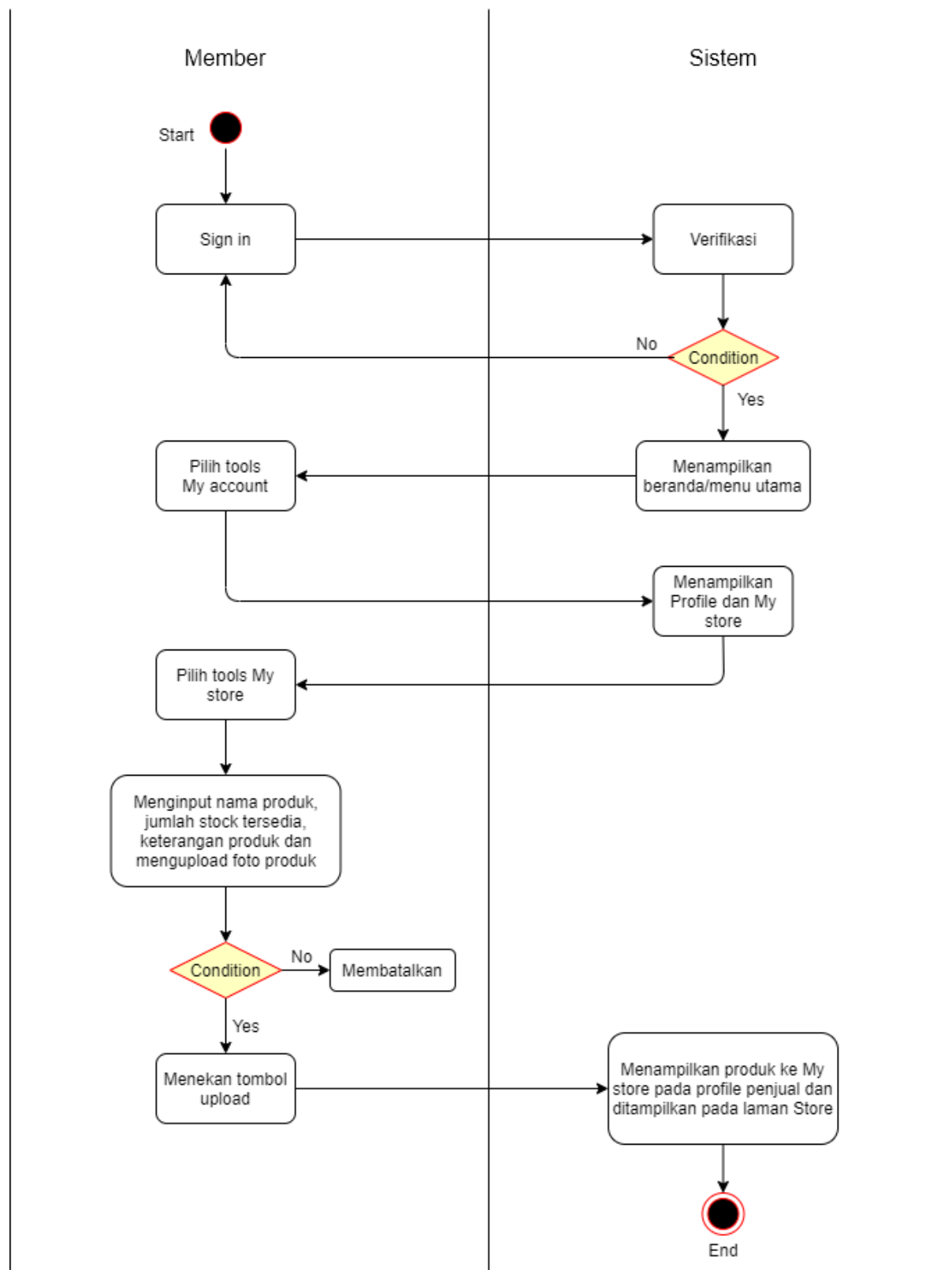
Nama use case	UC8: Maintenance
Aktor	Admin
Deskripsi	Melakukan perawatan kinerja pada website
<i>Successful completion</i>	<ol style="list-style-type: none"> 1. Admin memberikan pemberitahuan terkait maintenance dan estimasi selesai pengerjaannya 2. Web ditutup selama maintenance 3. Melakukan perbaikan jika terjadi bug 4. Memperbaiki permasalahan 5. Membuka kembali website yang telah diperbaiki masalahnya
Alternatif	Tidak ada
<i>Precondition</i>	Terdapat bug
<i>Postcondition</i>	Menyelesaikan sistem yang terkena bug/eror
Asumsi	Adanya laporan dari member terhadap sistem

Activity diagram

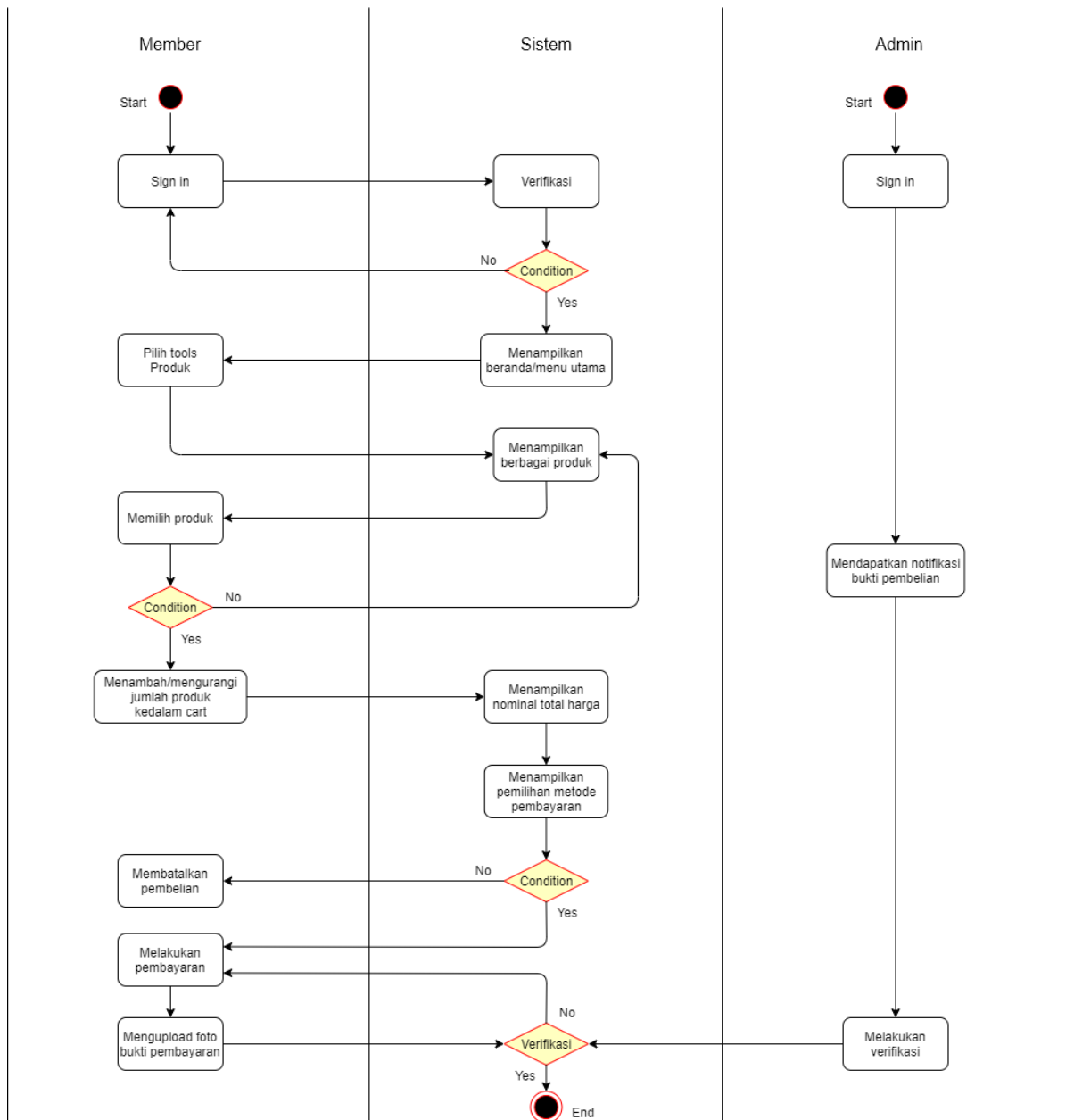
Activity diagram guest melakukan sign up dan melakukan visit/kunjungan ke website Orplant



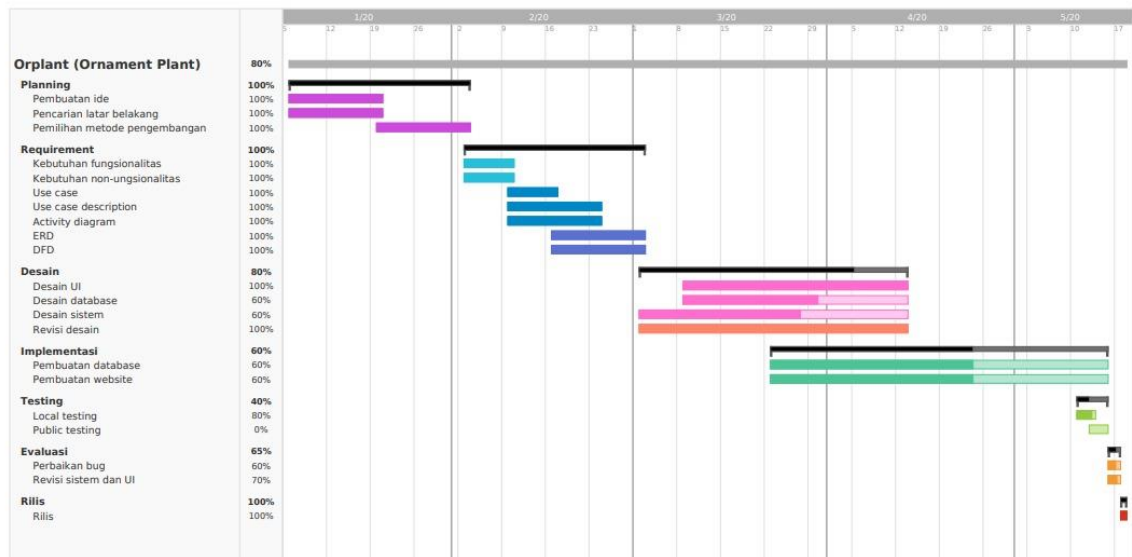
Activity diagram member melakukan penjualan



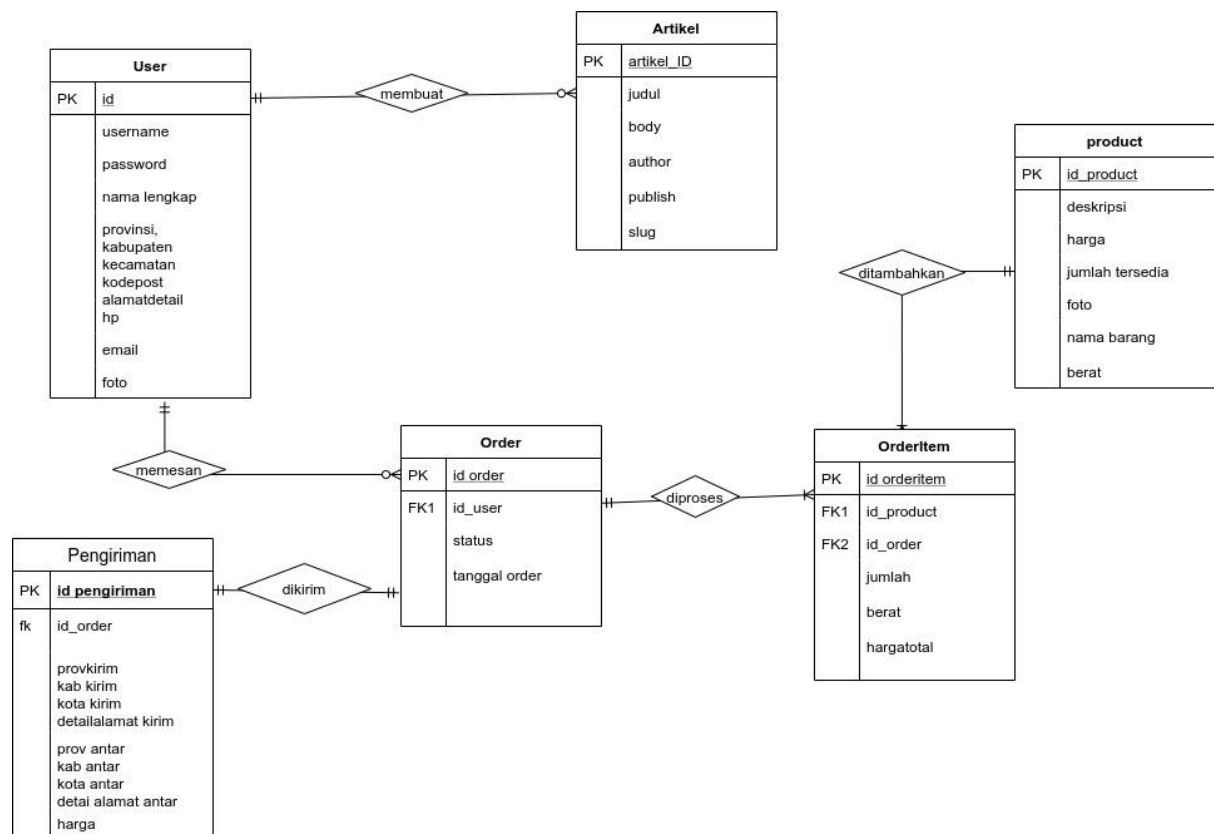
Activity diagram member melakukan pembelian dan konfirmasi pembayaran oleh admin



Gantt chart



ERD



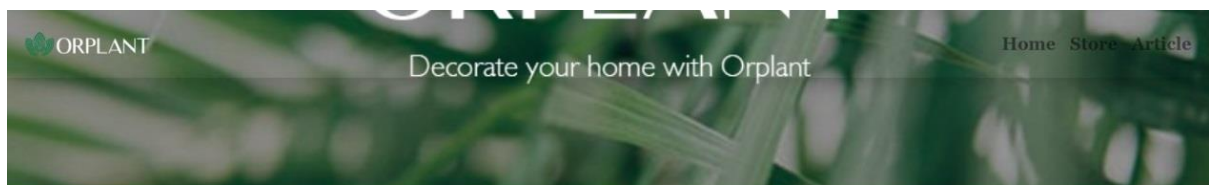
Hasil implementasi perangkat lunak

Beranda



About Us

Orplant adalah platform tanaman hias yang menjadi wadah para pencinta tanaman hias untuk melakukan transaksi jual-beli (Store) dan berbagi informasi mengenai tanaman hias (Artikel).



About Us

Orplant adalah platform tanaman hias yang menjadi wadah para pencinta tanaman hias untuk melakukan transaksi jual-beli (Store) dan berbagi informasi mengenai tanaman hias (Artikel).



Contact

- Orplant
- @Orplant
- Orplant

Forum




Sign in

Email address We'll never share your email with anyone else.

Password

☐ Check me out

Artikel



 **Membuat Artikel** [Home](#) [Store](#) [Article](#)


Penulis

Judul

Body


Store

 **ORPLANT** [Store](#) [Login](#) 




Lidah buaya

\$500.00




Lidah Mertua

\$3255465557.00




Aglonema


\$52353434556.00



Pothos




Calathea




Calathea

meet.google.com is sharing your screen.

Cart





 Store


Login 


[← Continue Shopping](#)

Items: 3 Total: \$107962334669.00

Checkout

	Item	Price	Quantity	Total
	Lidah Mertua	\$3255465557.00	1 	\$3255465557.00
	Aglonema	\$52353434556.00	2 	\$104706869112.00

 Store

Login 

[← Back to Cart](#)

Shipping Information:

kk

Bogor


ter

r

rte


Continue

Order Summary



Lidah Mertua

\$3255465557.00x1





Aglonema

\$52353434556.00x2

Items: 3

Total: \$107962334669.00

 Store

Login 

[← Back to Cart](#)

Shipping Information:

kk

Bogor


ter

r

rte


Paypal Options [Make payment](#)

Order Summary



Lidah Mertua

\$3255465557.00x1

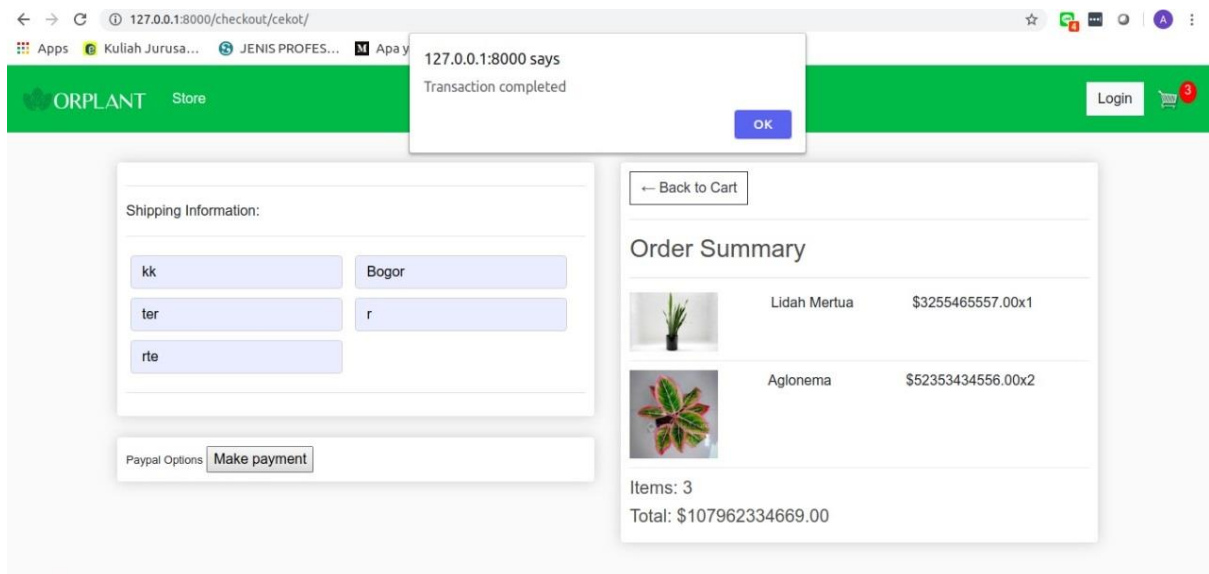


Aglonema

\$52353434556.00x2

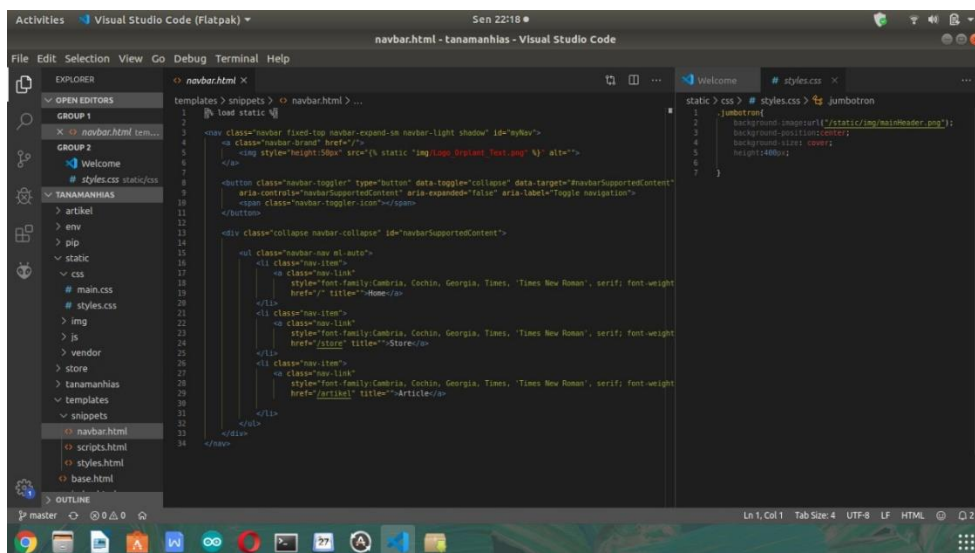
Items: 3

Total: \$107962334669.00

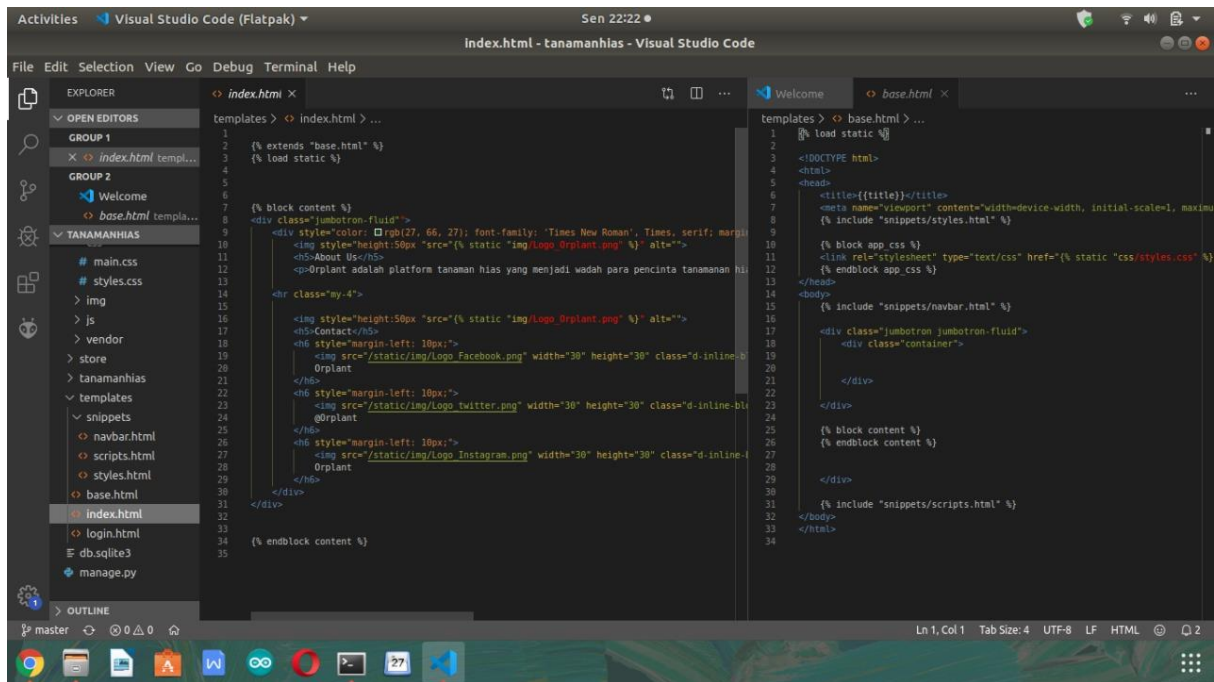


Screenshot coding :

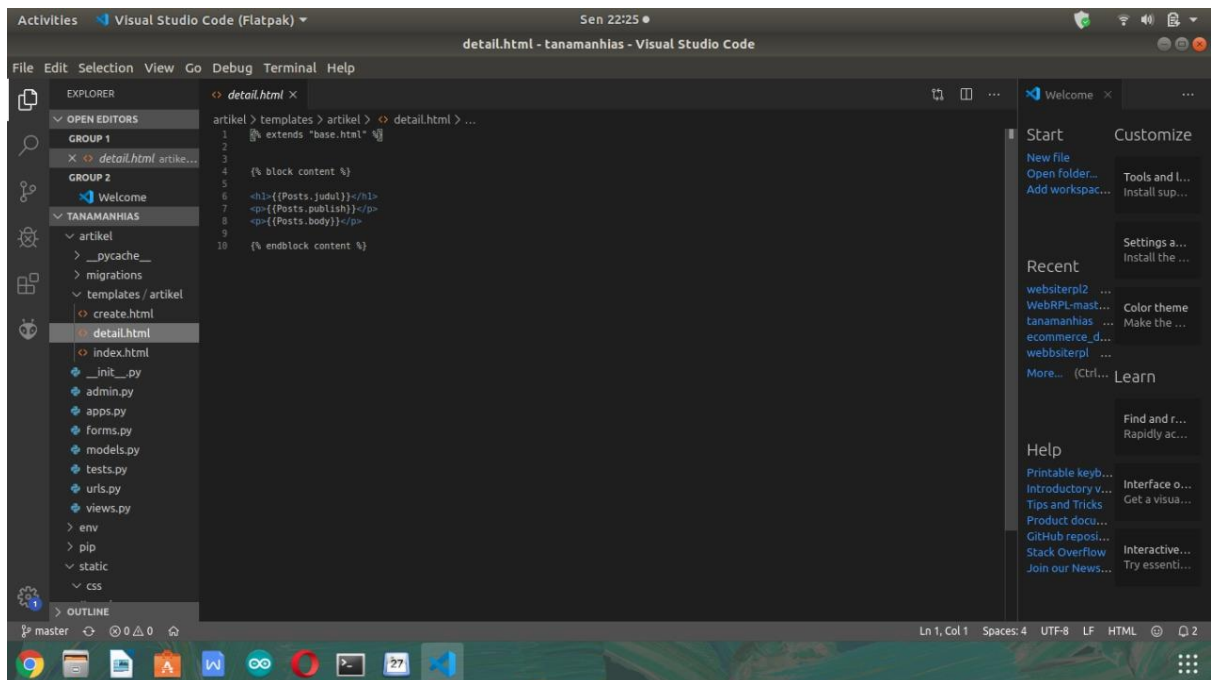
Home atas



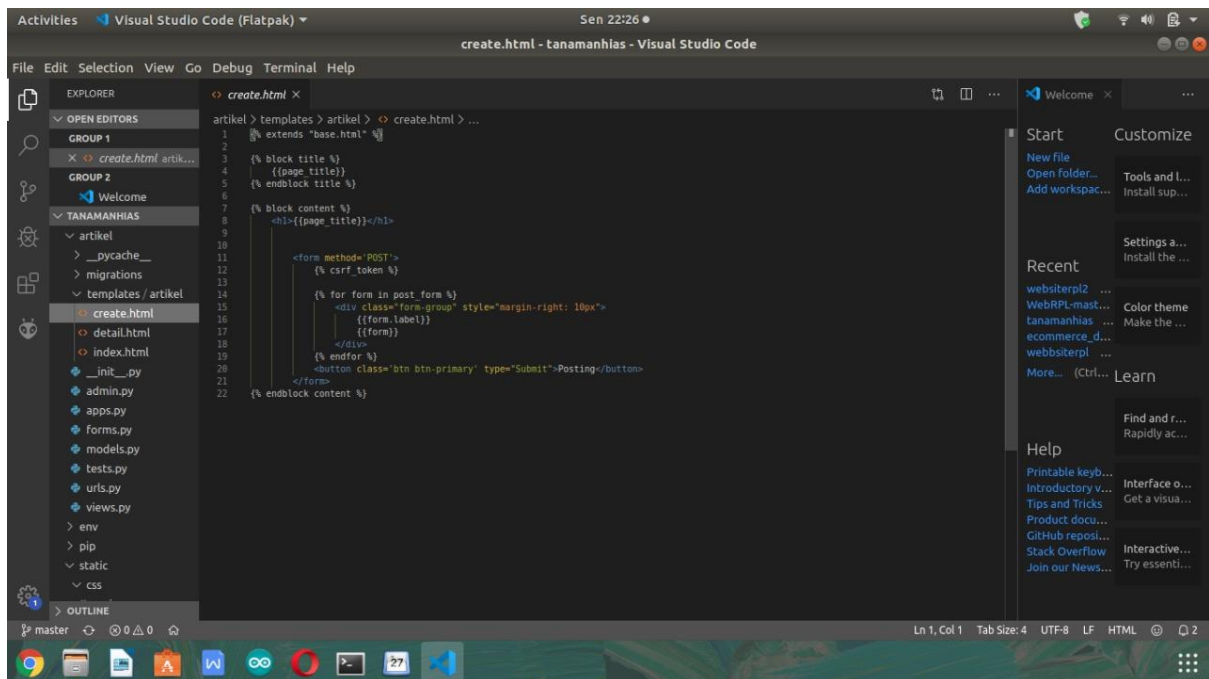
Home bawah



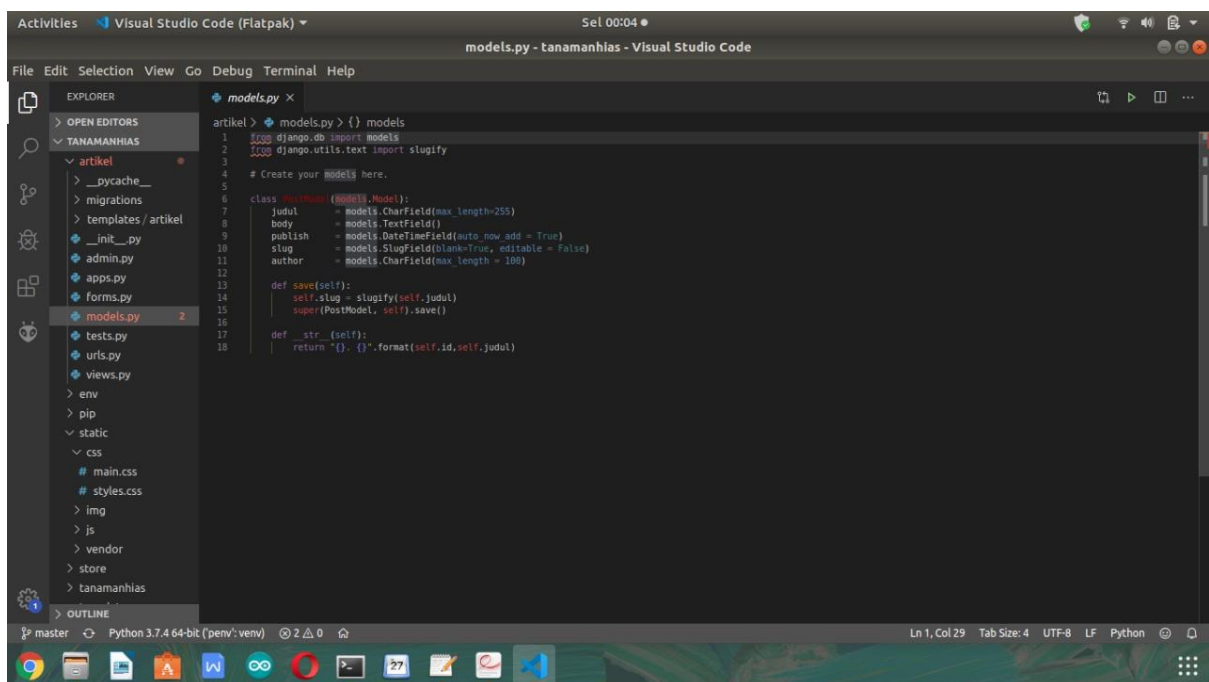
Tampilan awal artikel



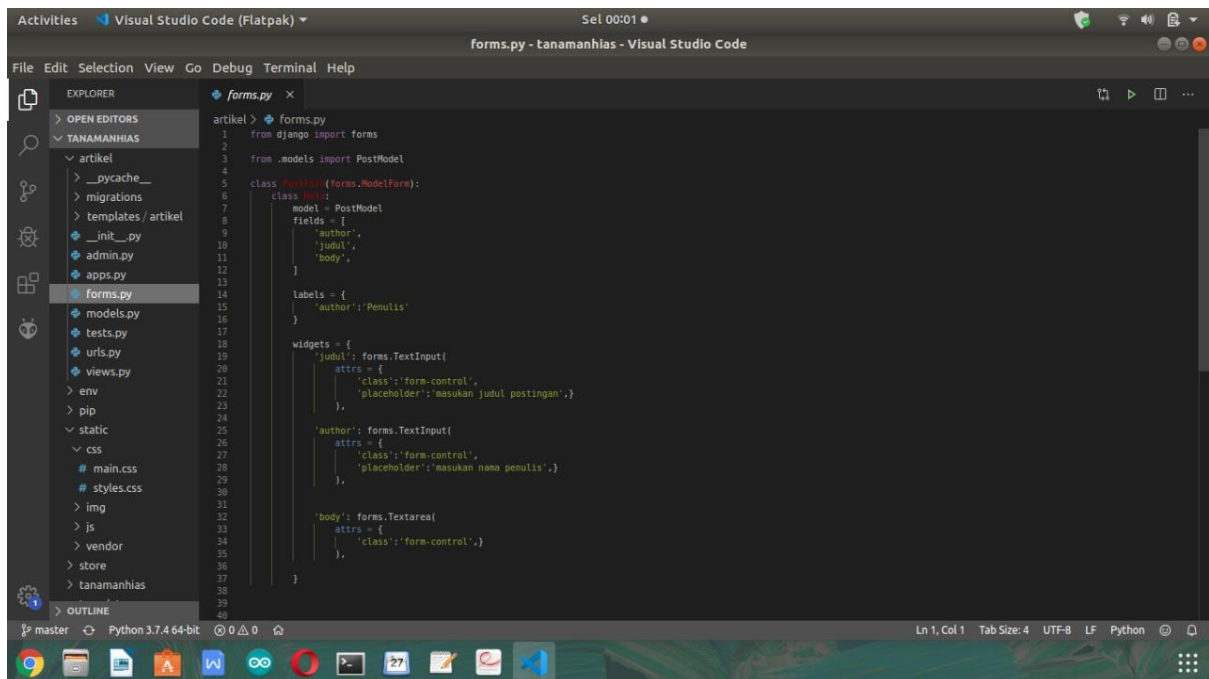
Laman membuat artikel



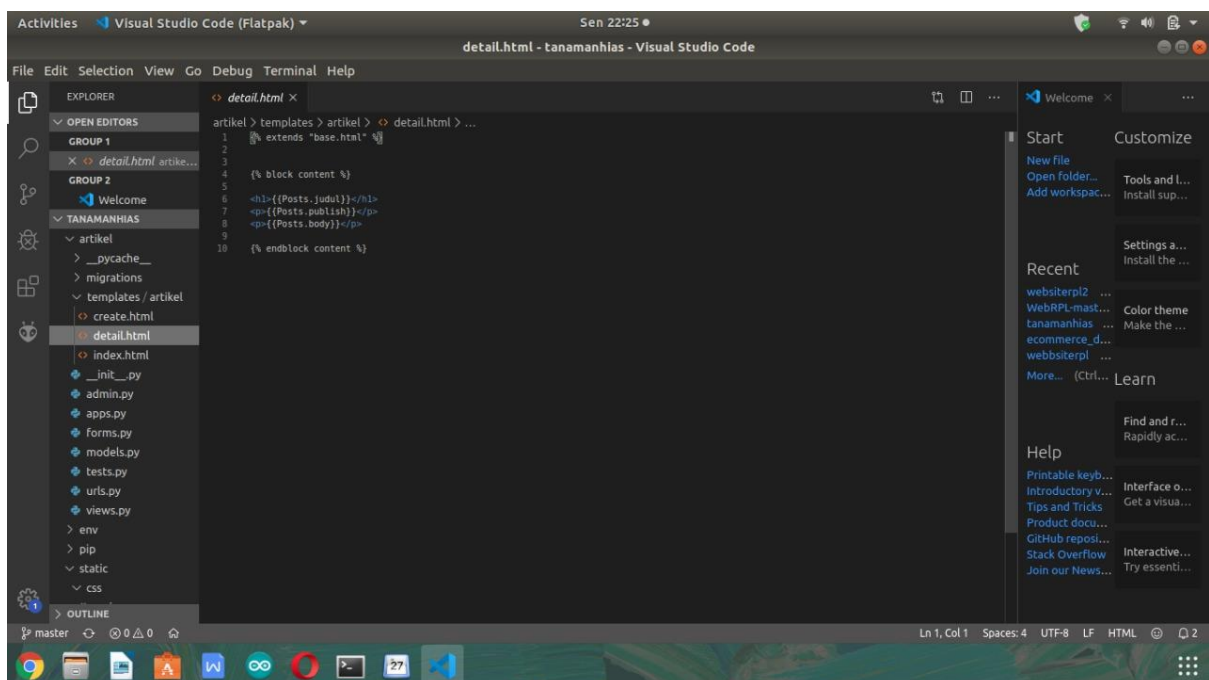
Model untuk artikel



Form membuat artikel



Tampilan 1 post secara keseluruhan



Sign in HTML

```
templates > login.html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>{{judul}}</title>
5  </head>
6  <body>
7      <h1>{{judul}}</h1>
8      <h2>{{subjudul}}</h2>
9
10
11     <form method="POST">
12         {% csrf_token %}
13         <label for="username">Username:</label>
14         <input type="text" name="username" id="username">
15         <br>
16         <label for="password"> Password:</label>
17         <input type="password" name="password" id="password">
18
19         <input type="submit" value="Login">
20     </form>
21
22 </body>
23 </html>
24
25
```

Cart HTML

```
terminal  Help  cart.html - tanamanhias - Visual Studio Code
store > templates > store > cart.html
1  {% extends 'store/index.html' %}
2  {% load static %}
3  {% block content %}
4
5      <div class="row">
6          <div class="col-lg-12">
7              <div class="box-element">
8
9                  <a class="btn btn-outline-dark" href="{% url 'store' %}">Continue Shopping</a>
10
11                  <br>
12                  <br>
13                  <table class="table">
14                      <tr>
15                          <th><h5>Items: <strong>{{order.get_cart_items}}</strong></h5></th>
16                          <th><h5>Total:<strong> ${{order.get_cart_total|floatformat:2}}</strong></h5></th>
17                      <tr>
18                          <td><a style="float:right; margin:5px;" class="btn btn-success" href="{% url 'checkout' %}">Checkout</a>
19                      </td>
20                  </tr>
21                  </table>
22              </div>
23          </div>
24      </div>
25
26      <div class="box-element">
27          <div class="cart-row">
28              <div style="flex:2"></div>
29              <div style="flex:2"><strong>Item</strong></div>
30              <div style="flex:1"><strong>Price</strong></div>
31              <div style="flex:1"><strong>Quantity</strong></div>
32              <div style="flex:1"><strong>Total</strong></div>
33          </div>
34      </div>
35  </block content %>
```



```
nal Help carhtml - tanamanhias - Visual Studio Code
carhtml x login.html
re > templates > store > carhtml
4
5
6 <br>
7 <div class="box-element">
8   <div class="cart-row">
9     <div style="flex:2"></div>
10    <div style="flex:2"><strong>Item</strong></div>
11    <div style="flex:1"><strong>Price</strong></div>
12    <div style="flex:1"><strong>Quantity</strong></div>
13    <div style="flex:1"><strong>Total</strong></div>
14  </div>
15  {% for item in items %}
16  <div class="cart-row">
17    <div style="flex:2"></div>
18    <div style="flex:2"><p>{{item.product.name}}</p></div>
19    <div style="flex:1"><p>${{item.product.price|floatformat:2}}</p></div>
20    <div style="flex:1">
21      <p class="quantity">{{item.quantity}}</p>
22      <div class="quantity">
23        
26    </div>
27    <div style="flex:1"><p>${{item.get_total|floatformat:2}}</p></div>
28  </div>
29  {% endfor %}
30 </div>
31 </div>
32 </div>
33 {% endblock content %}
```

Checkout HTML

```
terminal Help checkout.html - tanamanhias - Visual Studio Code
carhtml x checkout.html x login.html
store > templates > store > checkout.html
1 {% extends 'store/index.html' %}
2 {% load static %}
3 {% block content %}
4
5 <div class="row">
6   <div class="col-lg-6">
7     <div class="box-element" id="form-wrapper">
8       <form id="form">
9         <div id="user-info">
10          <div class="form-field">
11            <input required class="form-control" type="text" name="name" placeholder="Name..">
12          </div>
13          <div class="form-field">
14            <input required class="form-control" type="email" name="email" placeholder="Email..">
15          </div>
16        </div>
17
18        <div id="shipping-info">
19          <hr>
20          <p>Shipping Information:</p>
21          <hr>
22          <div class="form-field">
23            <input class="form-control" type="text" name="address" placeholder="Address..">
24          </div>
25          <div class="form-field">
26            <input class="form-control" type="text" name="city" placeholder="City..">
27          </div>
28          <div class="form-field">
29            <input class="form-control" type="text" name="state" placeholder="State..">
30          </div>
31          <div class="form-field">
32            <input class="form-control" type="text" name="zipcode" placeholder="Zip code..">
33          </div>
34          <div class="form-field">
```

View dan model store

```
models.py - tanamanhias - Visual Studio Code

models.py
1 from django.contrib.auth.models import User
2 from django.db import models
3
4 # Create your models here.
5
6 class Customer(models.Model):
7     user = models.OneToOneField(User, null=True, blank=True)
8     name = models.CharField(max_length=200, null=True)
9     email = models.CharField(max_length=200)
10
11     def __str__(self):
12         return self.name
13
14 class Product(models.Model):
15     name = models.CharField(max_length=200)
16     price = models.FloatField()
17     digital = models.NullBooleanField(default=False, null=True)
18     image = models.ImageField(null=True, blank=True)
19
20     def __str__(self):
21         return self.name
22
23 @property
24 def imageURL(self):
25     try:
26         url = self.image.url
27     except:
28         url = ''
29     return url
30
31 class Order(models.Model):
32     customer = models.ForeignKey(Customer, on_delete=models.SET_NULL, null=True)
33     date_ordered = models.DateTimeField(auto_now_add=True)
34     complete = models.BooleanField(default=False)
35     transaction_id = models.CharField(max_length=100, null=True)
36
37     def __str__(self):
38         return str(self.id)
```

```
views.py
1 # view nya si app store
2 from django.shortcuts import render
3 from .models import *
4 from django.http import JsonResponse
5 import json
6 import datetime
7 from .utils import cookieCart, cartData, guestOrder
8
9 from django.contrib.auth import authenticate, login, logout
10
11 from django.shortcuts import render, redirect
12
13
14 def store(request):
15     data = cartData(request)
16     cartItems = data['cartItems']
17     order = data['order']
18     items = data['items']
19
20
21     print(request.method)
22     if request.method == 'POST':
23         print(request.POST['Logout'])
24         if request.POST['Logout'] == "Logout":
25             logout(request)
26             return redirect('login')
27
28
29 products = Product.objects.all()
30 context = {'products':products, 'cartItems':cartItems}
31 return render(request, 'store/store.html', context)
32
33 def cart(request):
34     data = cartData(request)
35     cartItems = data['cartItems']
36     order = data['order']
```

Store dan index HTML

```
store > templates > store > index.html
56
57 console.log('cart:', cart)
58
59 </script>
60
61 <title>HOME</title>
62 </head>
63 <body>
64
65
66 <nav class="navbar navbar-expand-lg navbar-dark bg-success">
67     <a class="navbar-brand" href="/"></a>
68     
69     <a></a>
70     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarCollapse" data-ri
71     <span class="navbar-toggler-icon"></span>
72 </button>
73
74 <div class="collapse navbar-collapse" id="navbarSupporter">
75     <ul class="navbar-nav mr-auto">
76         <li class="nav-item active">
77             <a class="nav-link" href="{% url 'store' %}">Store</a>
78         </li>
79     </ul>
80
81 <div class="form-inline my-2 my-lg-0">
82     <a href="/login" class="btn btn-light">Login</a>
83
84     <a href="{% url 'cart' %}">
85         
86     </a>
87     <p id="cart-total">{{ cartItems }}</p>
88
89 </div>
90
91 </div>
92
93 </div>
```

```
store > templates > store > store.html
1 {% extends 'store/index.html' %}
2 {% load static %}
3 {% block content %}
4 <form method="POST" class="nav-item">
5     {% csrf_token %}
6     <input type="submit" name="Logout" value="Logout" class="btn btn-light">
7 </form>
8
9 <div class="row">
10
11     {% for product in products %}
12     <div class="col-lg-4">
13         
14         <div class="box-element product">
15             <h6><strong>{{ product.name }}</strong></h6>
16             <hr>
17             <button data-product="{{ product.id }}" data-action="add" class="btn btn-outline-success">Add to Cart</button>
18             <a class="btn btn-outline-success" href="{% url 'detail' %}?id={{ product.id }}">View Details</a>
19             <div style="display: inline-block; float: right;">
20                 <h4 style="display: inline-block; float: right;">{{ product.price|floatformat:2 }}</h4>
21             </div>
22         </div>
23     </div>
24     {% endfor %}
25 </div>
26
27 {% endblock content %}
```



```
store > templates > store > checkout.html
24     </div>
25     <div class="form-field">
26         <input class="form-control" type="text" name="city" placeholder="City..">
27     </div>
28     <div class="form-field">
29         <input class="form-control" type="text" name="state" placeholder="State..">
30     </div>
31     <div class="form-field">
32         <input class="form-control" type="text" name="zipcode" placeholder="Zip code..">
33     </div>
34     <div class="form-field">
35         <input class="form-control" type="text" name="country" placeholder="Zip code..">
36     </div>
37 </div>
38
39 <hr>
40 <input id="form-button" class="btn btn-success btn-block" type="submit" value="Continue">
41 </form>
42 </div>
43
44 <br>
45 <div class="box-element hidden" id="payment-info">
46     <small>Paypal Options</small>
47     <button id="make-payment">Make payment</button>
48 </div>
49
50 </div>
51
52 <div class="col-lg-6">
53     <div class="box-element">
54         <a class="btn btn-outline-dark" href="{% url 'cart' %}">Back to Cart</a>
55     <hr>
56     <h3>Order Summary</h3>
57     <hr>
```

```
terminal Help checkout.html - tanamanhias - Visual Studio Code
store > templates > store > checkout.html
44 <br>
45 <div class="box-element hidden" id="payment-info">
46     <small>Paypal Options</small>
47     <button id="make-payment">Make payment</button>
48 </div>
49
50 </div>
51
52 <div class="col-lg-6">
53     <div class="box-element">
54         <a class="btn btn-outline-dark" href="{% url 'cart' %}">Back to Cart</a>
55     <hr>
56     <h3>Order Summary</h3>
57     <hr>
58     {% for item in items %}
59     <div class="cart-row">
60         <div style="flex:2"></div>
61         <div style="flex:2"><p>{{item.product.name}}</p></div>
62         <div style="flex:1"><p>${{item.product.price|floatformat:2}}</p></div>
63         <div style="flex:1"><p>x{{item.quantity}}</p></div>
64     </div>
65     {% endfor %}
66     <h5>Items: {{order.get_cart_items}}</h5>
67     <h5>Total: ${{order.get_cart_total|floatformat:2}}</h5>
68 </div>
69 </div>
70 </div>
71
72 <script type="text/javascript">
73     var shipping = '{{order.shipping}}'
74     var total = '{{order.get_cart_total|floatformat:2}}'
75
76     if (shipping == 'False'){
```

```
car.html checkout.html X login.html
ore > templates > store > checkout.html
76 if (shipping == 'false'){
77   document.getElementById('shipping-info').innerHTML = ''
78 }
79
80 if (user != 'AnonymousUser'){
81   document.getElementById('user-info').innerHTML = ''
82 }
83
84 if (shipping == 'false' && user != 'AnonymousUser'){
85   //Hide entire form if user is logged in and shipping is false
86   document.getElementById('form-wrapper').classList.add("hidden");
87   //Show payment if logged in user wants to buy an item that does not require shipping
88   document.getElementById('payment-info').classList.remove("hidden");
89 }
90
91 var form = document.getElementById('form')
92 form.addEventListener('submit', function(e){
93   e.preventDefault()
94   console.log('Form Submitted...')
95   document.getElementById('form-button').classList.add("hidden");
96   document.getElementById('payment-info').classList.remove("hidden");
97 })
98
99 document.getElementById('make-payment').addEventListener('click', function(e){
100   submitFormData()
101 })
102
103 function submitFormData(){
104   console.log('Payment button clicked')
105
106   var userFormData = {
107     'name':null,
108     'email':null,
109     'total':total,
110   }
```

```
car.html checkout.html X login.html
store > templates > store > checkout.html
109     'total':total,
110   }
111
112   var shippingInfo = {
113     'address':null,
114     'city':null,
115     'state':null,
116     'zipcode':null,
117   }
118
119   if (shipping != 'false'){
120     shippingInfo.address = form.address.value
121     shippingInfo.city = form.city.value
122     shippingInfo.state = form.state.value
123     shippingInfo.zipcode = form.zipcode.value
124   }
125
126   if (user == 'AnonymousUser'){
127     userFormData.name = form.name.value
128     userFormData.email = form.email.value
129   }
130
131   console.log('Shipping Info:', shippingInfo)
132   console.log('User Info:', userFormData)
133
134   var url = "/process_order/"
135   fetch(url, {
136     method:'POST',
137     headers:{
138       'Content-Type':'applicaiton/json',
139       'X-CSRFToken':csrftoken,
140     },
141     body:JSON.stringify({'form':userFormData, 'shipping':shippingInfo}),
142   })
143
144   .then((response) => {
145     .then((data) => {
```

```
store > templates > store > checkout.html
125
126     if (user == 'AnonymousUser'){
127         userFormData.name = form.name.value
128         userFormData.email = form.email.value
129     }
130
131     console.log('Shipping Info:', shippingInfo)
132     console.log('User Info:', userFormData)
133
134     var url = "/process_order/"
135     fetch(url, {
136         method: 'POST',
137         headers: {
138             'Content-Type': 'application/json',
139             'X-CSRFToken': csrftoken,
140         },
141         body: JSON.stringify({'form': userFormData, 'shipping': shippingInfo}),
142     })
143
144     .then((response) => response.json())
145     .then((data) => {
146         console.log('Success:', data);
147         alert('Transaction completed');
148
149         cart = {}
150         document.cookie = 'cart=' + JSON.stringify(cart) + ";domain=;path=/"
151
152         window.location.href = "{% url 'store' %}"
153     })
154 }
155 </script>
156 {% endblock content %}
```

```
store > templates > store > index.html
1 <!DOCTYPE html>
2 {% load static %}
3 <html lang="en">
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1, minimum-scale=1">
7     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGs03+
8     <link rel="stylesheet" type="text/css" href="{% static 'css/main.css' %}">
9
10     <script type="text/javascript">
11
12         var user = '{{request.user}}'
13
14         function getToken(name) {
15             var cookieValue = null;
16             if (document.cookie && document.cookie != '') {
17                 var cookies = document.cookie.split(';');
18                 for (var i = 0; i < cookies.length; i++) {
19                     var cookie = cookies[i].trim();
20                     // Does this cookie string begin with the name we want?
21                     if (cookie.substring(0, name.length + 1) === (name + '=')) {
22                         cookieValue = decodeURIComponent(cookie.substring(name.length + 1));
23                         break;
24                     }
25                 }
26             }
27             return cookieValue;
28         }
29         var csrftoken = getToken('csrftoken')
30
31         function getCookie(name) {
32             // Split cookie string and get all individual name=value pairs in an array
33             var cookieArr = document.cookie.split(";");
34
35             // Loop through the array elements
```

```
terminal Help index.html - tanamanhias - Visual Studio Code
index.html x checkout.html login.html
store > templates > store > index.html
17 var cookies = document.cookie.split(';');
18 for (var i = 0; i < cookies.length; i++) {
19     var cookie = cookies[i].trim();
20     // Does this cookie string begin with the name we want?
21     if (cookie.substring(0, name.length + 1) === (name + '-')) {
22         cookieValue = decodeURIComponent(cookie.substring(name.length + 1));
23         break;
24     }
25 }
26
27 return cookieValue;
28 }
29 var csrftoken = getToken('csrftoken')
30
31 function getCookie(name) {
32     // Split cookie string and get all individual name=value pairs in an array
33     var cookieArr = document.cookie.split(";");
34
35     // Loop through the array elements
36     for(var i = 0; i < cookieArr.length; i++) {
37         var cookiePair = cookieArr[i].split("=");
38
39         /* Removing whitespace at the beginning of the cookie name
40         and compare it with the given string */
41         if(name == cookiePair[0].trim()) {
42             // Decode the cookie value and return
43             return decodeURIComponent(cookiePair[1]);
44         }
45     }
46
47     // Return null if not found
48     return null;
49 }
50 var cart = JSON.parse(getCookie('cart'))
51
52 if (cart == undefined) {
53     cart = {}
54     console.log('Cart Created!', cart)
```

```
terminal Help index.html - tanamanhias - Visual Studio Code
index.html x checkout.html login.html
store > templates > store > index.html
34
35 // Loop through the array elements
36 for(var i = 0; i < cookieArr.length; i++) {
37     var cookiePair = cookieArr[i].split("=");
38
39     /* Removing whitespace at the beginning of the cookie name
40     and compare it with the given string */
41     if(name == cookiePair[0].trim()) {
42         // Decode the cookie value and return
43         return decodeURIComponent(cookiePair[1]);
44     }
45 }
46
47 // Return null if not found
48 return null;
49 }
50 var cart = JSON.parse(getCookie('cart'))
51
52 if (cart == undefined){
53     cart = {}
54     console.log('Cart Created!', cart)
55     document.cookie = 'cart=' + JSON.stringify(cart) + ";domain=;path=/"
56 }
57 console.log('Cart:', cart)
58
59 </script>
60
61 <title>HOME</title>
62 </head>
63 <body>
64
65
66
67 <nav class="navbar navbar-expand-lg navbar-dark bg-success">
68     <a class="navbar-brand" href="#">
69         <img style="height: 50px; width: 50px; border-radius: 50%; alt="">
70     </a>
```