LAPORAN PRAKTIKUM PEMOGRAMAN PERANGKAT BERGERAK

(APLIKASI LAUNDRY)

Diajukan untuk memenuhi salah satu persyaratan dalam Mata Kuliah
Pemograman Perangkat Bergerak

Disusun Oleh :

WINDY AFRILYA (4311801001)

FARAH DIBA HARAHAP (4311801004)

TASYA WULANDARI (4311801007)

JURUSAN TEKNIK INFORMATIKA
PROGRAM STUDI MULTIMEDIA DAN JARINGAN
POLITEKNIK NEGERI BATAM
April, 2020

PENDAHULUAN

A. Latar Belakang Aplikasi Mobile

Aplikasi *Mobile* merupakan sebuah Program yang dikembangkan agar dapat digunakan pada perangkat bergerak seperti *Phone/Tablet* ataupun Jam tangan.

Android digunakan lebih dari 80% dari semua *smartphone* Lebih dari 2 Juta aplikasi Android di *Google Play store*.

Aplikasi *Mobile* pada saat ini merupakan teknologi yang snagat berkembang pesat. Perkembangan aplikasi *Mobile* yang pesat berdampak pada gaya hidup masyarakat sehati-hari. Aplikasi *Mobile* saat ini banyak digunakan untuk membantu aktivitas pada kehidupan sehari-hari. Keunggulan dari aplikasi *mobile* adalah sifatnya yang mudah dan dapat diguanakn dimana saja sehingga aplikasi ini sangat cocok untuk membantu aktivitas yang memiliki mobilitas Tinggi.

Laundry merupakan sebuah Usaha yang berada dikalangan masyarakat yang saat ini sangat bersatu dengan kehidupan masyarakat. Keberadaan *Laundry* dikalangan masyarakat sangat dibutuhkan terutama bagi masyarakat yang keseharian nya sangat *produktif* dalam pekerjaan atau pun kegiatan luar lainnya.

Dikarenakan Pesatnya perkembangan Aplikasi *Mobile* dikalangan Masyarakat disini kami membuat sebuah aplikasi *Mobile* yang dapat digunakan oleh pengusaha *Laundry* yang dapat membantu dalam Proses perhitungan jumlah *laundry* hingga pembayarannya.

B. Tujuan Pembuatan Aplikasi Laundry

Secara umum Tujuan kami membuat Aplikasi ini yaitu untuk memenuhi sebuah syarat dalam Mata Kuliah Pemograman Perangkat Bergerak, selain itu Tujuan kami yaitu untuk mempermudah dalam proses *Lundry* Pakaian dikalangan masyarakat, yang dimana seperti yang kita tahu saat ini *Laundry* masih banyak yang menggunakan kertas untuk proses Pembayaran nya, jadi kami membuat Aplikasi Mobile Laundry yang dapat diguanakan untuk Penghitungan dan pembayaran agar mempermudah dan mempercepat pekerjaan.

C. Batasan Masalah

Adapun batasan masalah yang penulis buat berguna sebagai pedoman agar tidak meluas dari pembahasan yang penulis lakukan, adapun batasan tersebut :

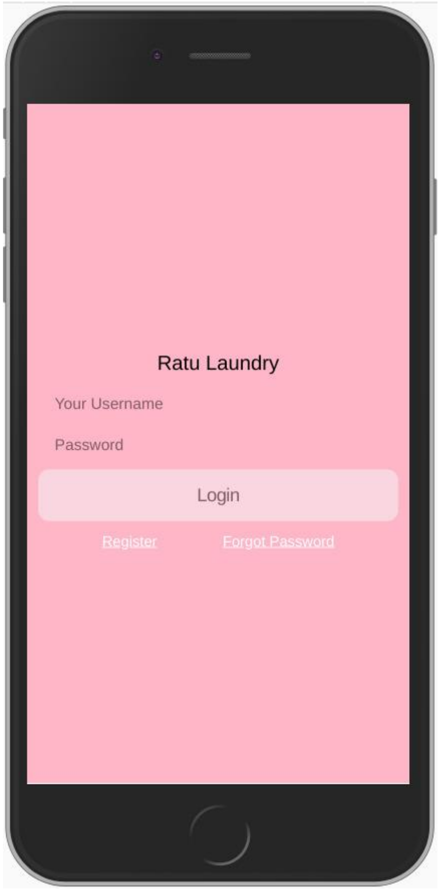
1. Kurang nya Pemahaman dalam menggunakan Visual Studio Code.
2. Alat Kerja yang kurang memadai.

C. Pembahasan

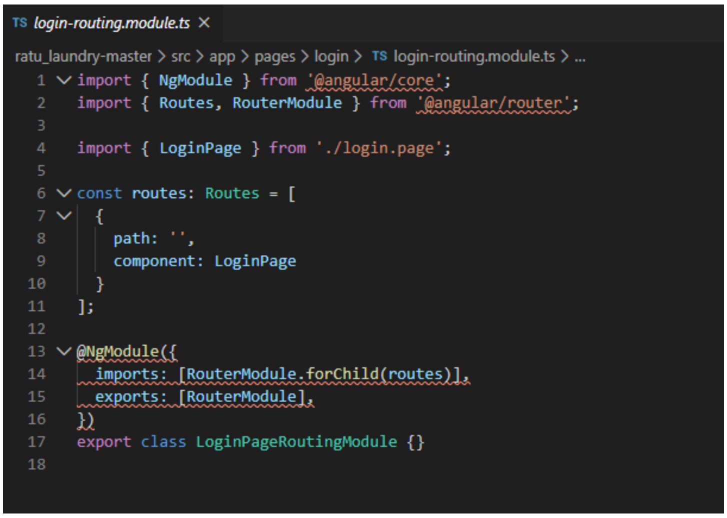
Aplikasi *Laundry* merupakan sebuah Aplikasi *Mobile* yang dimana berisikan mengenai pendataan jumlah kain yang akan dilaundry hingga proes pembayarannya, yang dimana aplikasi *Laundry* dapat digunakan di *Smartphone* yang di buat sederhana agar mudah dipahami dan juga mudah digunakan.

1. Login Page

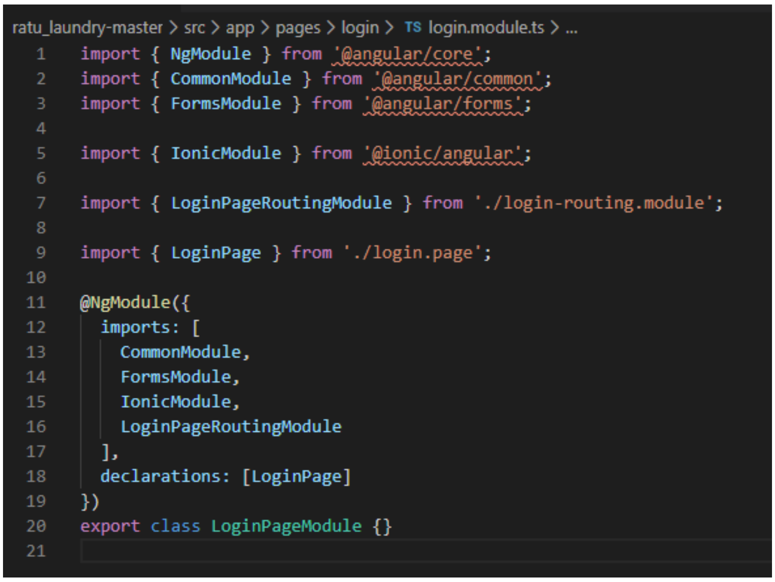
Login Page disini merupakan halaman untuk masuk kedalam aplikasi Ratu *Laundry*



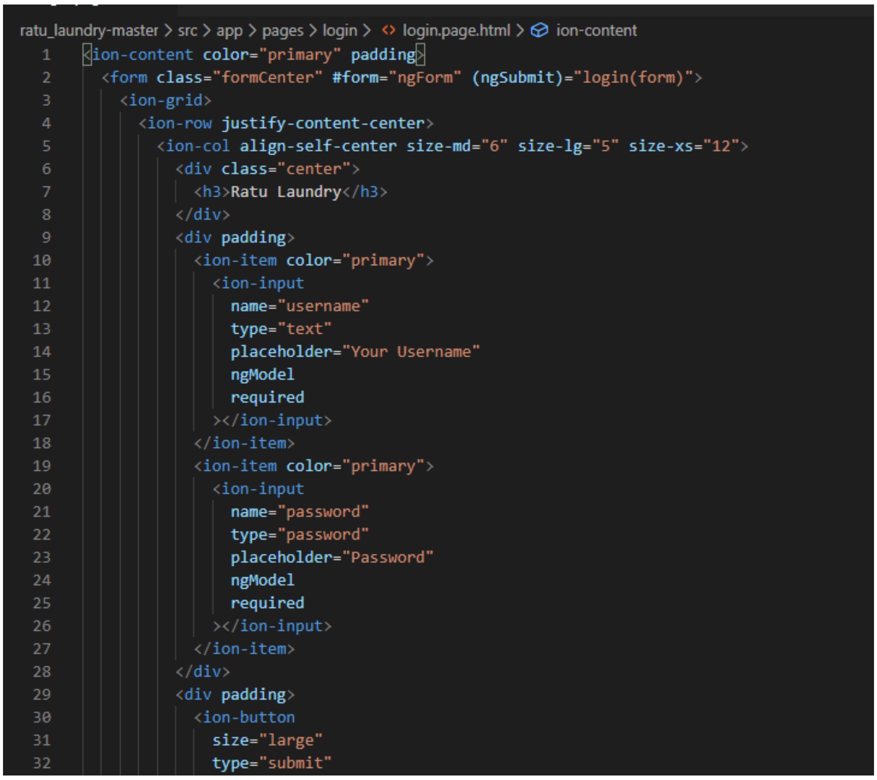
Login Page


TS login-routing.module.ts ×
ratu_laundry-master > src > app > pages > login > TS login-routing.module.ts > ...

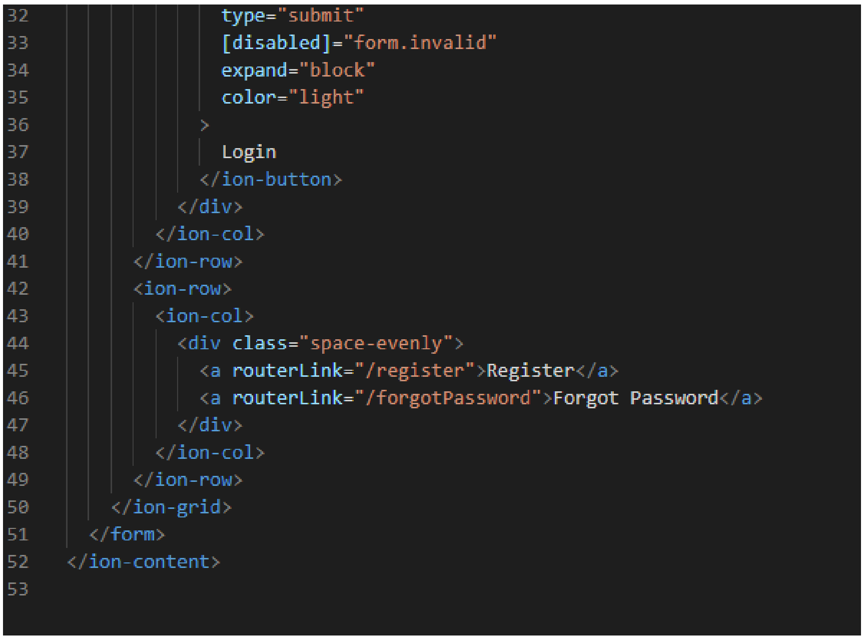
```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { LoginPage } from './login.page';

const routes: Routes = [
  {
    path: '',
    component: LoginPage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class LoginPageRoutingModule {}
```


ratu_laundry-master > src > app > pages > login > TS login.module.ts > ...

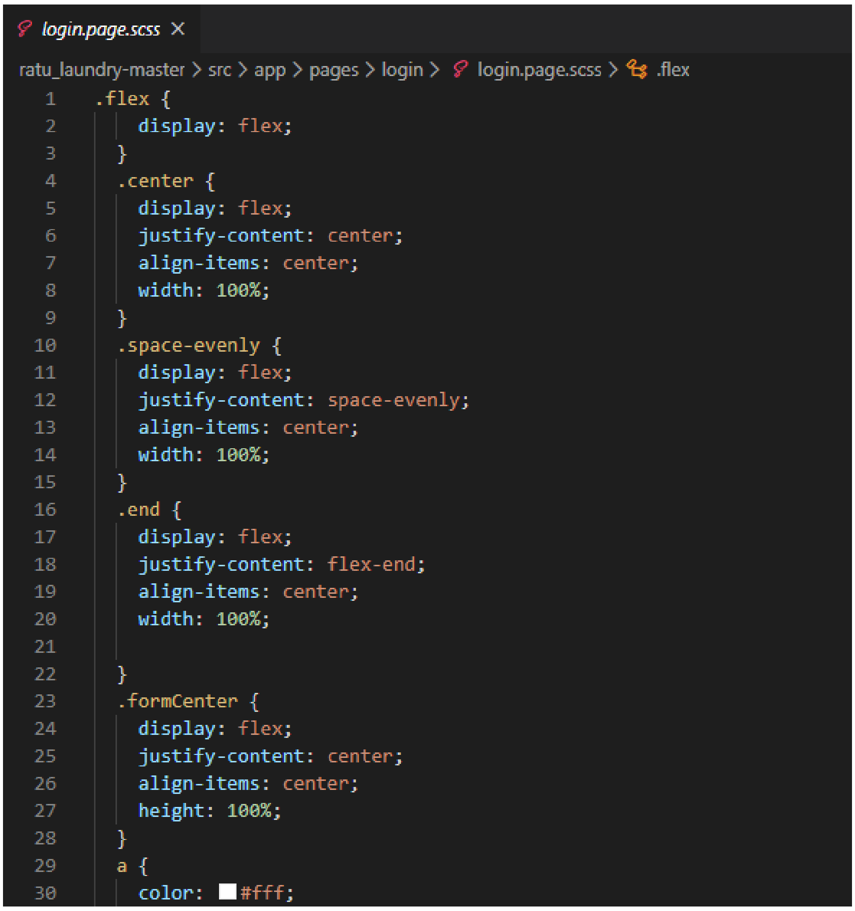
```typescript
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { LoginPageRoutingModule } from './login-routing.module';

import { LoginPage } from './login.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    LoginPageRoutingModule
  ],
  declarations: [LoginPage]
})
export class LoginPageModule {}
```


ratu_laundry-master > src > app > pages > login > <> login.page.html > ion-content

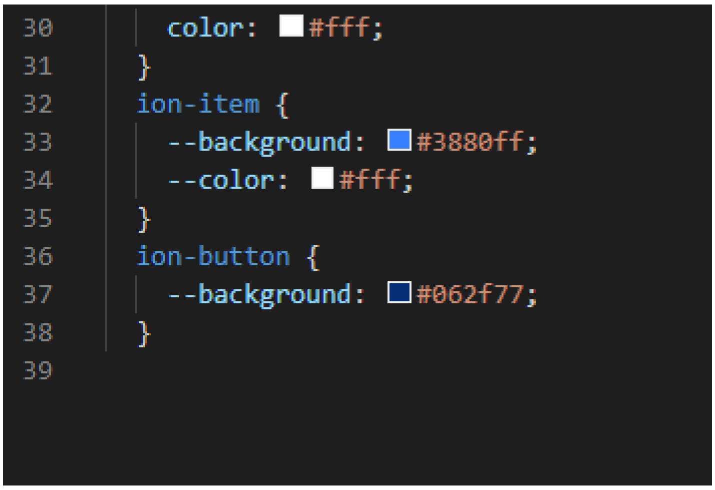
```html
<ion-content color="primary" padding>
  <form class="formCenter" #form="ngForm" (ngSubmit)="login(form)">
    <ion-grid>
      <ion-row justify-content-center>
        <ion-col align-self-center size-md="6" size-lg="5" size-xs="12">
          <div class="center">
            <h3>Ratu Laundry</h3>
          </div>
          <div padding>
            <ion-item color="primary">
              <ion-input
                name="username"
                type="text"
                placeholder="Your Username"
                ngModel
                required
              ></ion-input>
            </ion-item>
            <ion-item color="primary">
              <ion-input
                name="password"
                type="password"
                placeholder="Password"
                ngModel
                required
              ></ion-input>
            </ion-item>
          </div>
          <div padding>
            <ion-button
              size="large"
              type="submit"
```



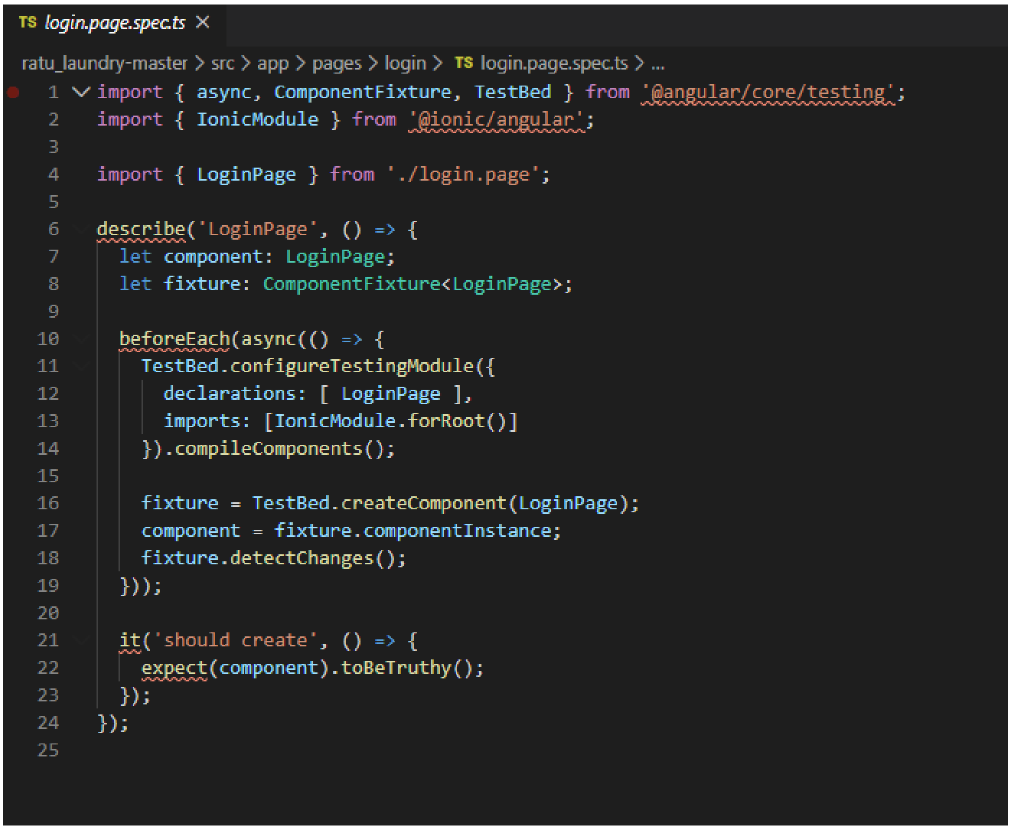
```html
              type="submit"
              [disabled]="form.invalid"
              expand="block"
              color="light"
            >
              Login
            </ion-button>
          </div>
        </ion-col>
      </ion-row>
      <ion-row>
        <ion-col>
          <div class="space-evenly">
            <a routerLink="/register">Register</a>
            <a routerLink="/forgotPassword">Forgot Password</a>
          </div>
        </ion-col>
      </ion-row>
    </ion-grid>
  </form>
</ion-content>

```


login.page.scss ×
ratu_laundry-master > src > app > pages > login > login.page.scss > .flex

```scss
.flex {
    display: flex;
  }
  .center {
    display: flex;
    justify-content: center;
    align-items: center;
    width: 100%;
  }
  .space-evenly {
    display: flex;
    justify-content: space-evenly;
    align-items: center;
    width: 100%;
  }
  .end {
    display: flex;
    justify-content: flex-end;
    align-items: center;
    width: 100%;

  }
  .formCenter {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100%;
  }
  a {
    color: #fff;
```

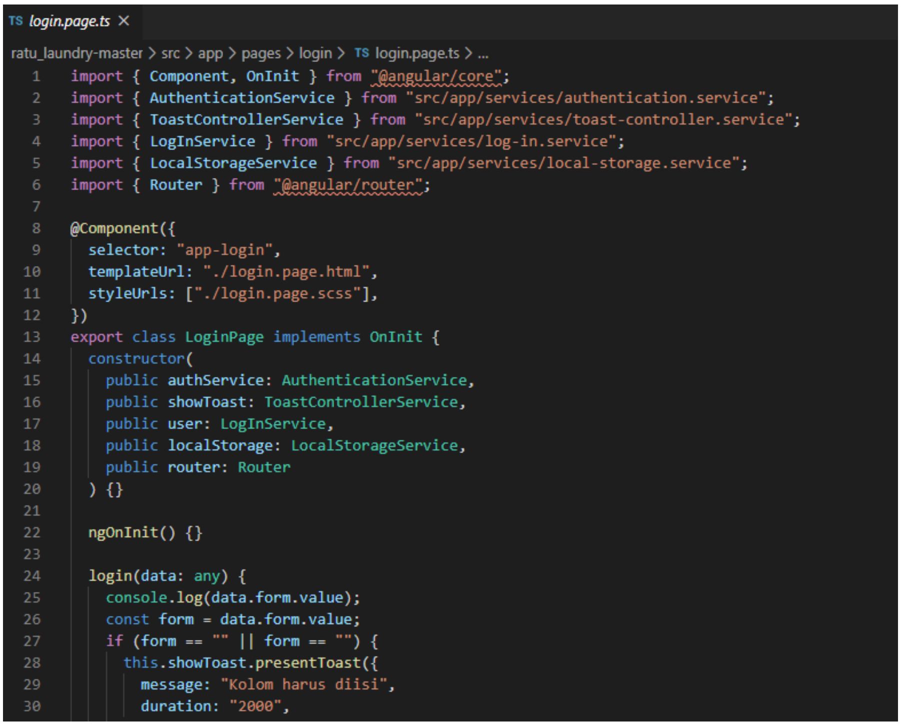


```
    color: #fff;
  }
  ion-item {
    --background: #3880ff;
    --color: #fff;
  }
  ion-button {
    --background: #062f77;
  }

```

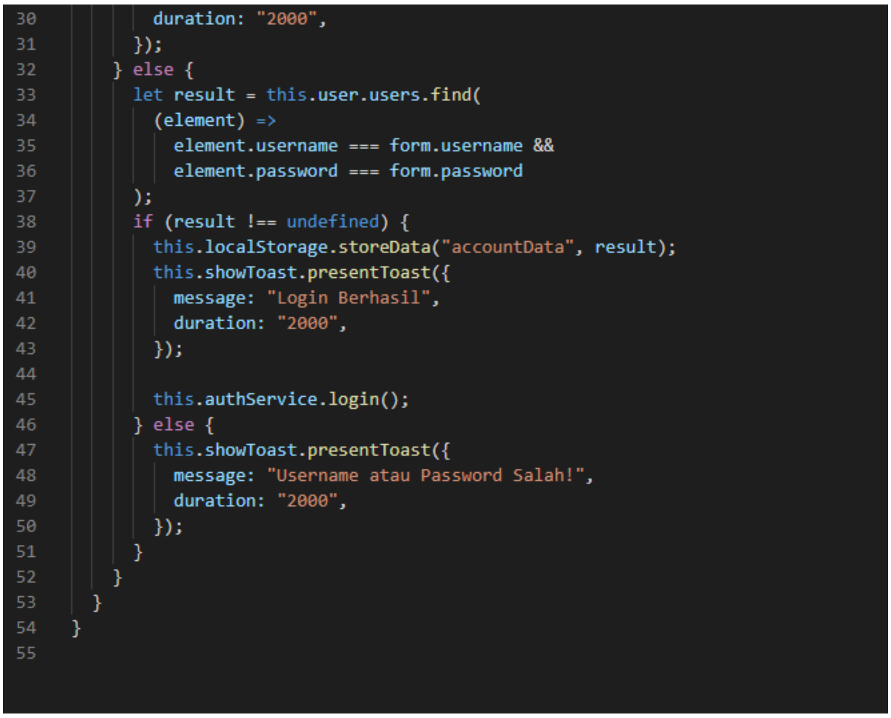


TS login.page.spec.ts ×
ratu_laundry-master > src > app > pages > login > TS login.page.spec.ts > ...

```
import { async, ComponentFixture, TestBed } from '@angular/core/testing';
import { IonicModule } from '@ionic/angular';

import { LoginPage } from './login.page';

describe('LoginPage', () => {
  let component: LoginPage;
  let fixture: ComponentFixture<LoginPage>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ LoginPage ],
      imports: [IonicModule.forRoot()]
    }).compileComponents();

    fixture = TestBed.createComponent(LoginPage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  }));

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```


TS login.page.ts
ratu_laundry-master > src > app > pages > login > TS login.page.ts > ...

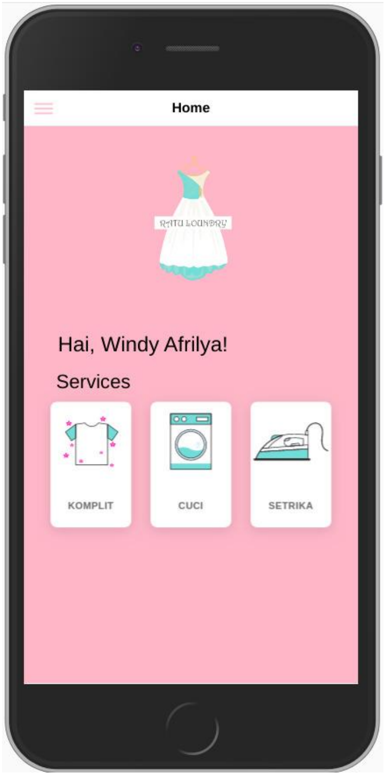
```typescript
import { Component, OnInit } from "@angular/core";
import { AuthenticationService } from "src/app/services/authentication.service";
import { ToastControllerService } from "src/app/services/toast-controller.service";
import { LogInService } from "src/app/services/log-in.service";
import { LocalStorageService } from "src/app/services/local-storage.service";
import { Router } from "@angular/router";

@Component({
  selector: "app-login",
  templateUrl: "./login.page.html",
  styleUrls: ["./login.page.scss"],
})
export class LoginPage implements OnInit {
  constructor(
    public authService: AuthenticationService,
    public showToast: ToastControllerService,
    public user: LogInService,
    public localStorage: LocalStorageService,
    public router: Router
  ) {}

  ngOnInit() {}

  login(data: any) {
    console.log(data.form.value);
    const form = data.form.value;
    if (form == "" || form == "") {
      this.showToast.presentToast({
        message: "Kolom harus diisi",
        duration: "2000",
```



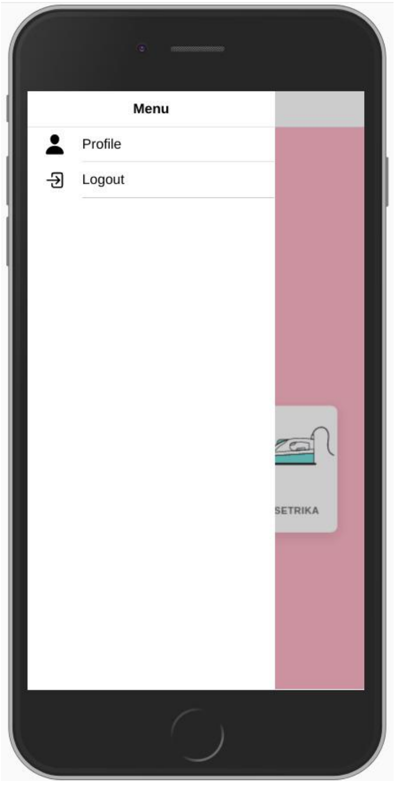
```typescript
        duration: "2000",
      });
    } else {
      let result = this.user.users.find(
        (element) =>
          element.username === form.username &&
          element.password === form.password
      );
      if (result !== undefined) {
        this.localStorage.storeData("accountData", result);
        this.showToast.presentToast({
          message: "Login Berhasil",
          duration: "2000",
        });

        this.authService.login();
      } else {
        this.showToast.presentToast({
          message: "Username atau Password Salah!",
          duration: "2000",
        });
      }
    }
  }
}
```

2. Home Page

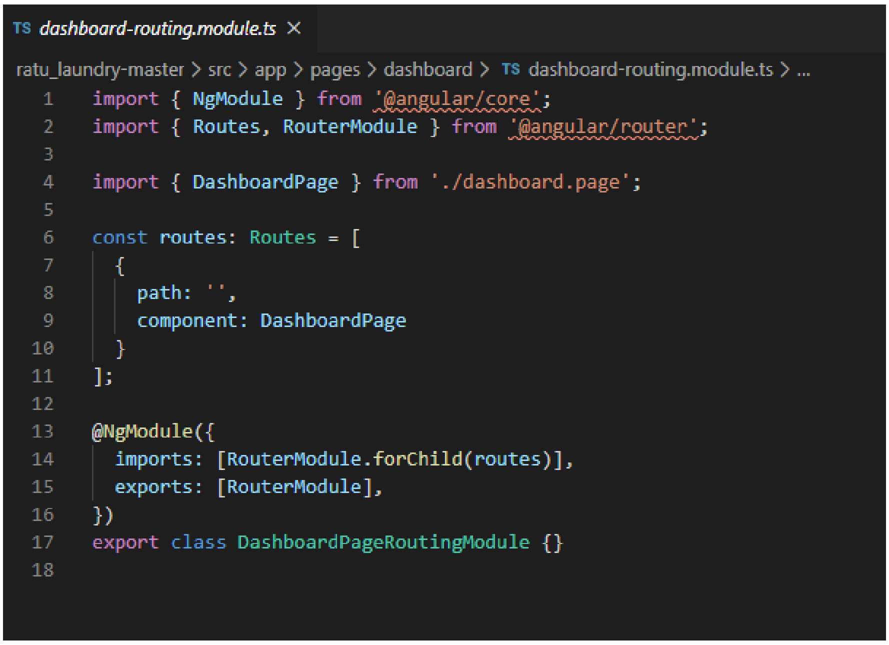
Home Page disini merupakan halaman awal dalam aplikasi Ratu *Laundry*



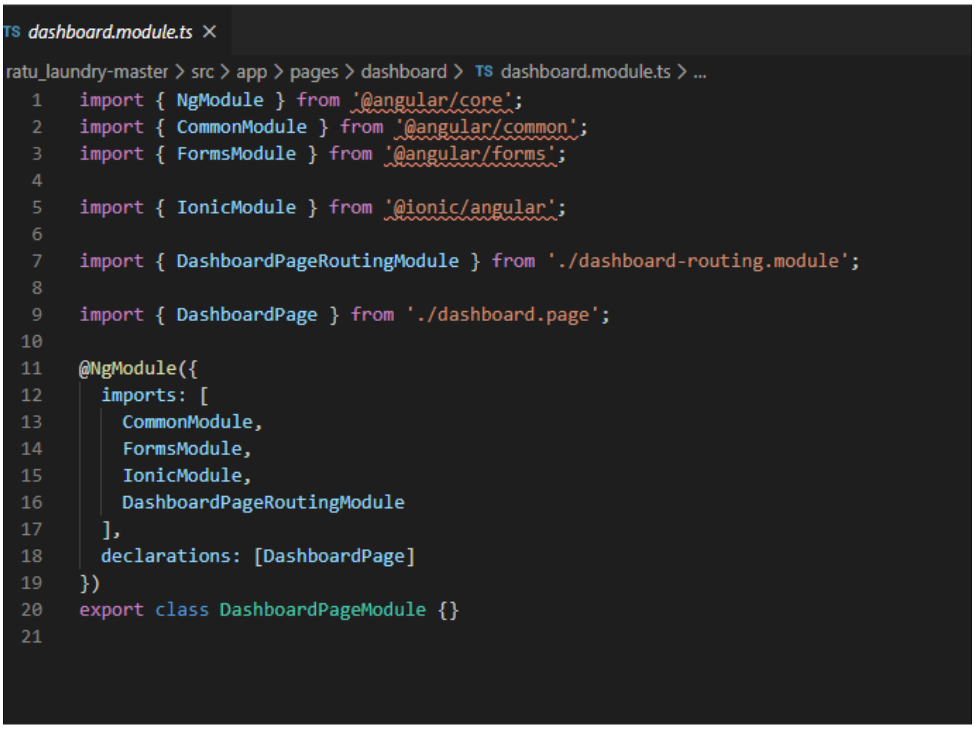
Hai, Windy Afrilya!



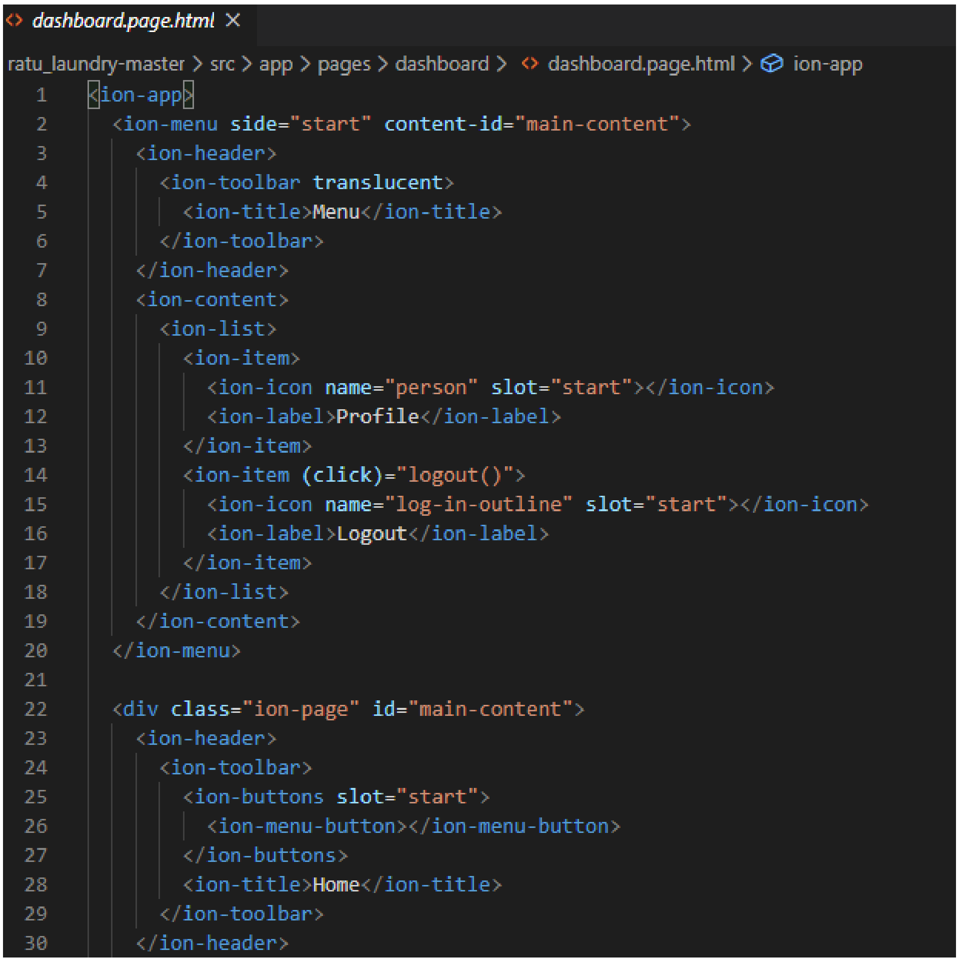
Home Page


TS dashboard-routing.module.ts ×
ratu_laundry-master > src > app > pages > dashboard > TS dashboard-routing.module.ts > ...

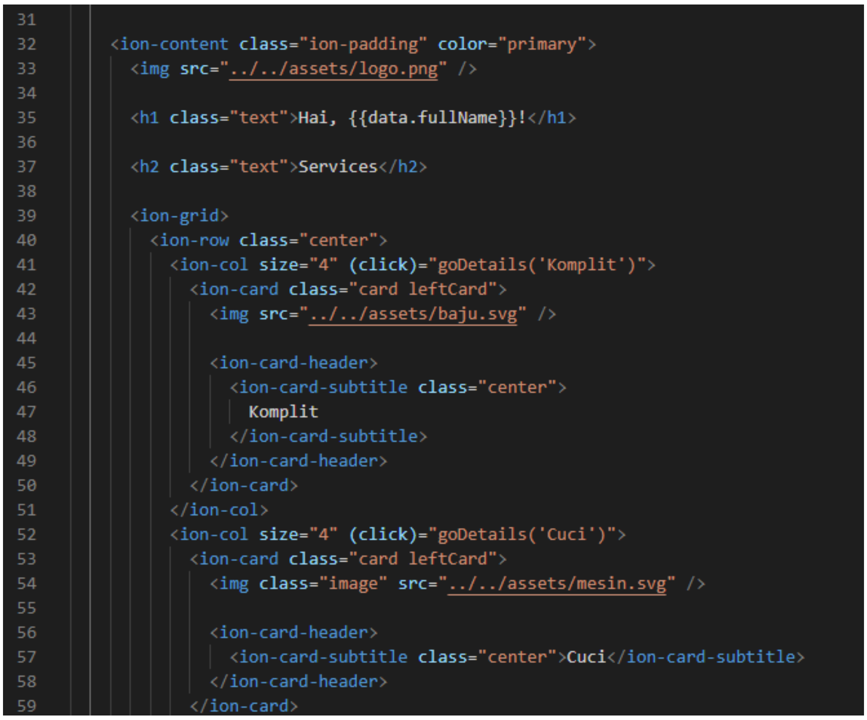
```typescript
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { DashboardPage } from './dashboard.page';

const routes: Routes = [
  {
    path: '',
    component: DashboardPage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class DashboardPageRoutingModule {}

```


TS dashboard.module.ts ×
ratu_laundry-master > src > app > pages > dashboard > TS dashboard.module.ts > ...

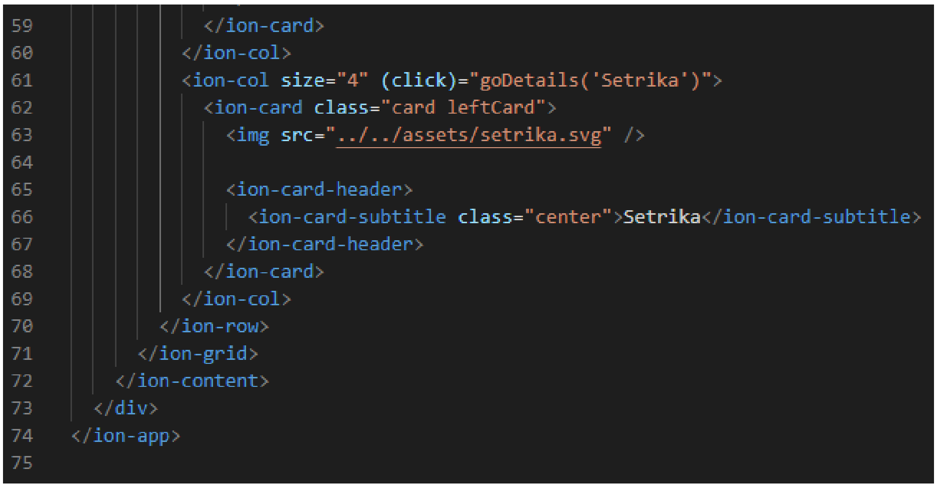
```typescript
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { DashboardPageRoutingModule } from './dashboard-routing.module';

import { DashboardPage } from './dashboard.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    DashboardPageRoutingModule
  ],
  declarations: [DashboardPage]
})
export class DashboardPageModule {}

```


dashboard.page.html
ratu_laundry-master > src > app > pages > dashboard > dashboard.page.html > ion-app

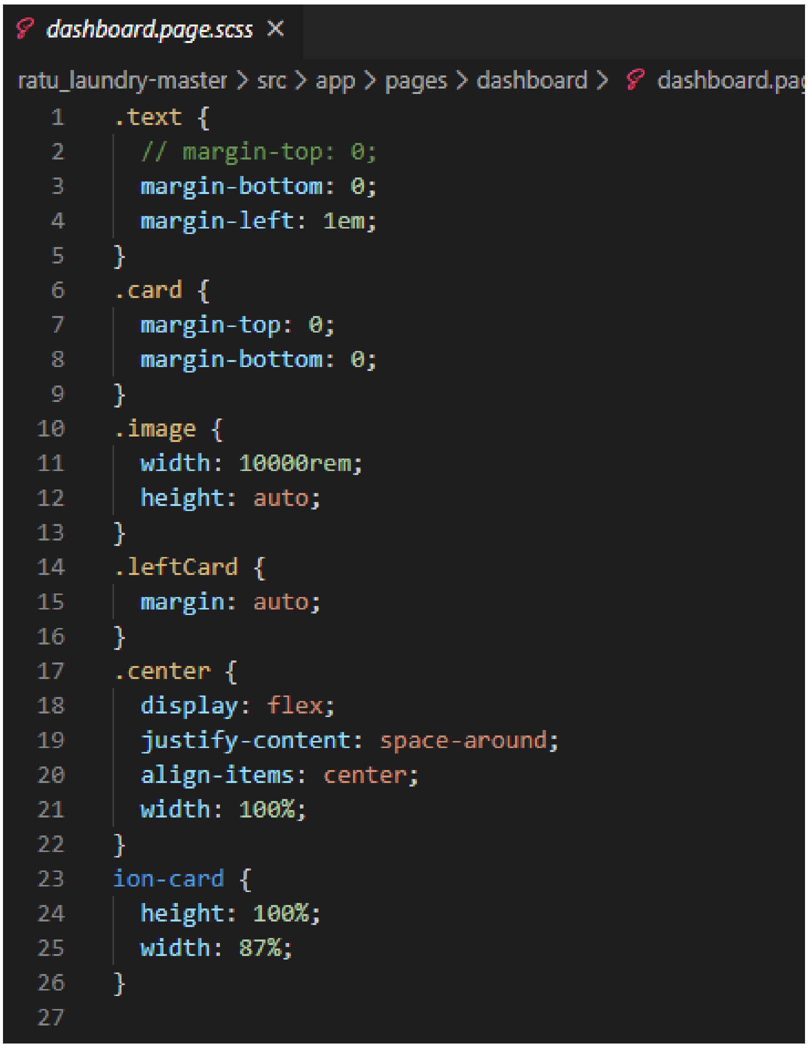
```html
<ion-app>
  <ion-menu side="start" content-id="main-content">
    <ion-header>
      <ion-toolbar translucent>
        <ion-title>Menu</ion-title>
      </ion-toolbar>
    </ion-header>
    <ion-content>
      <ion-list>
        <ion-item>
          <ion-icon name="person" slot="start"></ion-icon>
          <ion-label>Profile</ion-label>
        </ion-item>
        <ion-item (click)="logout()">
          <ion-icon name="log-in-outline" slot="start"></ion-icon>
          <ion-label>Logout</ion-label>
        </ion-item>
      </ion-list>
    </ion-content>
  </ion-menu>

  <div class="ion-page" id="main-content">
    <ion-header>
      <ion-toolbar>
        <ion-buttons slot="start">
          <ion-menu-button></ion-menu-button>
        </ion-buttons>
        <ion-title>Home</ion-title>
      </ion-toolbar>
    </ion-header>
```



```html

    <ion-content class="ion-padding" color="primary">
      <img src="../../assets/logo.png" />

      <h1 class="text">Hai, {{data.fullName}}!</h1>

      <h2 class="text">Services</h2>

      <ion-grid>
        <ion-row class="center">
          <ion-col size="4" (click)="goDetails('Komplit')">
            <ion-card class="card leftCard">
              <img src="../../assets/baju.svg" />

              <ion-card-header>
                <ion-card-subtitle class="center">
                  Komplit
                </ion-card-subtitle>
              </ion-card-header>
            </ion-card>
          </ion-col>
          <ion-col size="4" (click)="goDetails('Cuci')">
            <ion-card class="card leftCard">
              <img class="image" src="../../assets/mesin.svg" />

              <ion-card-header>
                <ion-card-subtitle class="center">Cuci</ion-card-subtitle>
              </ion-card-header>
            </ion-card>
```

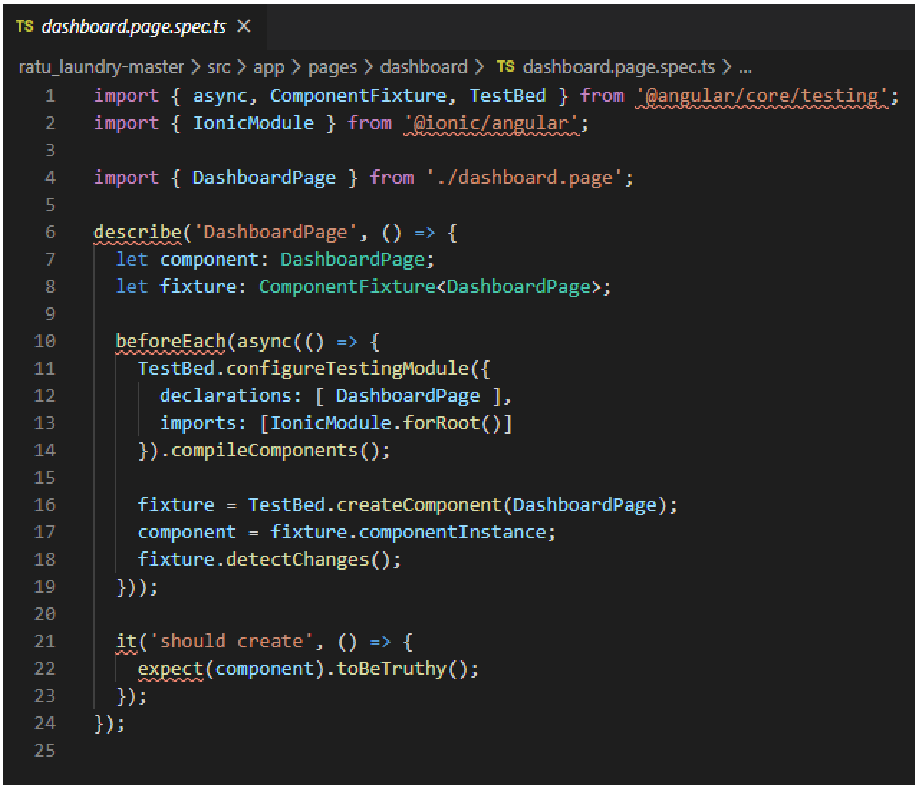


```
            </ion-card>
          </ion-col>
          <ion-col size="4" (click)="goDetails('Setrika')">
            <ion-card class="card leftCard">
              <img src="../../assets/setrika.svg" />

              <ion-card-header>
                <ion-card-subtitle class="center">Setrika</ion-card-subtitle>
              </ion-card-header>
            </ion-card>
          </ion-col>
        </ion-row>
      </ion-grid>
    </ion-content>
  </div>
</ion-app>

```



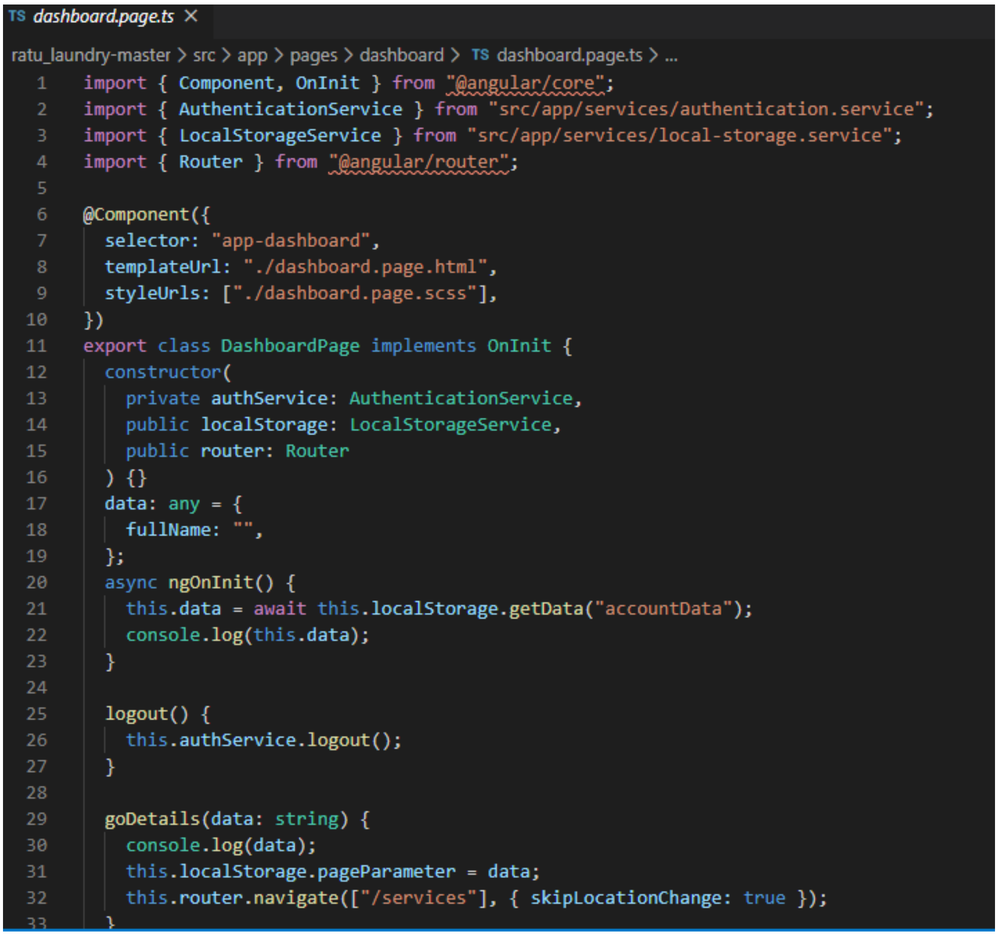
dashboard.page.scss ×
ratu_laundry-master > src > app > pages > dashboard > dashboard.pag

```
.text {
  // margin-top: 0;
  margin-bottom: 0;
  margin-left: 1em;
}
.card {
  margin-top: 0;
  margin-bottom: 0;
}
.image {
  width: 10000rem;
  height: auto;
}
.leftCard {
  margin: auto;
}
.center {
  display: flex;
  justify-content: space-around;
  align-items: center;
  width: 100%;
}
ion-card {
  height: 100%;
  width: 87%;
}

```



TS dashboard.page.spec.ts ×
ratu_laundry-master > src > app > pages > dashboard > TS dashboard.page.spec.ts > ...

```typescript
import { async, ComponentFixture, TestBed } from '@angular/core/testing';
import { IonicModule } from '@ionic/angular';

import { DashboardPage } from './dashboard.page';

describe('DashboardPage', () => {
  let component: DashboardPage;
  let fixture: ComponentFixture<DashboardPage>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ DashboardPage ],
      imports: [IonicModule.forRoot()]
    }).compileComponents();

    fixture = TestBed.createComponent(DashboardPage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  }));

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

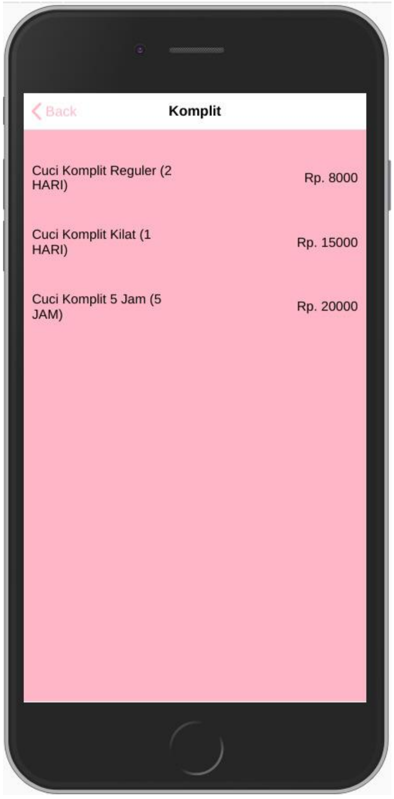


TS dashboard.page.ts ×
ratu_laundry-master > src > app > pages > dashboard > TS dashboard.page.ts > ...

```typescript
import { Component, OnInit } from "@angular/core";
import { AuthenticationService } from "src/app/services/authentication.service";
import { LocalStorageService } from "src/app/services/local-storage.service";
import { Router } from "@angular/router";

@Component({
  selector: "app-dashboard",
  templateUrl: "./dashboard.page.html",
  styleUrls: ["./dashboard.page.scss"],
})
export class DashboardPage implements OnInit {
  constructor(
    private authService: AuthenticationService,
    public localStorage: LocalStorageService,
    public router: Router
  ) {}
  data: any = {
    fullName: "",
  };
  async ngOnInit() {
    this.data = await this.localStorage.getData("accountData");
    console.log(this.data);
  }

  logout() {
    this.authService.logout();
  }

  goDetails(data: string) {
    console.log(data);
    this.localStorage.pageParameter = data;
    this.router.navigate(["/services"], { skipLocationChange: true });
  }
```

3. Halaman Kategori Paket

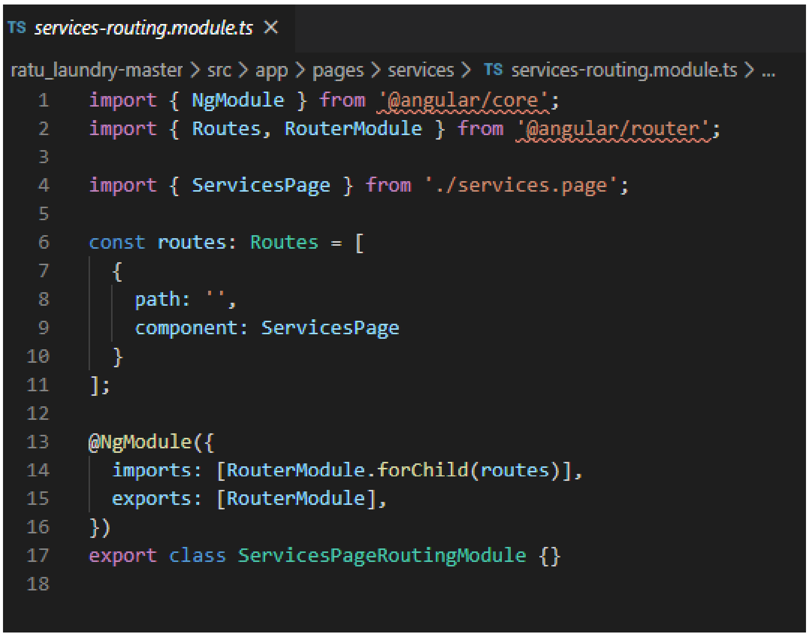
Halaman kategori paket disini berisi paket apa yang akan dipilih untuk proses *Laundry* nya.



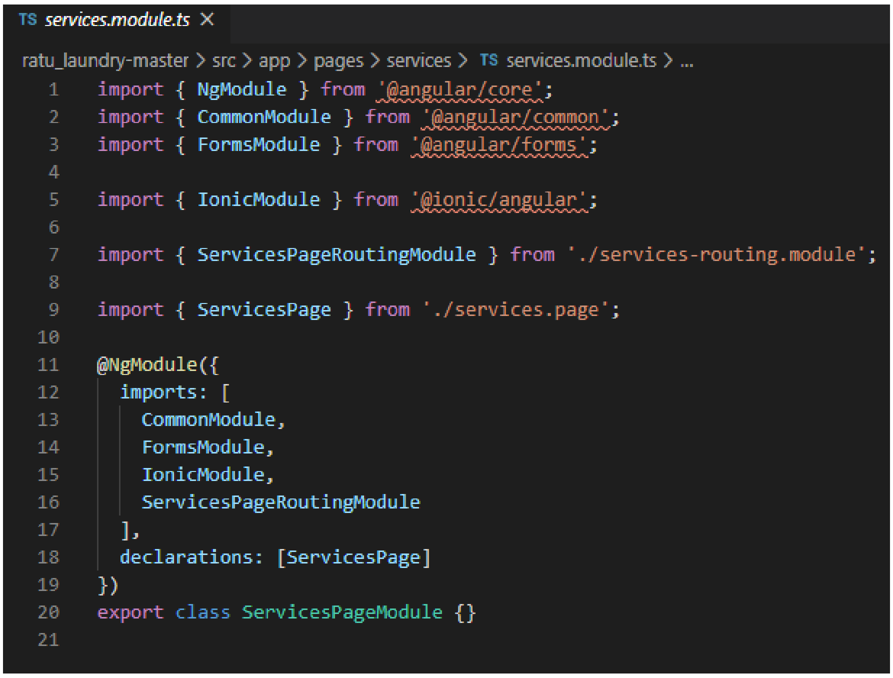
Cuci Komplit Reguler (2 HARI) Rp. 8000

Cuci Komplit Kilat (1 HARI) Rp. 15000

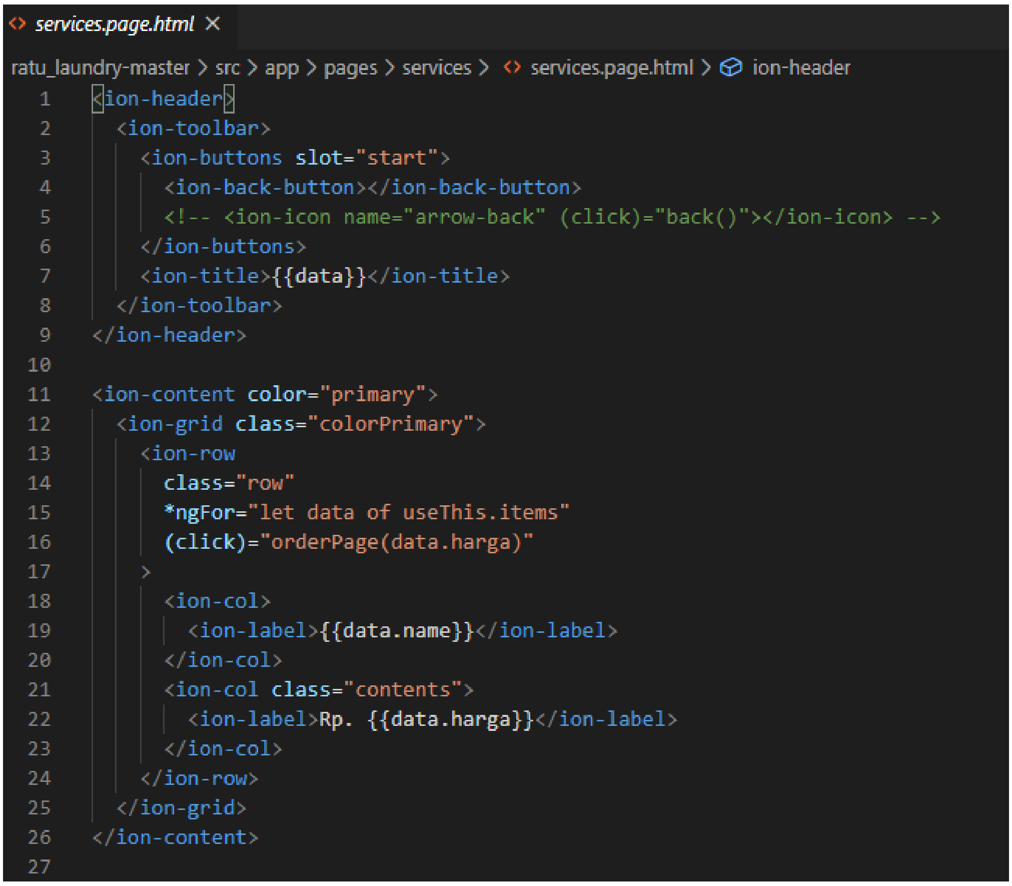
Cuci Komplit 5 Jam (5 JAM) Rp. 20000


TS services-routing.module.ts ×
ratu_laundry-master > src > app > pages > services > TS services-routing.module.ts > ...

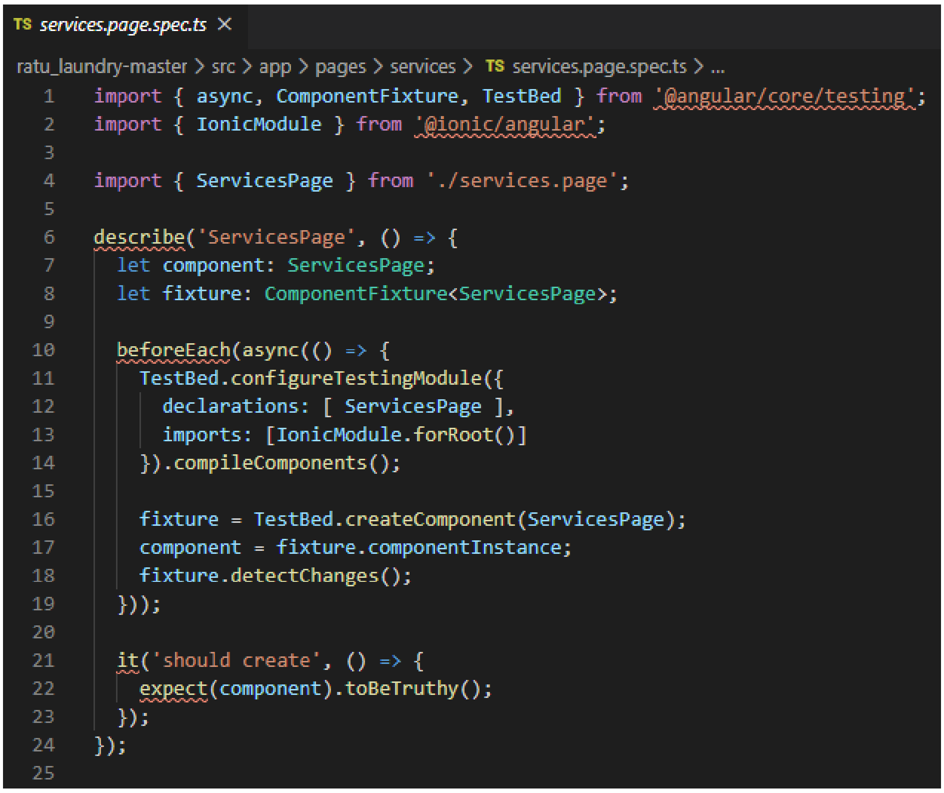
```typescript
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { ServicesPage } from './services.page';

const routes: Routes = [
  {
    path: '',
    component: ServicesPage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class ServicesPageRoutingModule {}
```


services.module.ts
ratu_laundry-master > src > app > pages > services > services.module.ts > ...

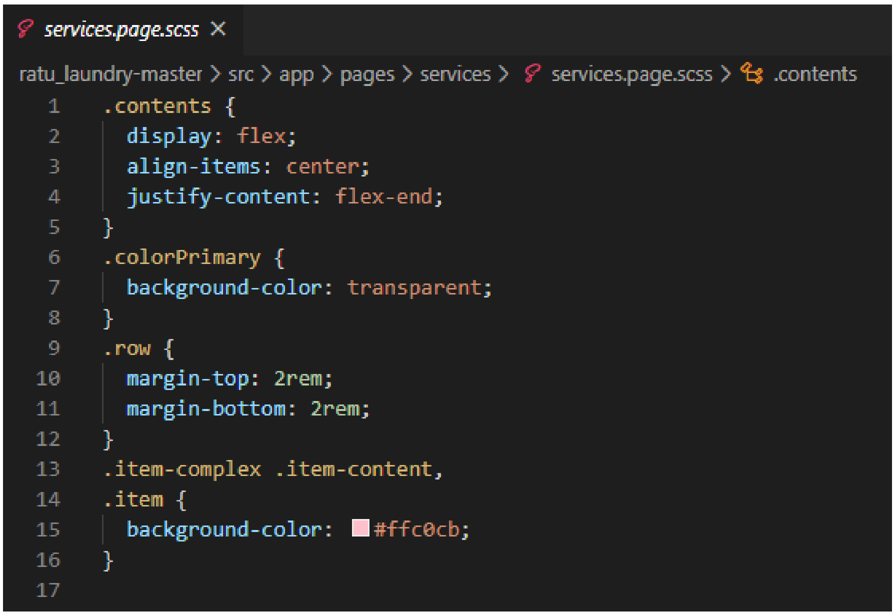
```typescript
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { ServicesPageRoutingModule } from './services-routing.module';

import { ServicesPage } from './services.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    ServicesPageRoutingModule
  ],
  declarations: [ServicesPage]
})
export class ServicesPageModule {}

```


services.page.html
ratu_laundry-master > src > app > pages > services > services.page.html > ion-header

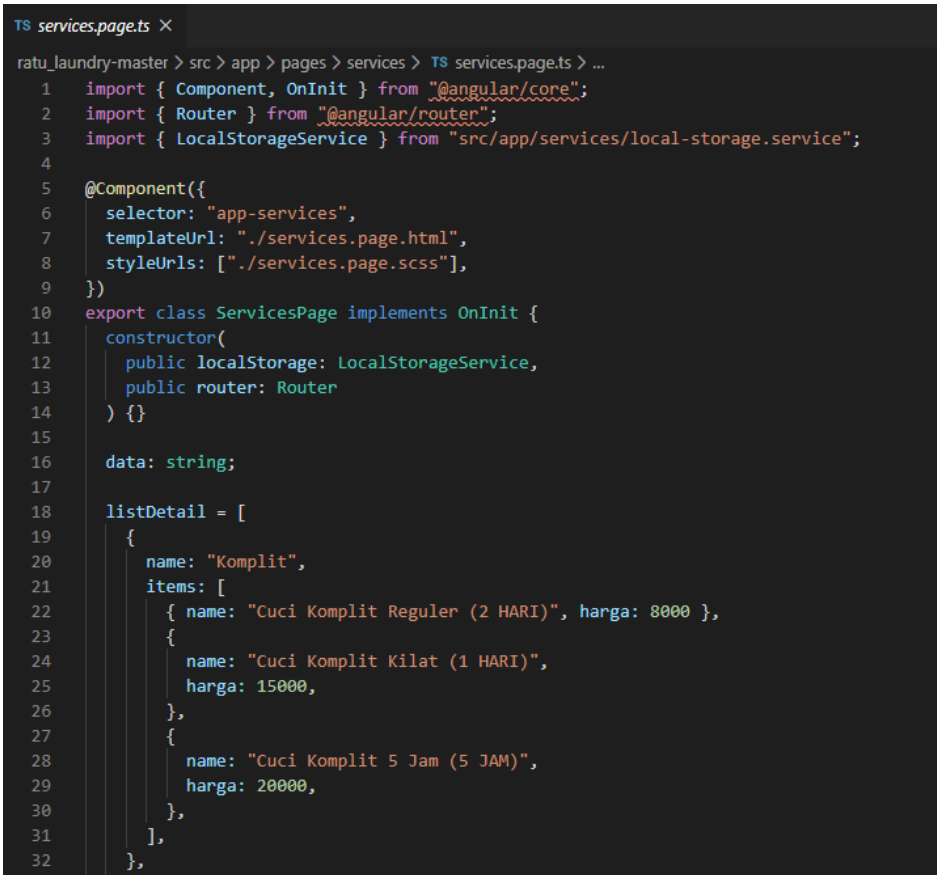
```html
<ion-header>
  <ion-toolbar>
    <ion-buttons slot="start">
      <ion-back-button></ion-back-button>
      <!-- <ion-icon name="arrow-back" (click)="back()"></ion-icon> -->
    </ion-buttons>
    <ion-title>{{data}}</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content color="primary">
  <ion-grid class="colorPrimary">
    <ion-row
      class="row"
      *ngFor="let data of useThis.items"
      (click)="orderPage(data.harga)"
    >
      <ion-col>
        <ion-label>{{data.name}}</ion-label>
      </ion-col>
      <ion-col class="contents">
        <ion-label>Rp. {{data.harga}}</ion-label>
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-content>

```


TS services.page.spec.ts ×
ratu_laundry-master > src > app > pages > services > TS services.page.spec.ts > ...

```typescript
import { async, ComponentFixture, TestBed } from '@angular/core/testing';
import { IonicModule } from '@ionic/angular';

import { ServicesPage } from './services.page';

describe('ServicesPage', () => {
  let component: ServicesPage;
  let fixture: ComponentFixture<ServicesPage>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ ServicesPage ],
      imports: [IonicModule.forRoot()]
    }).compileComponents();

    fixture = TestBed.createComponent(ServicesPage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  }));

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```


services.page.scss ×
ratu_laundry-master > src > app > pages > services > services.page.scss > .contents

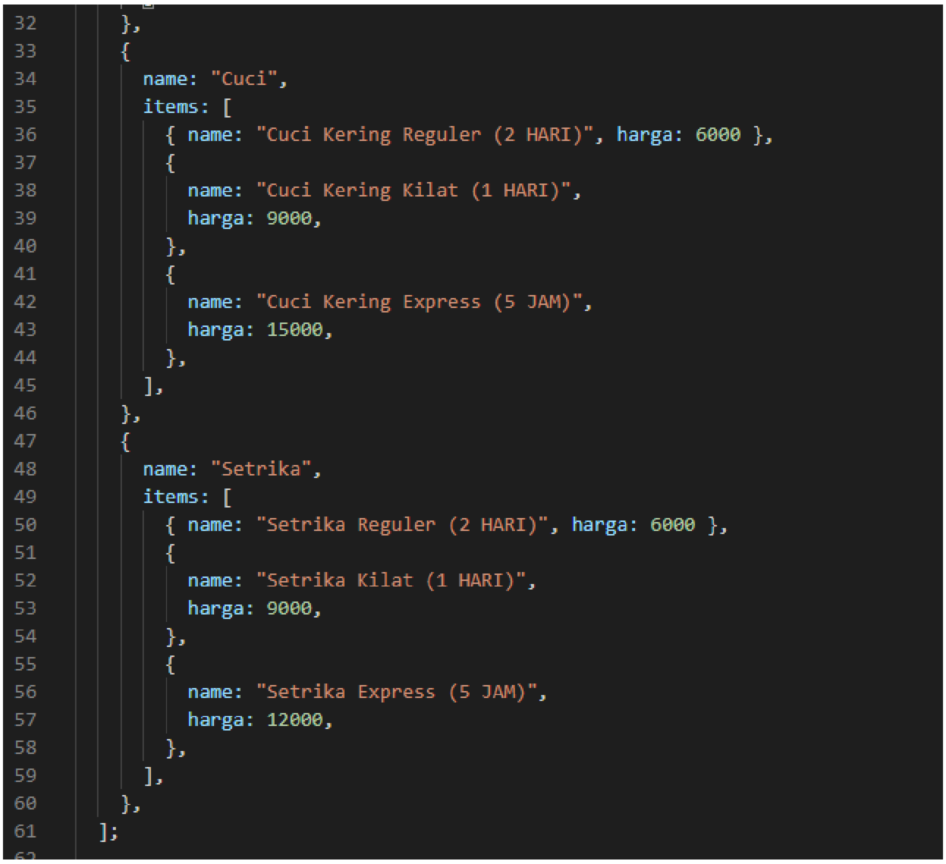
```scss
.contents {
  display: flex;
  align-items: center;
  justify-content: flex-end;
}
.colorPrimary {
  background-color: transparent;
}
.row {
  margin-top: 2rem;
  margin-bottom: 2rem;
}
.item-complex .item-content,
.item {
  background-color: #ffc0cb;
}
```



services.page.ts ×

ratu_laundry-master > src > app > pages > services > TS services.page.ts > ...

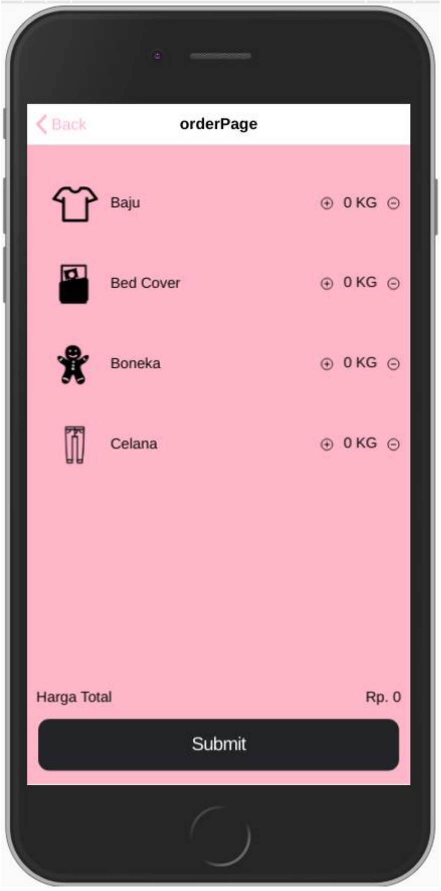
```typescript
import { Component, OnInit } from "@angular/core";
import { Router } from "@angular/router";
import { LocalStorageService } from "src/app/services/local-storage.service";

@Component({
  selector: "app-services",
  templateUrl: "./services.page.html",
  styleUrls: ["./services.page.scss"],
})
export class ServicesPage implements OnInit {
  constructor(
    public localStorage: LocalStorageService,
    public router: Router
  ) {}

  data: string;

  listDetail = [
    {
      name: "Komplit",
      items: [
        { name: "Cuci Komplit Reguler (2 HARI)", harga: 8000 },
        {
          name: "Cuci Komplit Kilat (1 HARI)",
          harga: 15000,
        },
        {
          name: "Cuci Komplit 5 Jam (5 JAM)",
          harga: 20000,
        },
      ],
    },
```

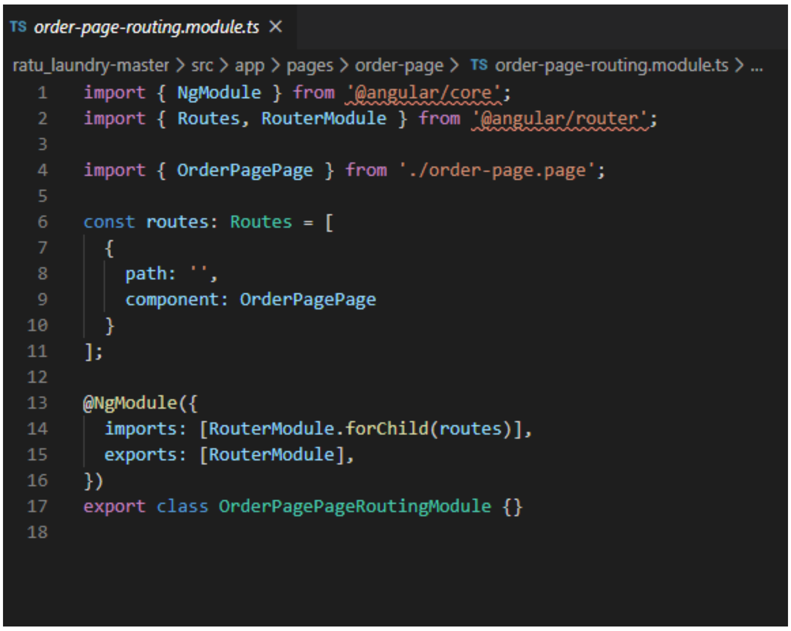


```typescript
    },
    {
      name: "Cuci",
      items: [
        { name: "Cuci Kering Reguler (2 HARI)", harga: 6000 },
        {
          name: "Cuci Kering Kilat (1 HARI)",
          harga: 9000,
        },
        {
          name: "Cuci Kering Express (5 JAM)",
          harga: 15000,
        },
      ],
    },
    {
      name: "Setrika",
      items: [
        { name: "Setrika Reguler (2 HARI)", harga: 6000 },
        {
          name: "Setrika Kilat (1 HARI)",
          harga: 9000,
        },
        {
          name: "Setrika Express (5 JAM)",
          harga: 12000,
        },
      ],
    },
  ];
```

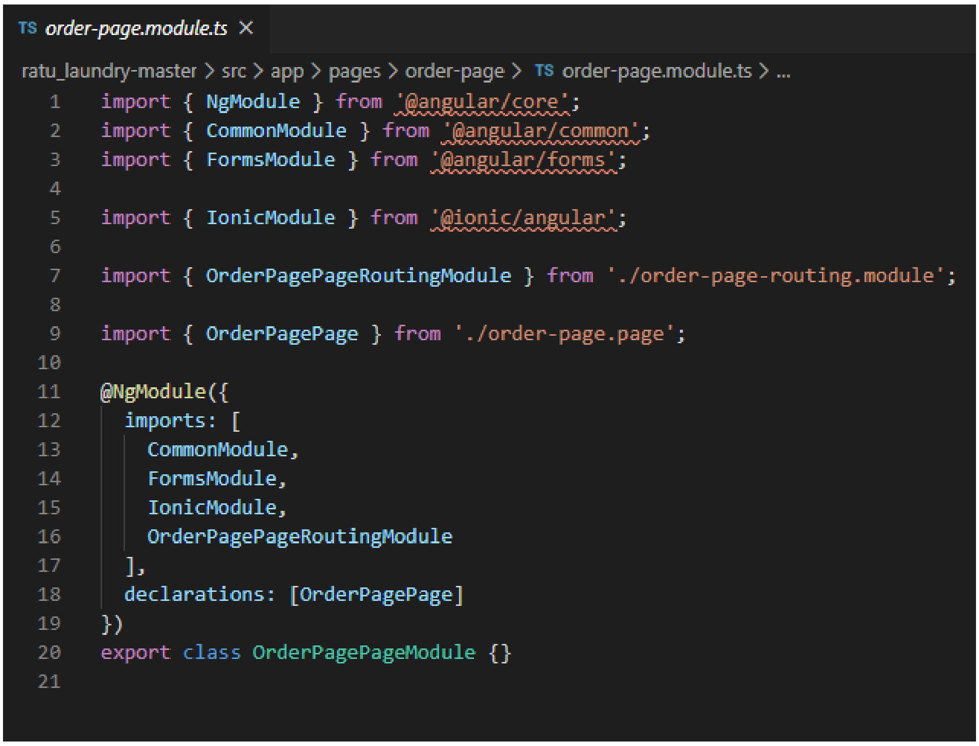
4. Order Page

Merupakan halaman dimana bisa menambah jumlah *Laundry* dan mengurang kan nya yang berguna untuk menghitung berapa biaya total *laundry* an kita.

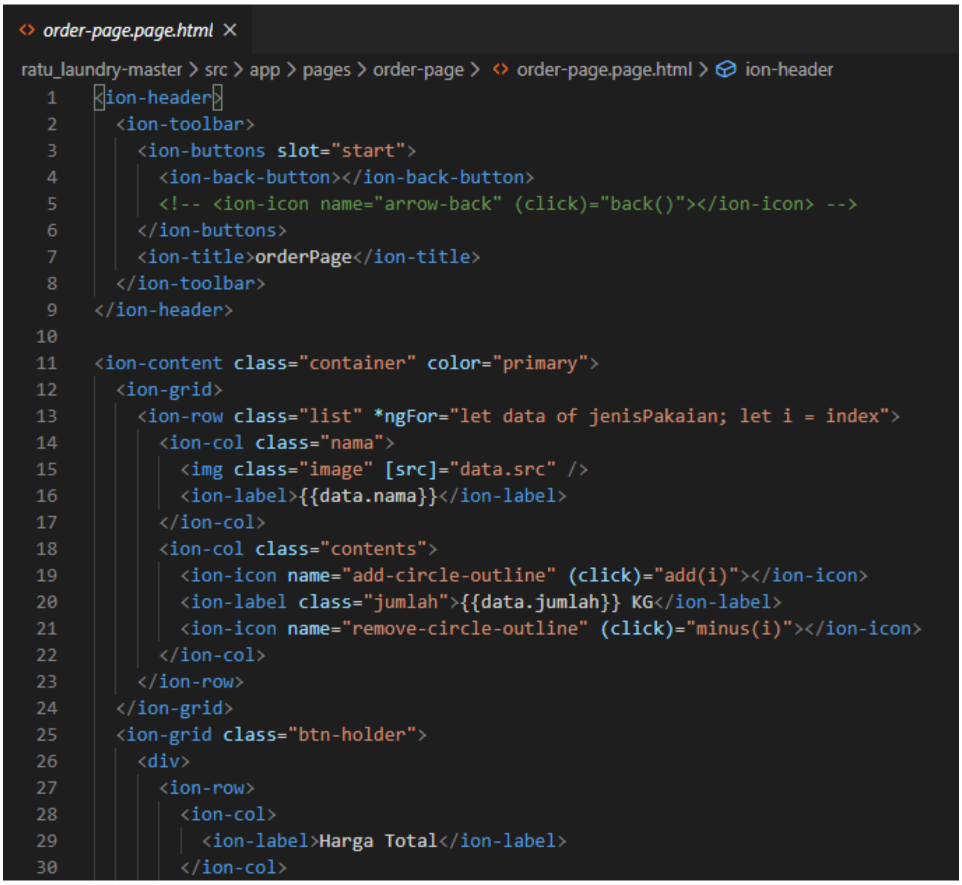



TS order-page-routing.module.ts ×
ratu_laundry-master > src > app > pages > order-page > TS order-page-routing.module.ts > ...

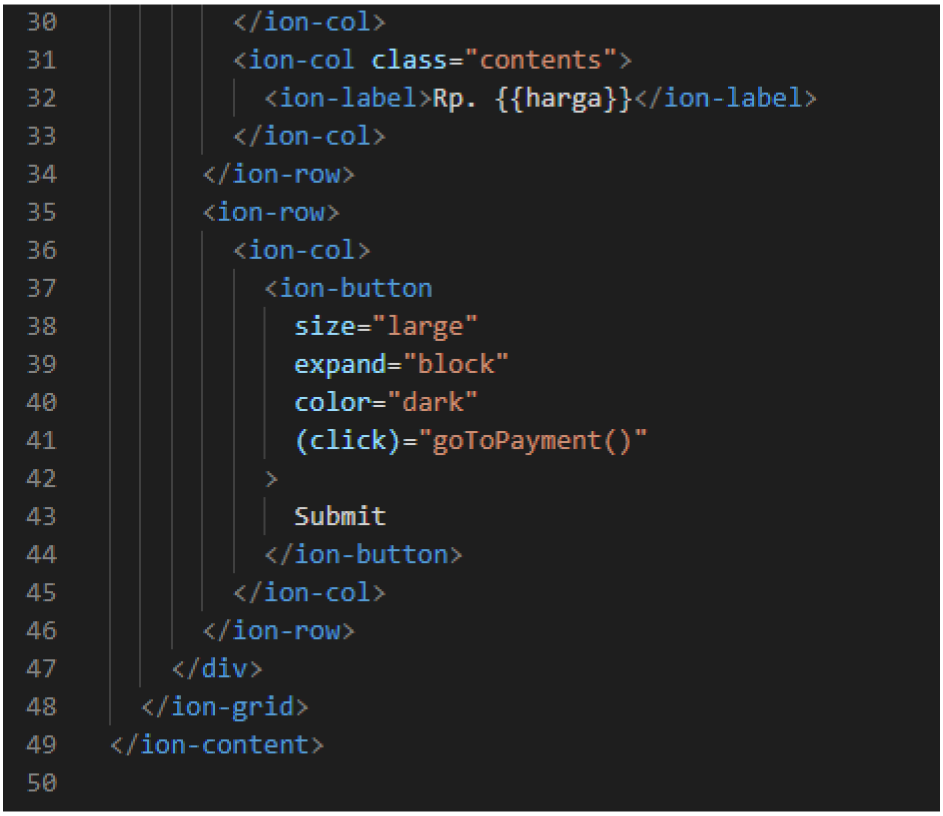
```typescript
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { OrderPagePage } from './order-page.page';

const routes: Routes = [
  {
    path: '',
    component: OrderPagePage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class OrderPagePageRoutingModule {}

```


TS order-page.module.ts ×
ratu_laundry-master > src > app > pages > order-page > TS order-page.module.ts > ...

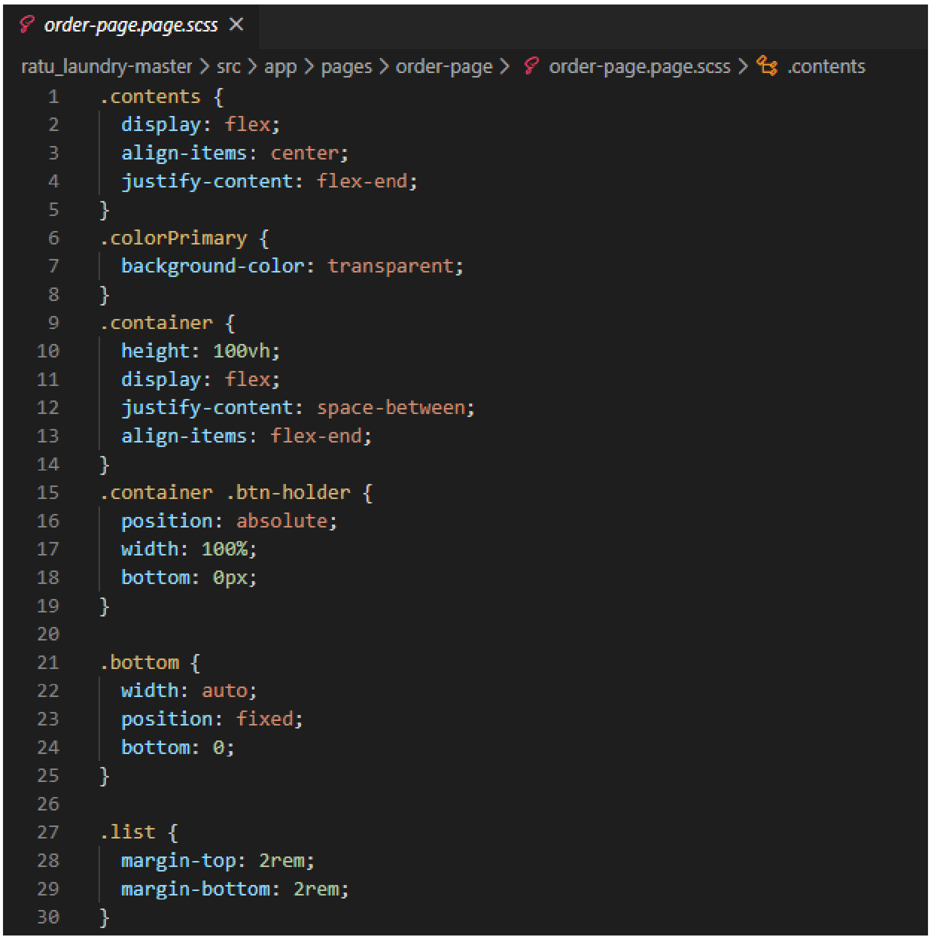
```typescript
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { OrderPagePageRoutingModule } from './order-page-routing.module';

import { OrderPagePage } from './order-page.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    OrderPagePageRoutingModule
  ],
  declarations: [OrderPagePage]
})
export class OrderPagePageModule {}
```


order-page.page.html ×
ratu_laundry-master > src > app > pages > order-page > order-page.page.html > ion-header

```html
<ion-header>
  <ion-toolbar>
    <ion-buttons slot="start">
      <ion-back-button></ion-back-button>
      <!-- <ion-icon name="arrow-back" (click)="back()"></ion-icon> -->
    </ion-buttons>
    <ion-title>orderPage</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content class="container" color="primary">
  <ion-grid>
    <ion-row class="list" *ngFor="let data of jenisPakaian; let i = index">
      <ion-col class="nama">
        <img class="image" [src]="data.src" />
        <ion-label>{{data.nama}}</ion-label>
      </ion-col>
      <ion-col class="contents">
        <ion-icon name="add-circle-outline" (click)="add(i)"></ion-icon>
        <ion-label class="jumlah">{{data.jumlah}} KG</ion-label>
        <ion-icon name="remove-circle-outline" (click)="minus(i)"></ion-icon>
      </ion-col>
    </ion-row>
  </ion-grid>
  <ion-grid class="btn-holder">
    <div>
      <ion-row>
        <ion-col>
          <ion-label>Harga Total</ion-label>
        </ion-col>
```

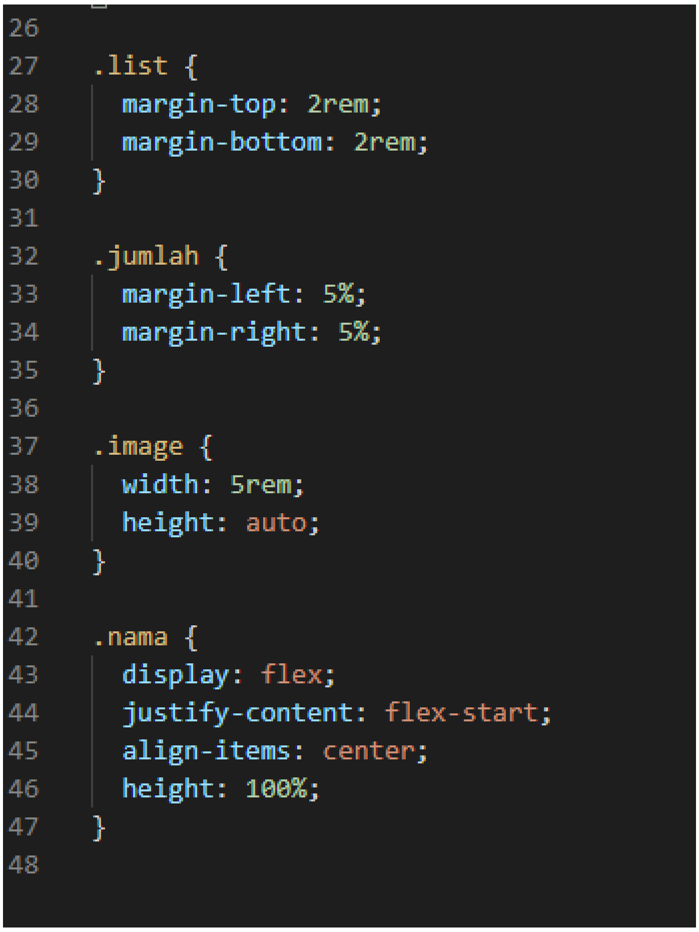


```html
        </ion-col>
        <ion-col class="contents">
          <ion-label>Rp. {{harga}}</ion-label>
        </ion-col>
      </ion-row>
      <ion-row>
        <ion-col>
          <ion-button
            size="large"
            expand="block"
            color="dark"
            (click)="goToPayment()"
          >
            Submit
          </ion-button>
        </ion-col>
      </ion-row>
    </div>
  </ion-grid>
</ion-content>

```

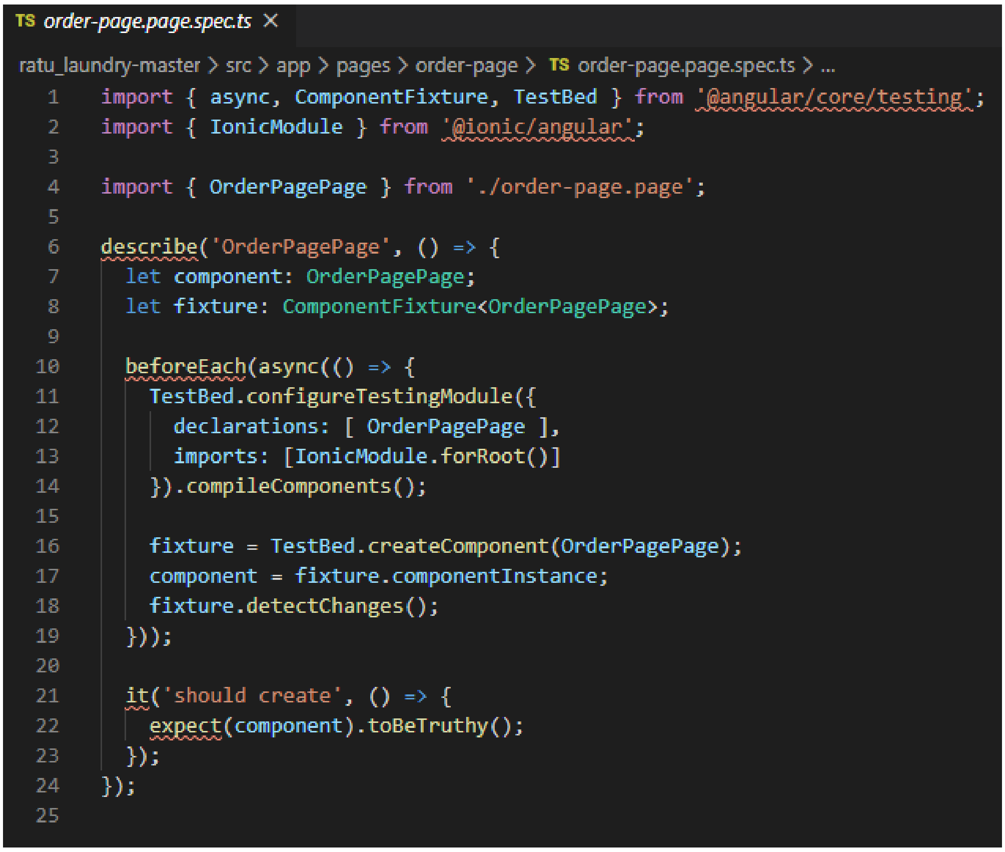


order-page.page.scss ×
ratu_laundry-master > src > app > pages > order-page > order-page.page.scss > .contents

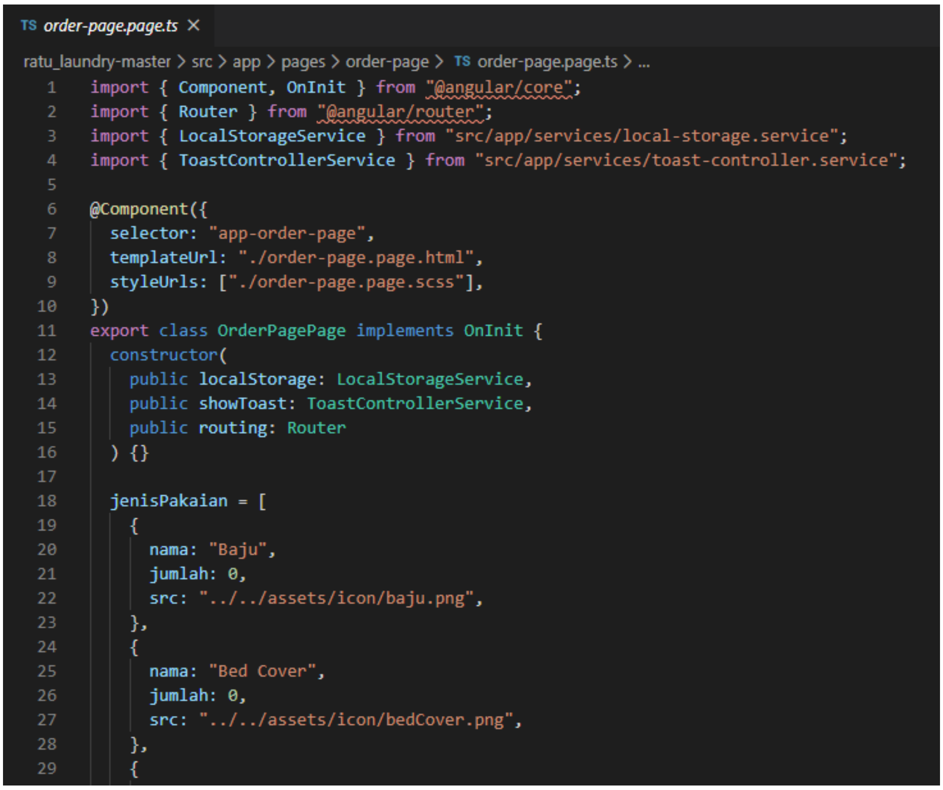
```scss
.contents {
  display: flex;
  align-items: center;
  justify-content: flex-end;
}
.colorPrimary {
  background-color: transparent;
}
.container {
  height: 100vh;
  display: flex;
  justify-content: space-between;
  align-items: flex-end;
}
.container .btn-holder {
  position: absolute;
  width: 100%;
  bottom: 0px;
}

.bottom {
  width: auto;
  position: fixed;
  bottom: 0;
}

.list {
  margin-top: 2rem;
  margin-bottom: 2rem;
}
```



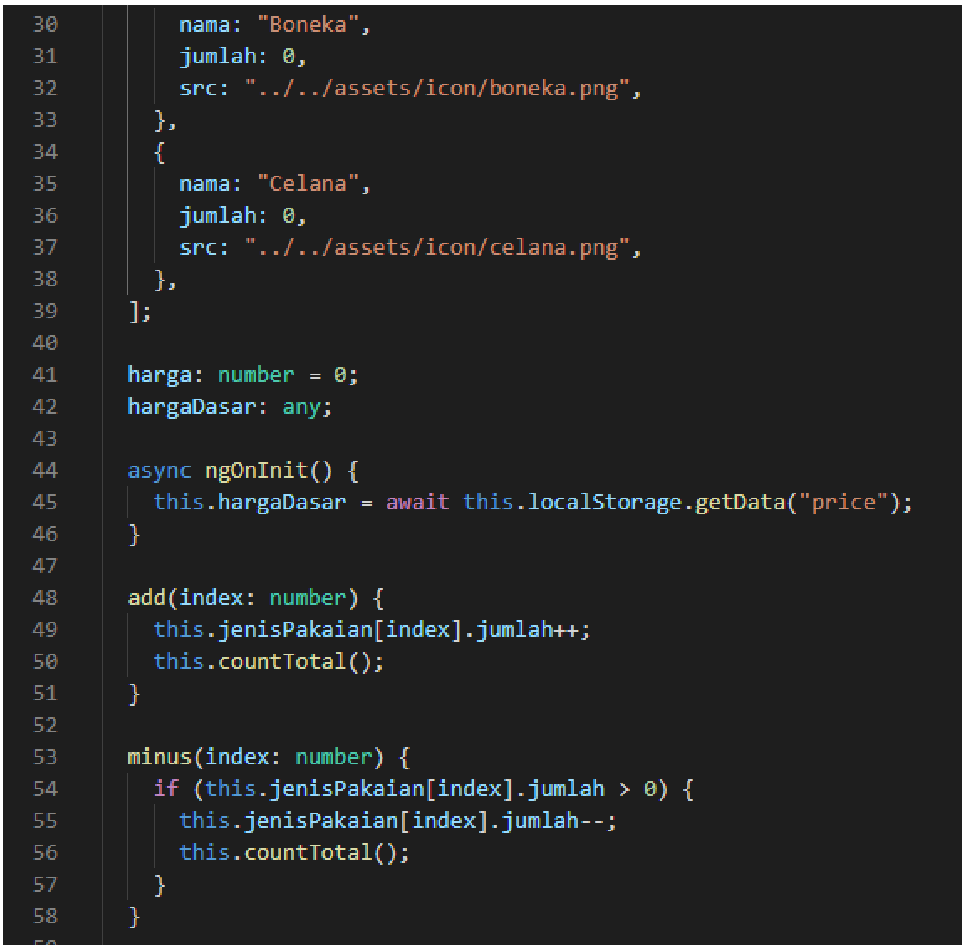
```scss

.list {
  margin-top: 2rem;
  margin-bottom: 2rem;
}

.jumlah {
  margin-left: 5%;
  margin-right: 5%;
}

.image {
  width: 5rem;
  height: auto;
}

.nama {
  display: flex;
  justify-content: flex-start;
  align-items: center;
  height: 100%;
}

```


TS order-page.page.spec.ts ×
ratu_laundry-master > src > app > pages > order-page > TS order-page.page.spec.ts > ...

```typescript
import { async, ComponentFixture, TestBed } from '@angular/core/testing';
import { IonicModule } from '@ionic/angular';

import { OrderPagePage } from './order-page.page';

describe('OrderPagePage', () => {
  let component: OrderPagePage;
  let fixture: ComponentFixture<OrderPagePage>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ OrderPagePage ],
      imports: [IonicModule.forRoot()]
    }).compileComponents();

    fixture = TestBed.createComponent(OrderPagePage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  }));

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```


TS order-page.page.ts ×
ratu_laundry-master > src > app > pages > order-page > TS order-page.page.ts > ...

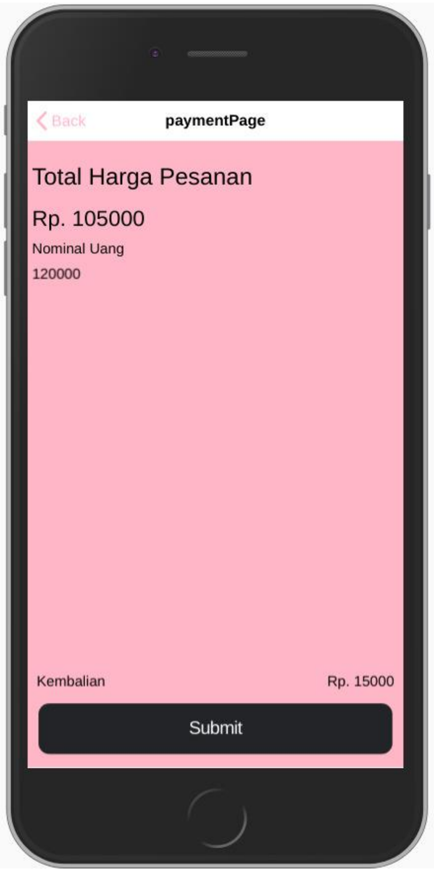
```typescript
import { Component, OnInit } from "@angular/core";
import { Router } from "@angular/router";
import { LocalStorageService } from "src/app/services/local-storage.service";
import { ToastControllerService } from "src/app/services/toast-controller.service";

@Component({
  selector: "app-order-page",
  templateUrl: "./order-page.page.html",
  styleUrls: ["./order-page.page.scss"],
})
export class OrderPagePage implements OnInit {
  constructor(
    public localStorage: LocalStorageService,
    public showToast: ToastControllerService,
    public routing: Router
  ) {}

  jenisPakaian = [
    {
      nama: "Baju",
      jumlah: 0,
      src: "../../assets/icon/baju.png",
    },
    {
      nama: "Bed Cover",
      jumlah: 0,
      src: "../../assets/icon/bedCover.png",
    },
    {
```



```typescript
      nama: "Boneka",
      jumlah: 0,
      src: "../../assets/icon/boneka.png",
    },
    {
      nama: "Celana",
      jumlah: 0,
      src: "../../assets/icon/celana.png",
    },
  ];

  harga: number = 0;
  hargaDasar: any;

  async ngOnInit() {
    this.hargaDasar = await this.localStorage.getData("price");
  }

  add(index: number) {
    this.jenisPakaian[index].jumlah++;
    this.countTotal();
  }

  minus(index: number) {
    if (this.jenisPakaian[index].jumlah > 0) {
      this.jenisPakaian[index].jumlah--;
      this.countTotal();
    }
  }
```

5. Payment Page

Merupakan halaman dimana total dan uang yang anda masukkan kemudian transaksi.



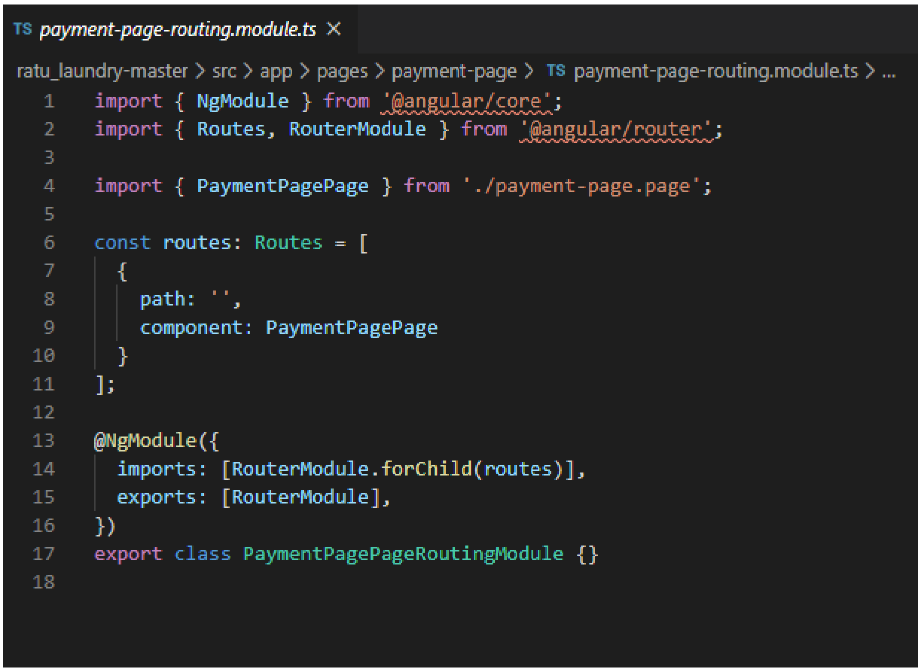
Total Harga Pesanan

Rp. 105000

Nominal Uang

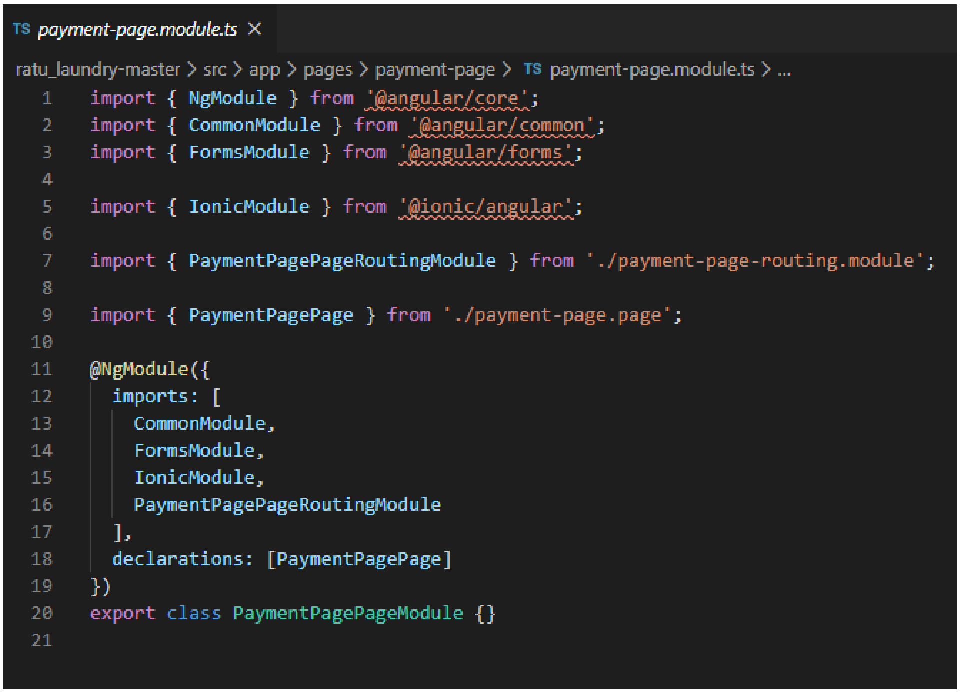
120000

Kembalian Rp. 15000

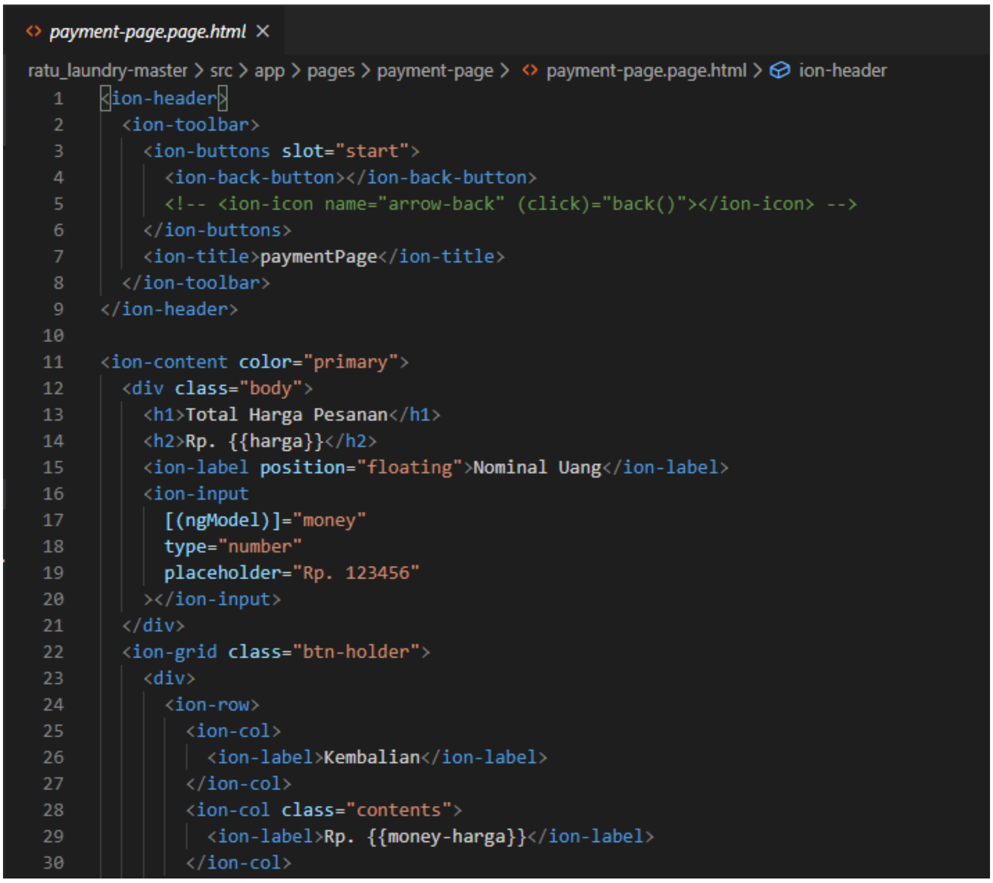


payment-page-routing.module.ts
ratu_laundry-master > src > app > pages > payment-page > payment-page-routing.module.ts > ...

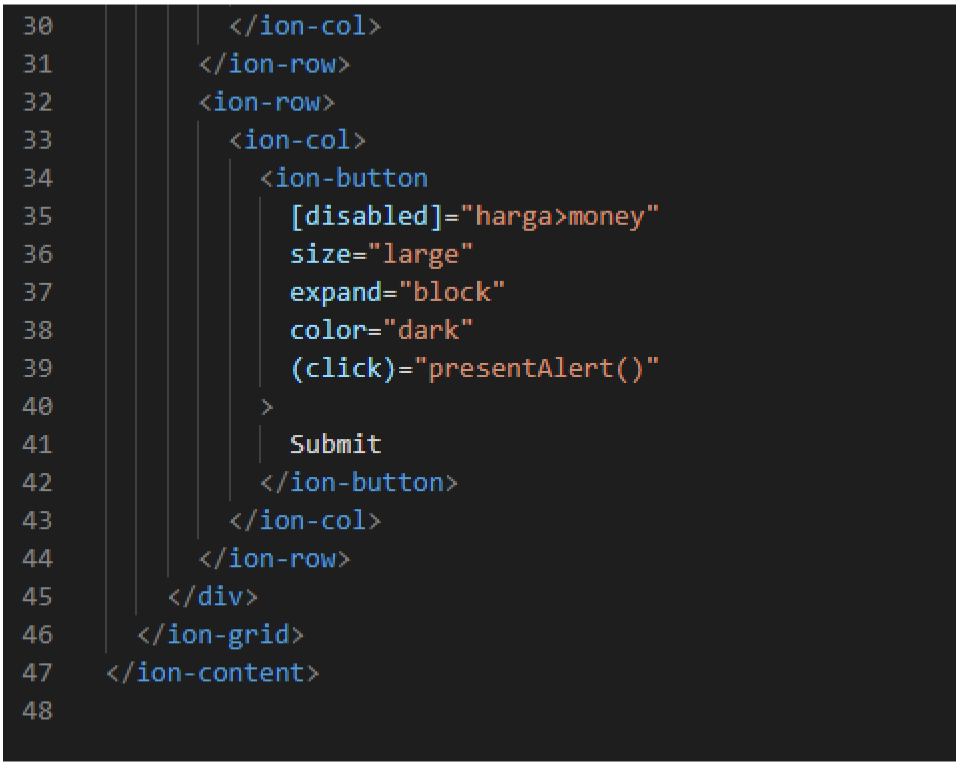
```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { PaymentPagePage } from './payment-page.page';

const routes: Routes = [
  {
    path: '',
    component: PaymentPagePage
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class PaymentPagePageRoutingModule {}
```


TS payment-page.module.ts ×
ratu_laundry-master > src > app > pages > payment-page > TS payment-page.module.ts > ...

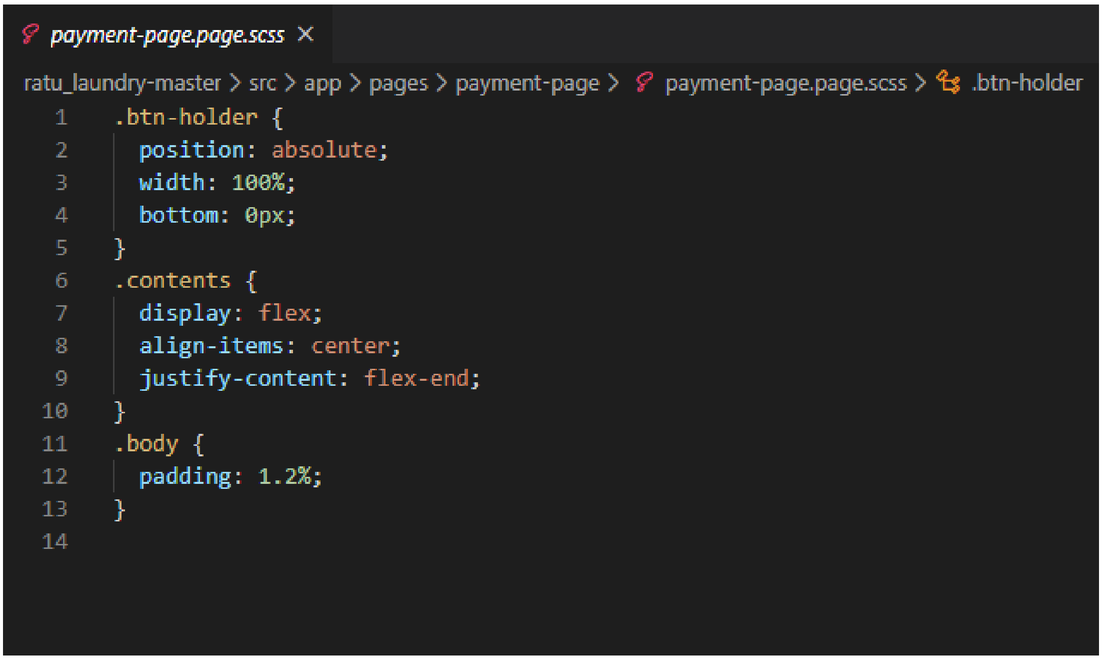
```typescript
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';

import { IonicModule } from '@ionic/angular';

import { PaymentPagePageRoutingModule } from './payment-page-routing.module';

import { PaymentPagePage } from './payment-page.page';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    PaymentPagePageRoutingModule
  ],
  declarations: [PaymentPagePage]
})
export class PaymentPagePageModule {}

```


payment-page.page.html ×
ratu_laundry-master > src > app > pages > payment-page > payment-page.page.html > ion-header

```html
<ion-header>
  <ion-toolbar>
    <ion-buttons slot="start">
      <ion-back-button></ion-back-button>
      <!-- <ion-icon name="arrow-back" (click)="back()"></ion-icon> -->
    </ion-buttons>
    <ion-title>paymentPage</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content color="primary">
  <div class="body">
    <h1>Total Harga Pesanan</h1>
    <h2>Rp. {{harga}}</h2>
    <ion-label position="floating">Nominal Uang</ion-label>
    <ion-input
      [(ngModel)]="money"
      type="number"
      placeholder="Rp. 123456"
    ></ion-input>
  </div>
  <ion-grid class="btn-holder">
    <div>
      <ion-row>
        <ion-col>
          <ion-label>Kembalian</ion-label>
        </ion-col>
        <ion-col class="contents">
          <ion-label>Rp. {{money-harga}}</ion-label>
        </ion-col>
```



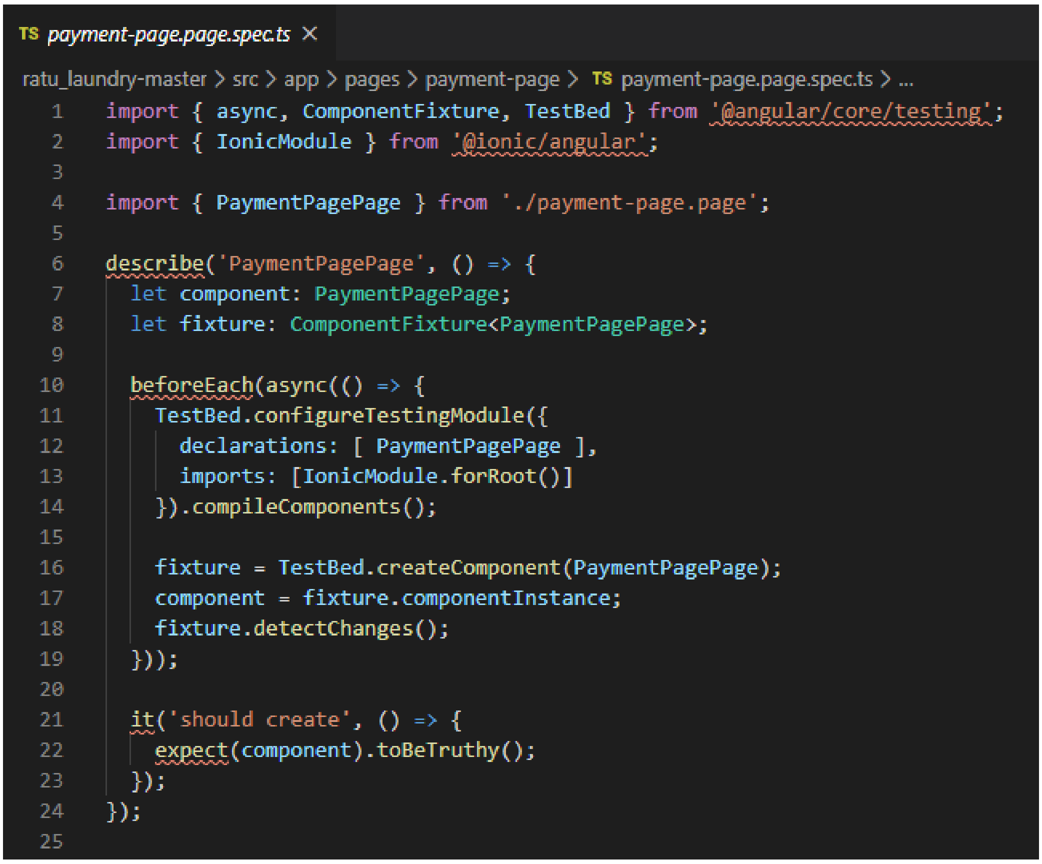
```
        </ion-col>
      </ion-row>
      <ion-row>
        <ion-col>
          <ion-button
            [disabled]="harga>money"
            size="large"
            expand="block"
            color="dark"
            (click)="presentAlert()"
          >
            Submit
          </ion-button>
        </ion-col>
      </ion-row>
    </div>
  </ion-grid>
</ion-content>

```



payment-page.page.scss ×

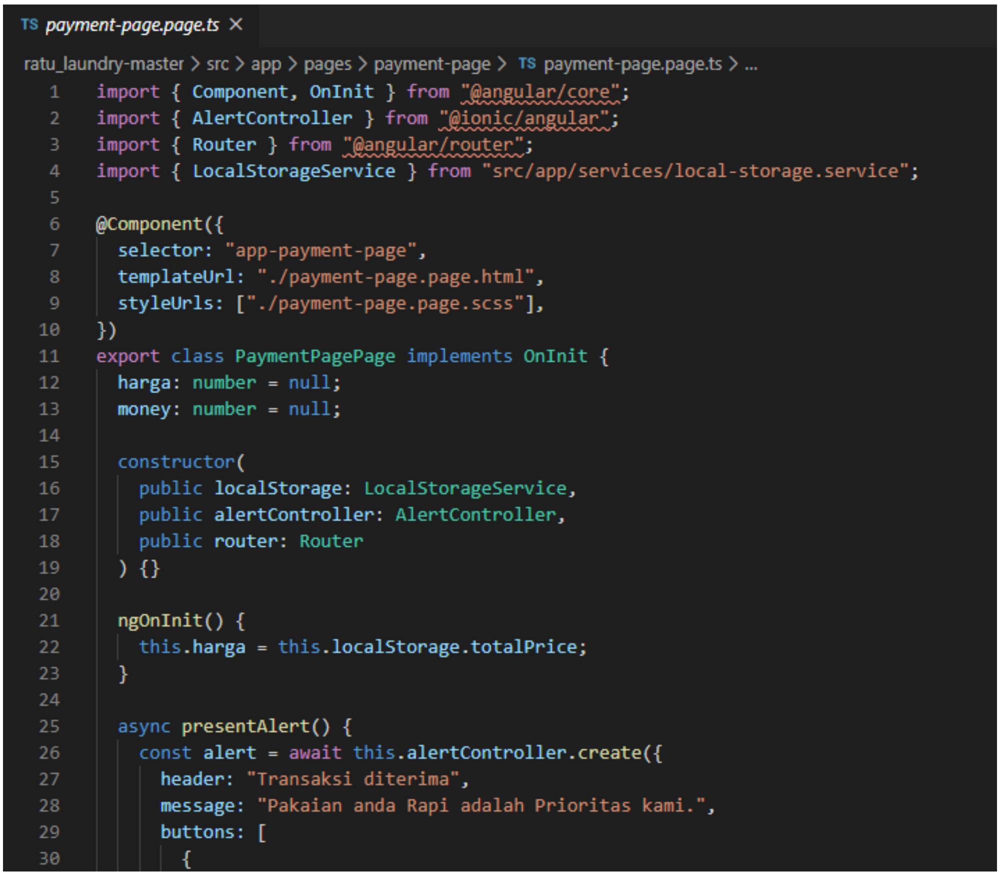
ratu_laundry-master > src > app > pages > payment-page > payment-page.page.scss > .btn-holder

```
.btn-holder {
  position: absolute;
  width: 100%;
  bottom: 0px;
}
.contents {
  display: flex;
  align-items: center;
  justify-content: flex-end;
}
.body {
  padding: 1.2%;
}

```



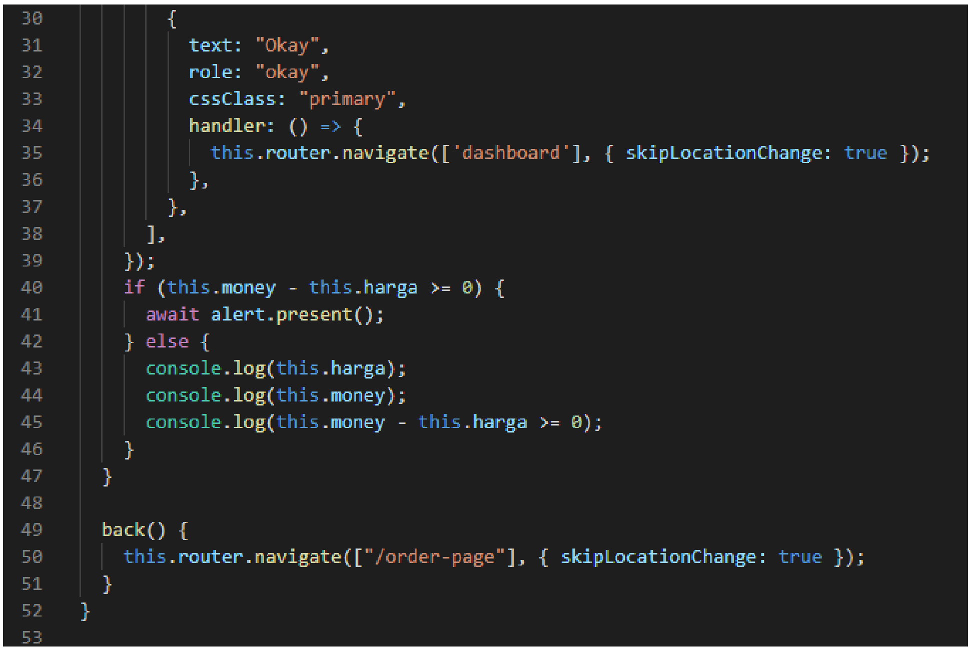
TS payment-page.page.spec.ts ×
ratu_laundry-master > src > app > pages > payment-page > TS payment-page.page.spec.ts > ...

```typescript
import { async, ComponentFixture, TestBed } from '@angular/core/testing';
import { IonicModule } from '@ionic/angular';

import { PaymentPagePage } from './payment-page.page';

describe('PaymentPagePage', () => {
  let component: PaymentPagePage;
  let fixture: ComponentFixture<PaymentPagePage>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ PaymentPagePage ],
      imports: [IonicModule.forRoot()]
    }).compileComponents();

    fixture = TestBed.createComponent(PaymentPagePage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  }));

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

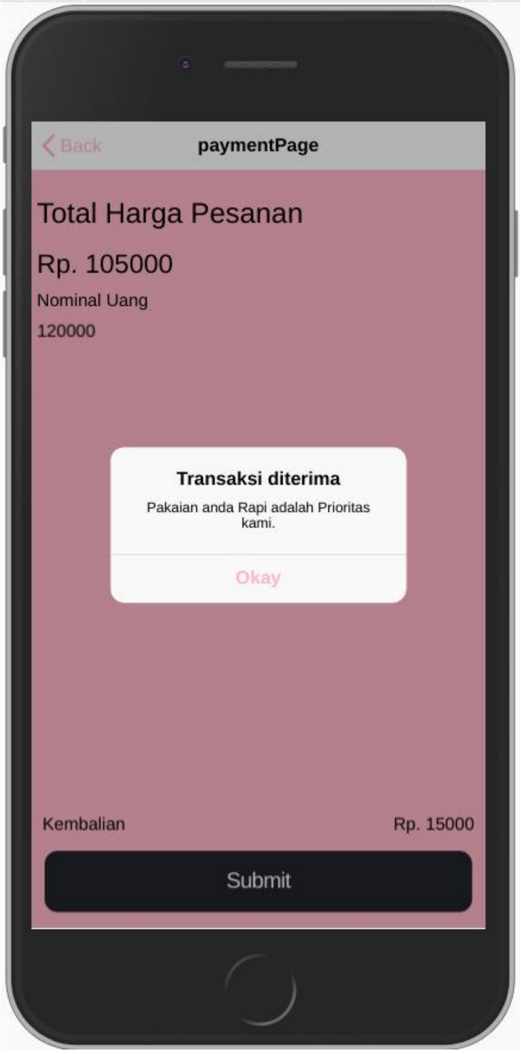


TS payment-page.page.ts ×
ratu_laundry-master > src > app > pages > payment-page > TS payment-page.page.ts > ...

```typescript
import { Component, OnInit } from "@angular/core";
import { AlertController } from "@ionic/angular";
import { Router } from "@angular/router";
import { LocalStorageService } from "src/app/services/local-storage.service";

@Component({
  selector: "app-payment-page",
  templateUrl: "./payment-page.page.html",
  styleUrls: ["./payment-page.page.scss"],
})
export class PaymentPagePage implements OnInit {
  harga: number = null;
  money: number = null;

  constructor(
    public localStorage: LocalStorageService,
    public alertController: AlertController,
    public router: Router
  ) {}

  ngOnInit() {
    this.harga = this.localStorage.totalPrice;
  }

  async presentAlert() {
    const alert = await this.alertController.create({
      header: "Transaksi diterima",
      message: "Pakaian anda Rapi adalah Prioritas kami.",
      buttons: [
        {
```



```
        {
          text: "Okay",
          role: "okay",
          cssClass: "primary",
          handler: () => {
            this.router.navigate(['dashboard'], { skipLocationChange: true });
          },
        },
      ],
    });
    if (this.money - this.harga >= 0) {
      await alert.present();
    } else {
      console.log(this.harga);
      console.log(this.money);
      console.log(this.money - this.harga >= 0);
    }
  }

  back() {
    this.router.navigate(["/order-page"], { skipLocationChange: true });
  }
}

```



paymentPage

Total Harga Pesanan

Rp. 105000

Nominal Uang

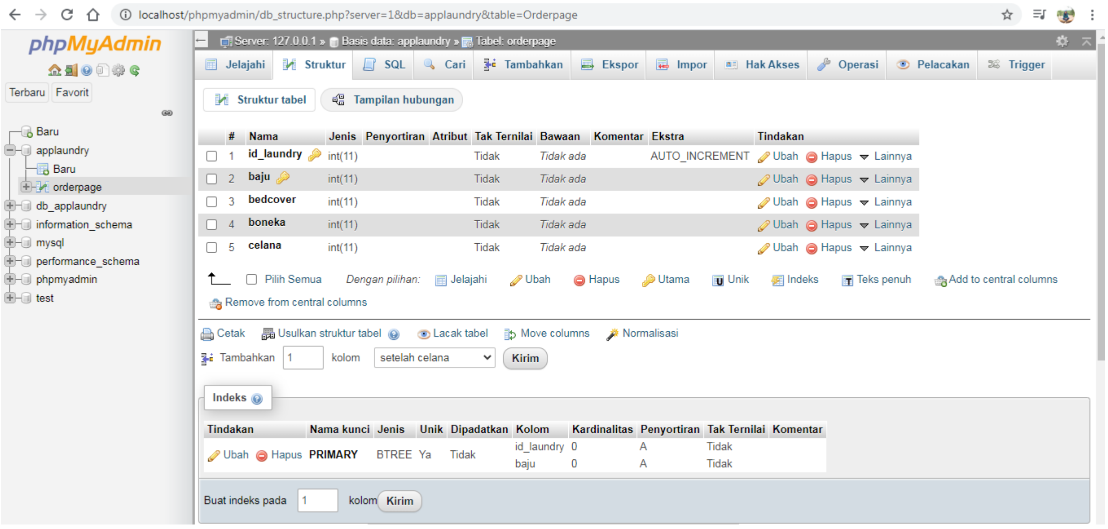
120000

Transaksi diterima

Pakaian anda Rapi adalah Prioritas kami.

Kembalian Rp. 15000

DATABASE ORDERPAGE



	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
□	1	**id_laundry**	int(11)			Tidak	*Tidak ada*		AUTO_INCREMENT	Ubah Hapus Lainnya
□	2	**baju**	int(11)			Tidak	*Tidak ada*			Ubah Hapus Lainnya
□	3	**bedcover**	int(11)			Tidak	*Tidak ada*			Ubah Hapus Lainnya
□	4	**boneka**	int(11)			Tidak	*Tidak ada*			Ubah Hapus Lainnya
□	5	**celana**	int(11)			Tidak	*Tidak ada*			Ubah Hapus Lainnya

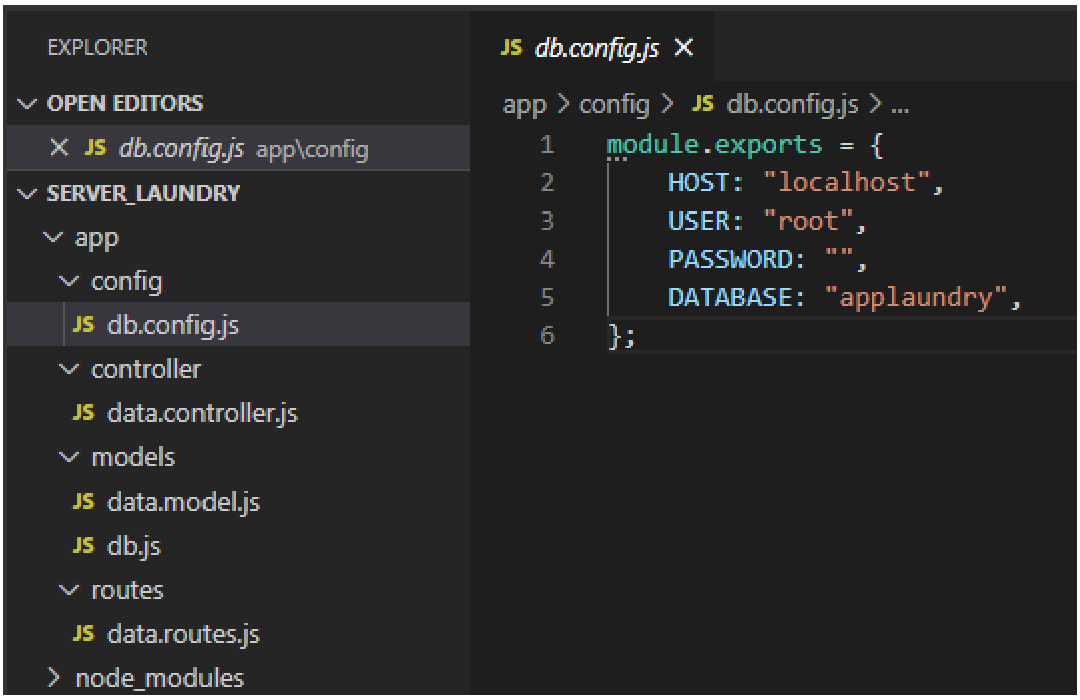
Tambahkan 1 kolom setelah celana Kirim

Indeks

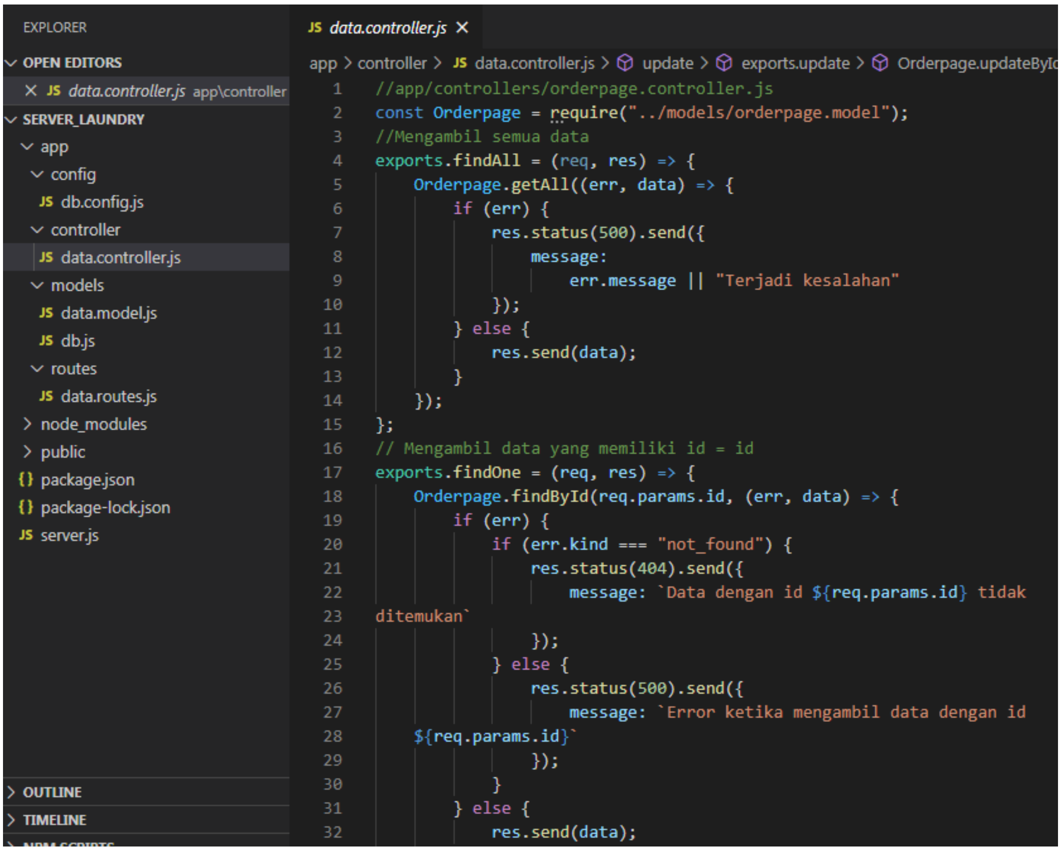
Tindakan	Nama kunci	Jenis	Unik	Dipadatkan	Kolom	Kardinalitas	Penyortiran	Tak Ternilai	Komentar
Ubah Hapus	**PRIMARY**	BTREE	Ya	Tidak	id_laundry	0	A	Tidak	
					baju	0	A	Tidak	

Buat indeks pada 1 kolom Kirim

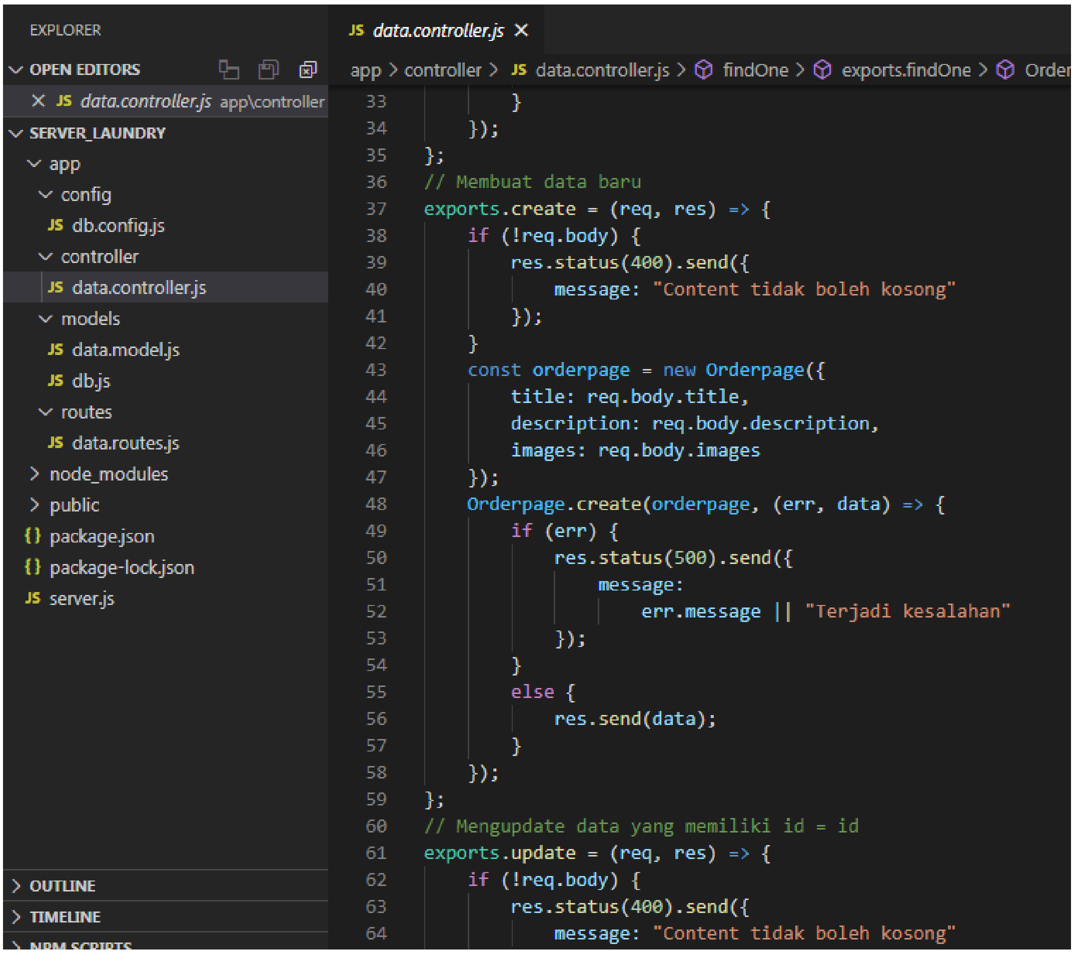
SERVER



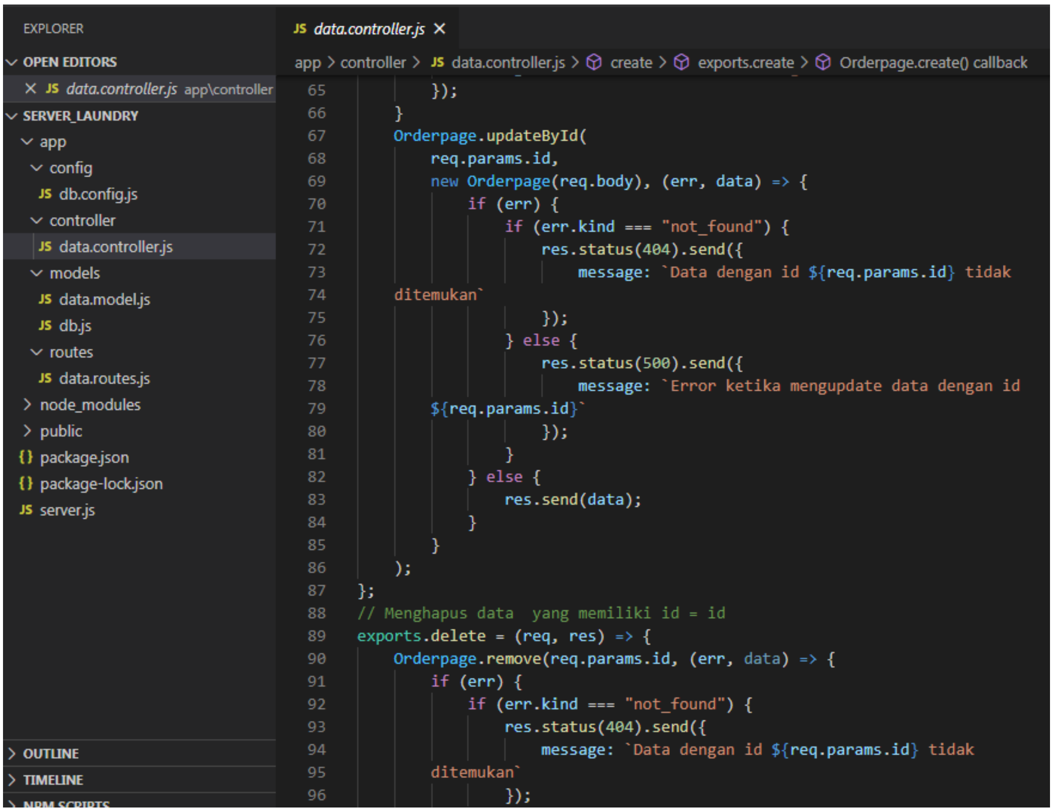
```
module.exports = {
    HOST: "localhost",
    USER: "root",
    PASSWORD: "",
    DATABASE: "applaundry",
};
```



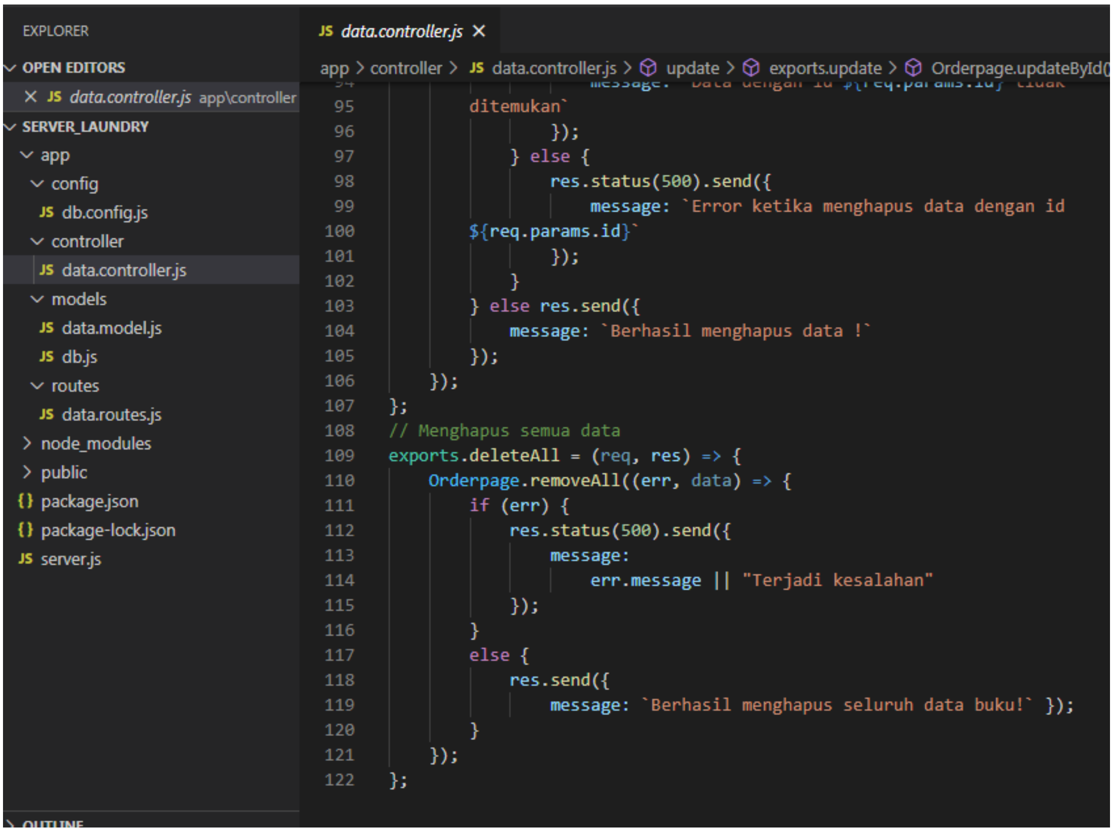
```
//app/controllers/orderpage.controller.js
const Orderpage = require("../models/orderpage.model");
//Mengambil semua data
exports.findAll = (req, res) => {
    Orderpage.getAll((err, data) => {
        if (err) {
            res.status(500).send({
                message:
                    err.message || "Terjadi kesalahan"
            });
        } else {
            res.send(data);
        }
    });
};
// Mengambil data yang memiliki id = id
exports.findOne = (req, res) => {
    Orderpage.findById(req.params.id, (err, data) => {
        if (err) {
            if (err.kind === "not_found") {
                res.status(404).send({
                    message: `Data dengan id ${req.params.id} tidak
ditemukan`
                });
            } else {
                res.status(500).send({
                    message: `Error ketika mengambil data dengan id
    ${req.params.id}`
                });
            }
        } else {
            res.send(data);
```



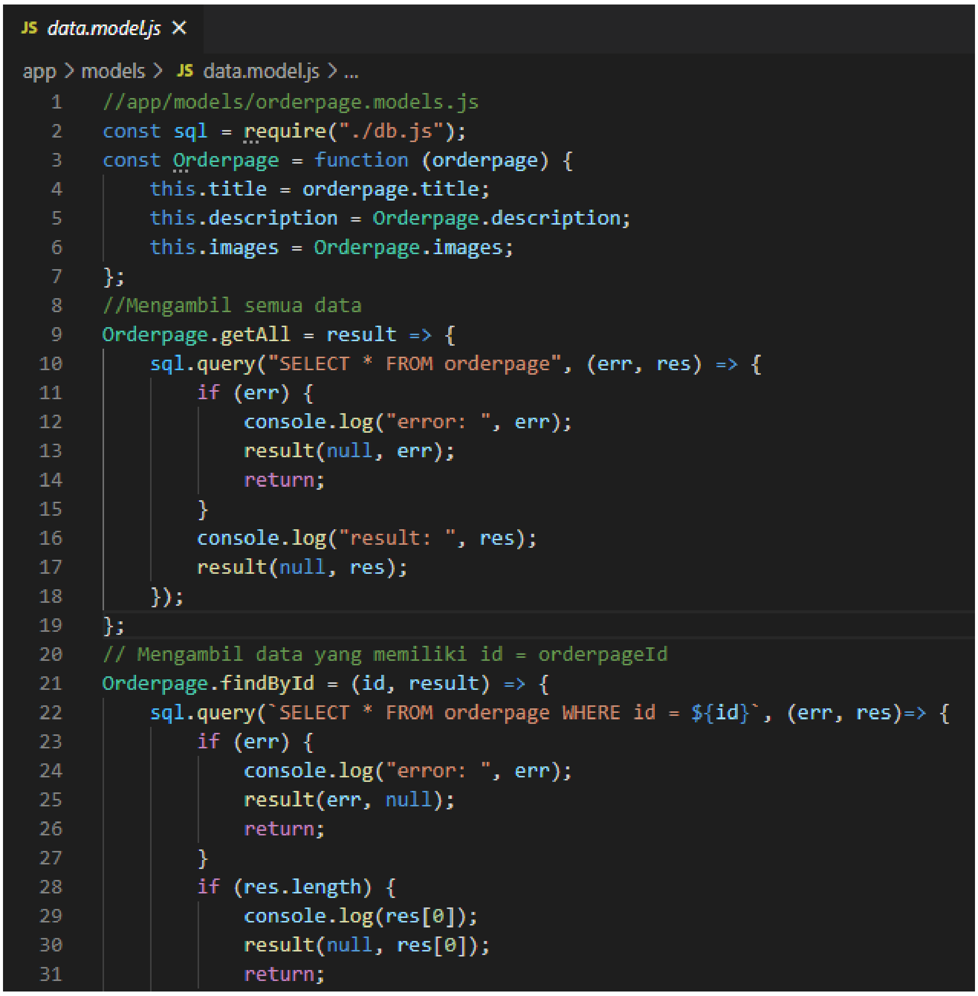
```
        }
    });
};
// Membuat data baru
exports.create = (req, res) => {
    if (!req.body) {
        res.status(400).send({
            message: "Content tidak boleh kosong"
        });
    }
    const orderpage = new Orderpage({
        title: req.body.title,
        description: req.body.description,
        images: req.body.images
    });
    Orderpage.create(orderpage, (err, data) => {
        if (err) {
            res.status(500).send({
                message:
                    err.message || "Terjadi kesalahan"
            });
        }
        else {
            res.send(data);
        }
    });
};
// Mengupdate data yang memiliki id = id
exports.update = (req, res) => {
    if (!req.body) {
        res.status(400).send({
            message: "Content tidak boleh kosong"
```



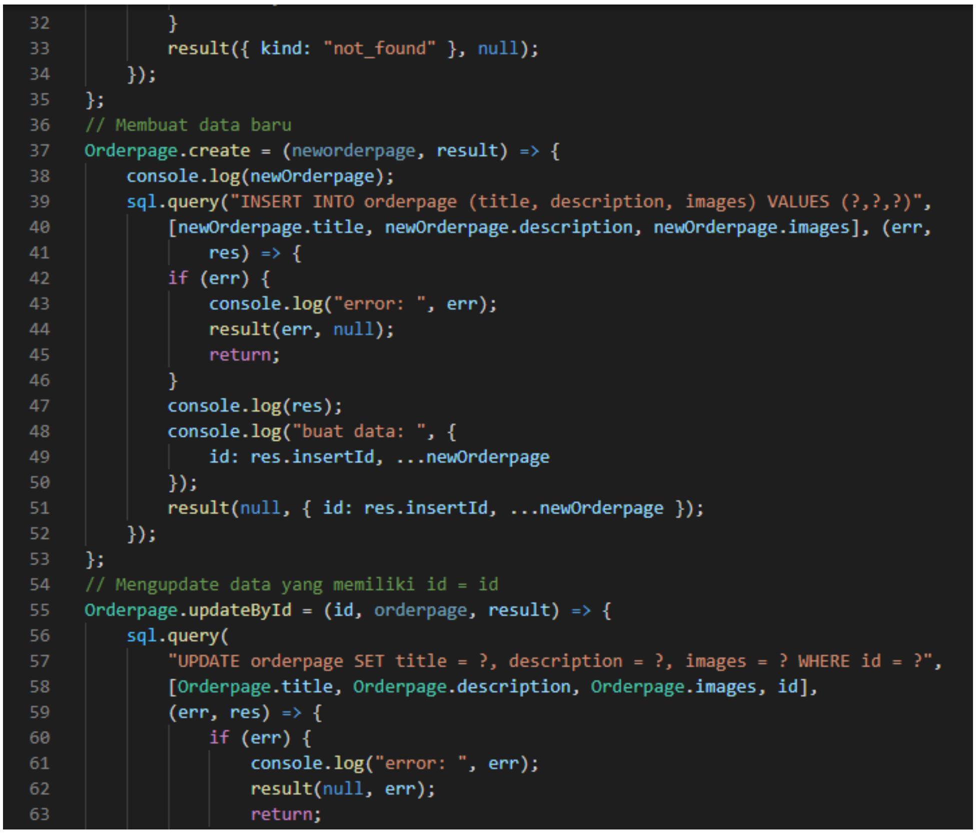
```
        });
    }
    Orderpage.updateById(
        req.params.id,
        new Orderpage(req.body), (err, data) => {
            if (err) {
                if (err.kind === "not_found") {
                    res.status(404).send({
                        message: `Data dengan id ${req.params.id} tidak
    ditemukan`
                    });
                } else {
                    res.status(500).send({
                        message: `Error ketika mengupdate data dengan id
        ${req.params.id}`
                    });
                }
            } else {
                res.send(data);
            }
        }
    );
};
// Menghapus data  yang memiliki id = id
exports.delete = (req, res) => {
    Orderpage.remove(req.params.id, (err, data) => {
        if (err) {
            if (err.kind === "not_found") {
                res.status(404).send({
                    message: `Data dengan id ${req.params.id} tidak
        ditemukan`
                });
```



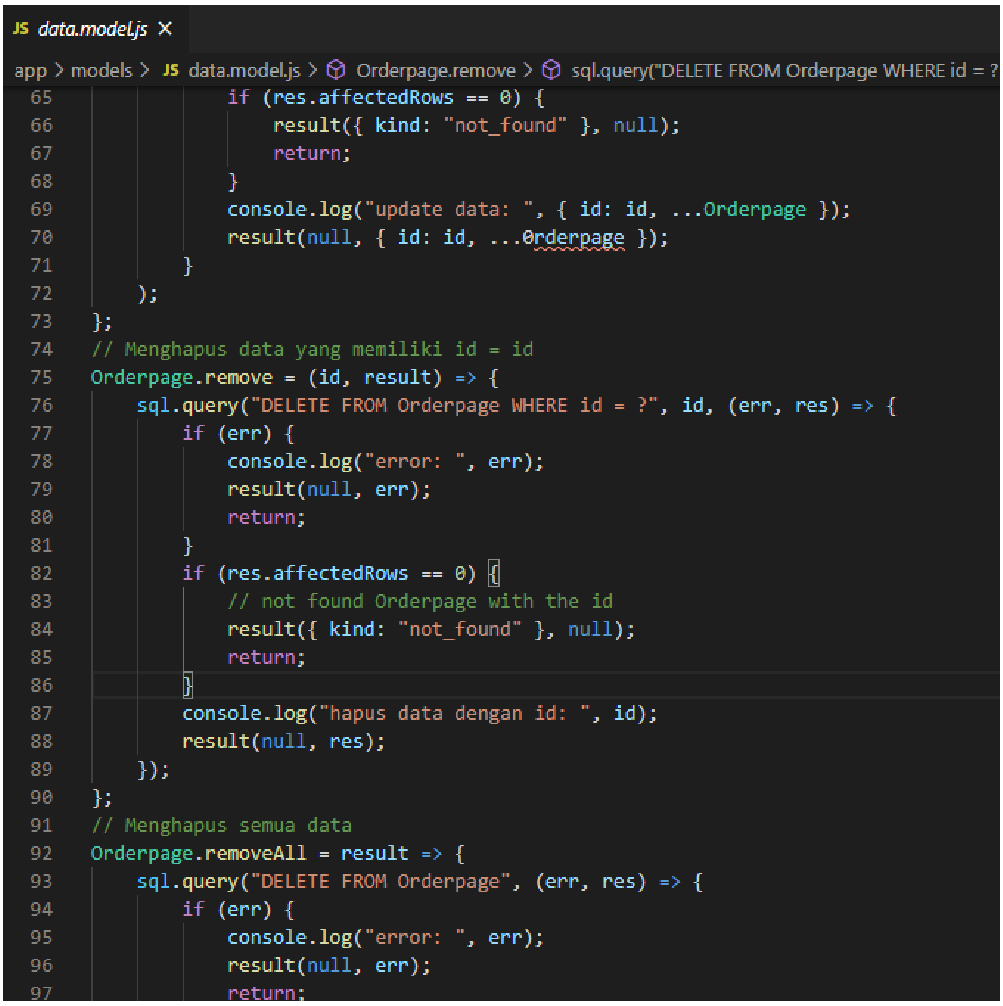
```
        ditemukan`
                });
            } else {
                res.status(500).send({
                    message: `Error ketika menghapus data dengan id
        ${req.params.id}`
                });
            }
        } else res.send({
            message: `Berhasil menghapus data !`
        });
    });
};
// Menghapus semua data
exports.deleteAll = (req, res) => {
    Orderpage.removeAll((err, data) => {
        if (err) {
            res.status(500).send({
                message:
                    err.message || "Terjadi kesalahan"
            });
        }
        else {
            res.send({
                message: `Berhasil menghapus seluruh data buku!` });
        }
    });
};
```



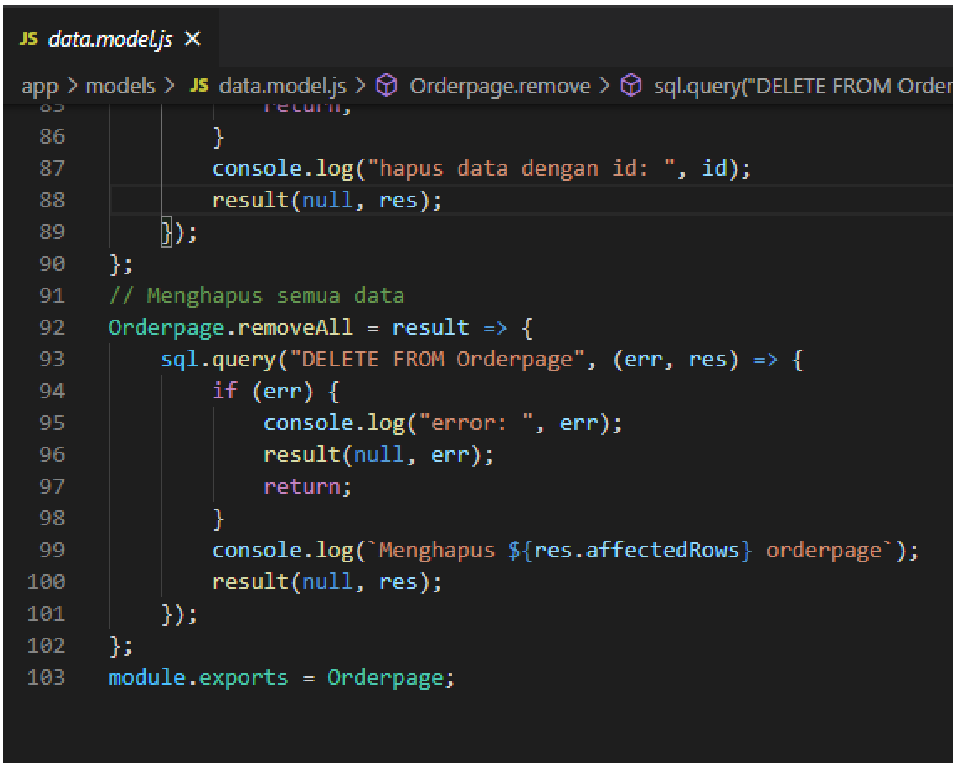
```javascript
//app/models/orderpage.models.js
const sql = require("./db.js");
const Orderpage = function (orderpage) {
    this.title = orderpage.title;
    this.description = Orderpage.description;
    this.images = Orderpage.images;
};
//Mengambil semua data
Orderpage.getAll = result => {
    sql.query("SELECT * FROM orderpage", (err, res) => {
        if (err) {
            console.log("error: ", err);
            result(null, err);
            return;
        }
        console.log("result: ", res);
        result(null, res);
    });
};
// Mengambil data yang memiliki id = orderpageId
Orderpage.findById = (id, result) => {
    sql.query(`SELECT * FROM orderpage WHERE id = ${id}`, (err, res)=> {
        if (err) {
            console.log("error: ", err);
            result(err, null);
            return;
        }
        if (res.length) {
            console.log(res[0]);
            result(null, res[0]);
            return;
```



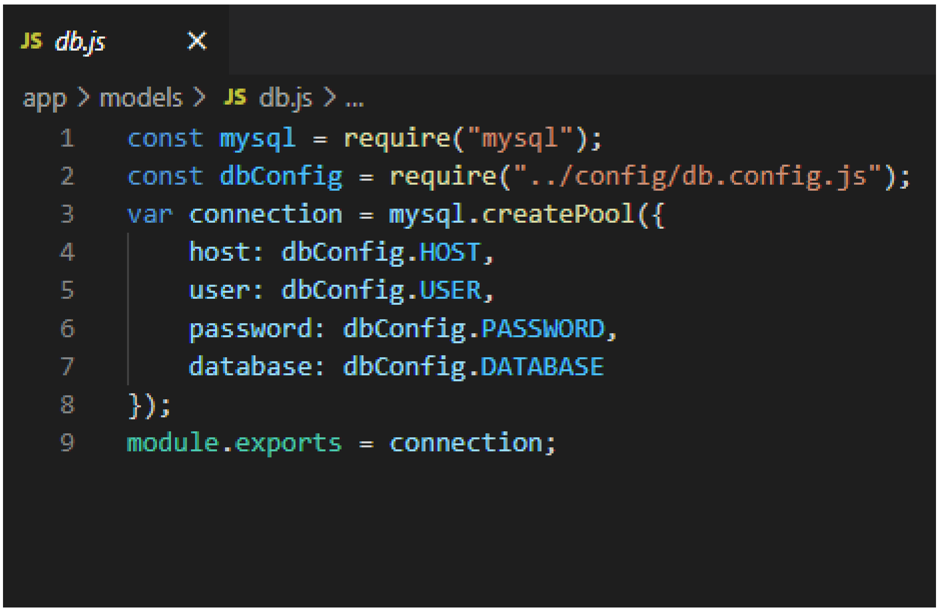
```javascript
        }
        result({ kind: "not_found" }, null);
    });
};
// Membuat data baru
Orderpage.create = (neworderpage, result) => {
    console.log(newOrderpage);
    sql.query("INSERT INTO orderpage (title, description, images) VALUES (?,?,?)",
        [newOrderpage.title, newOrderpage.description, newOrderpage.images], (err,
            res) => {
        if (err) {
            console.log("error: ", err);
            result(err, null);
            return;
        }
        console.log(res);
        console.log("buat data: ", {
            id: res.insertId, ...newOrderpage
        });
        result(null, { id: res.insertId, ...newOrderpage });
    });
};
// Mengupdate data yang memiliki id = id
Orderpage.updateById = (id, orderpage, result) => {
    sql.query(
        "UPDATE orderpage SET title = ?, description = ?, images = ? WHERE id = ?",
        [Orderpage.title, Orderpage.description, Orderpage.images, id],
        (err, res) => {
            if (err) {
                console.log("error: ", err);
                result(null, err);
                return;
```


JS data.model.js ×
app > models > JS data.model.js > Orderpage.remove > sql.query("DELETE FROM Orderpage WHERE id = ?

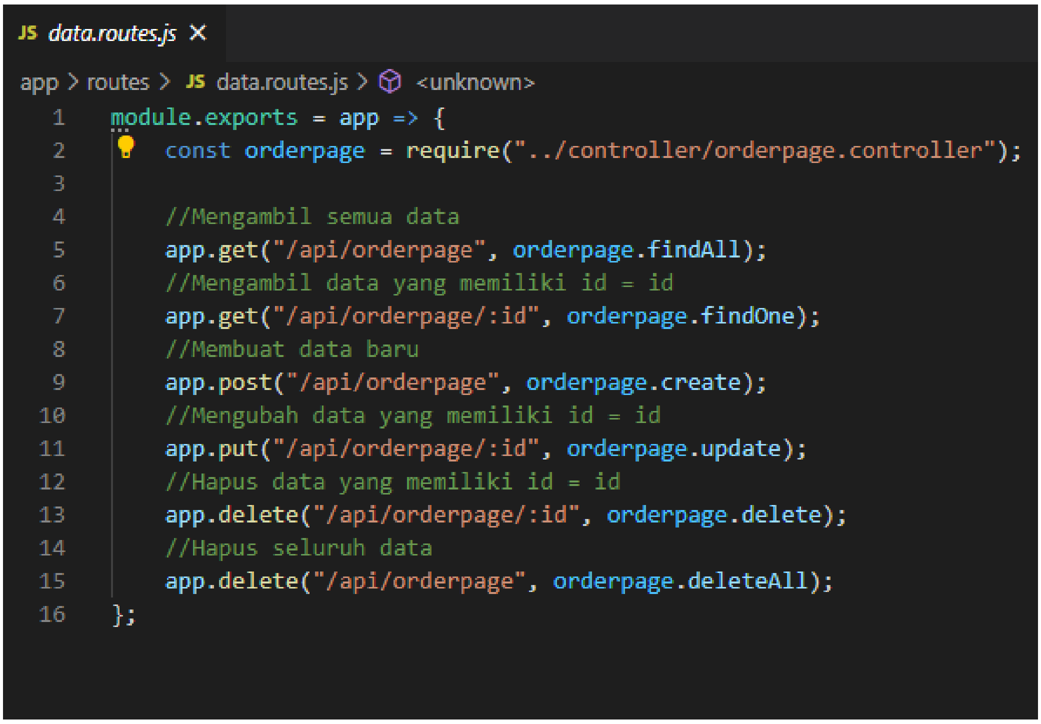
```
            if (res.affectedRows == 0) {
                result({ kind: "not_found" }, null);
                return;
            }
            console.log("update data: ", { id: id, ...Orderpage });
            result(null, { id: id, ...0rderpage });
        }
    );
};
// Menghapus data yang memiliki id = id
Orderpage.remove = (id, result) => {
    sql.query("DELETE FROM Orderpage WHERE id = ?", id, (err, res) => {
        if (err) {
            console.log("error: ", err);
            result(null, err);
            return;
        }
        if (res.affectedRows == 0) {
            // not found Orderpage with the id
            result({ kind: "not_found" }, null);
            return;
        }
        console.log("hapus data dengan id: ", id);
        result(null, res);
    });
};
// Menghapus semua data
Orderpage.removeAll = result => {
    sql.query("DELETE FROM Orderpage", (err, res) => {
        if (err) {
            console.log("error: ", err);
            result(null, err);
            return;
```


JS data.model.js ×
app > models > JS data.model.js > Orderpage.remove > sql.query("DELETE FROM Order

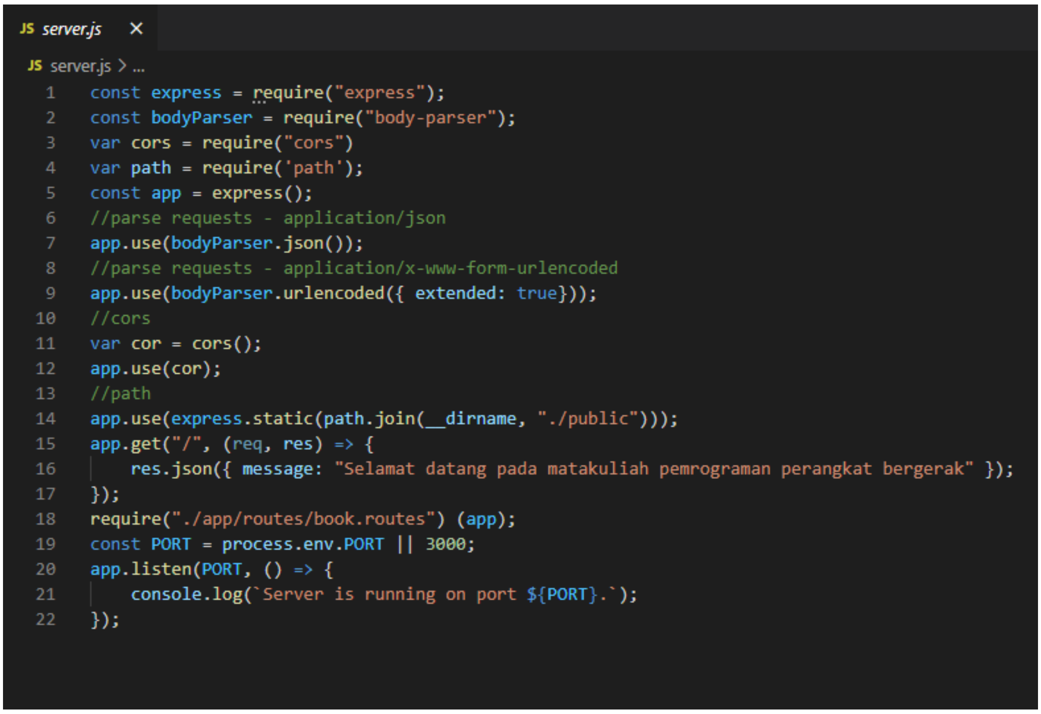
```
            return;
        }
        console.log("hapus data dengan id: ", id);
        result(null, res);
    });
};
// Menghapus semua data
Orderpage.removeAll = result => {
    sql.query("DELETE FROM Orderpage", (err, res) => {
        if (err) {
            console.log("error: ", err);
            result(null, err);
            return;
        }
        console.log(`Menghapus ${res.affectedRows} orderpage`);
        result(null, res);
    });
};
module.exports = Orderpage;
```



```javascript
const mysql = require("mysql");
const dbConfig = require("../config/db.config.js");
var connection = mysql.createPool({
    host: dbConfig.HOST,
    user: dbConfig.USER,
    password: dbConfig.PASSWORD,
    database: dbConfig.DATABASE
});
module.exports = connection;
```



```javascript
module.exports = app => {
    const orderpage = require("../controller/orderpage.controller");

    //Mengambil semua data
    app.get("/api/orderpage", orderpage.findAll);
    //Mengambil data yang memiliki id = id
    app.get("/api/orderpage/:id", orderpage.findOne);
    //Membuat data baru
    app.post("/api/orderpage", orderpage.create);
    //Mengubah data yang memiliki id = id
    app.put("/api/orderpage/:id", orderpage.update);
    //Hapus data yang memiliki id = id
    app.delete("/api/orderpage/:id", orderpage.delete);
    //Hapus seluruh data
    app.delete("/api/orderpage", orderpage.deleteAll);
};
```



```javascript
const express = require("express");
const bodyParser = require("body-parser");
var cors = require("cors")
var path = require('path');
const app = express();
//parse requests - application/json
app.use(bodyParser.json());
//parse requests - application/x-www-form-urlencoded
app.use(bodyParser.urlencoded({ extended: true}));
//cors
var cor = cors();
app.use(cor);
//path
app.use(express.static(path.join(__dirname, "./public")));
app.get("/", (req, res) => {
    res.json({ message: "Selamat datang pada matakuliah pemrograman perangkat bergerak" });
});
require("./app/routes/book.routes") (app);
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
    console.log(`Server is running on port ${PORT}.`);
});
```