

**LAPORAN UKK**  
**APLIKASI KASIR ( PENGOLAHAN BASIS DATA )**



Disusun Oleh :

**Nama : MUHAMMAD HIDAYAT**

**NISN : 0076567353**

**NIS : 22030052**

**REKAYASA PERANGKAT LUNAK  
SEKOLAH MENEGAH KEJURUAN NEGERI 6 KENDARI  
SULAWESI TENGGARA**

**2025**

## **LEMBARAN PENGESAHAN SEKOLAH**

Lembar Pengesahan Ujian Kompetensi Keahlian (UKK) dengan Jurusan Rekayasa Perangkat Lunak dalam membuat Aplikasi Kasir ( Pengolahan Basis Data ) dengan ini telah diperiksa, disetujui dan disahkan oleh Kepala SMKN 6 Kendari pada :

Nama : MUHAMMAD HIDAYAT

NISN : 0076567353

Hari : Senin

Tanggal : 24 Februari 2025

Tempat : SMKN 06 Kendari

Telah diperiksa dan dinyatakan sah oleh :

| No | Nama  | Instansi               | Tanda Tangan | Tanggal |
|----|---|------------------------|--------------|---------|
| 1. | <b>Fandiansyah, S.T</b>   | CV. TECHNO'S STUDIO    |              |         |
| 2. | <b>Firawati Asrul, S.Si., M.Si.<br/>NIP.1981022120102013</b>      | <b>SMK N 6 KENDARI</b> |              |         |
| 3. | <b>Juniarsi Amelia Abas, S.Kom<br/>NIPPPK. 198806122024212042</b> | <b>SMK N 6 KENDARI</b> |              |         |
| 4. | <b>Humairah, S.Kom<br/>NIPPPK. 198905272024212044</b>             | <b>SMK N 6 KENDARI</b> |              |         |
| 5. | <b>Muh. Ryand January, S.Kom</b>                                  | <b>SMK N 6 KENDARI</b> |              |         |
| 6. | <b>Lita Lusiyanti</b>   | <b>SMK N 6 KENDARI</b> |              |         |

**Mengetahui,  
Kepala Sekolah SMK N 6 Kendari,**

**Sitti Ma'wa Rohani, S.Pd  
NIP. 197905172006042029**

## KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Allah SWT atas segala rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan laporan ini dengan baik. Shalawat serta salam senantiasa tercurah kepada Nabi Muhammad SAW, yang telah menjadi teladan bagi umat manusia dalam berbagai aspek kehidupan.

Laporan ini merupakan hasil dari Ujian Kompetensi Kejuruan (UKK) jurusan Rekayasa Perangkat Lunak di SMKN 6 Kendari, yang menyoroti pembuatan aplikasi kasir berbasis pengelolaan basis data. Dalam laporan ini, kami memaparkan gambaran singkat pembuatan aplikasi, fitur-fitur yang kami implementasikan, serta evaluasi kinerja aplikasi.

Kami mengucapkan terima kasih kepada semua pihak yang telah membantu dan mendukung kami selama proses pembuatan aplikasi ini. Semoga aplikasi yang kami hasilkan dapat bermanfaat bagi pengguna dan menjadi langkah awal dalam inovasi pendidikan.

Akhir kata, kami mohon maaf atas segala kekurangan yang terdapat dalam laporan ini. Semoga laporan ini dapat memberikan informasi yang berguna dan menjadi referensi bagi pembaca yang ingin mengembangkan aplikasi serupa di masa depan.

Kendari, 24 Februari 2025

Muhammad Hidayat

## DAFTAR ISI

|   |     |
|---|-----|
| <b>LEMBARAN PENGESAHAN SEKOLAH.....</b>           | ii  |
| <b>KATA PENGANTAR.....</b>                        | iii |
| <b>DAFTAR ISI.....</b>                            | iv  |
| <b>BAB I.....</b>                                 | 1   |
| <b>PENDAHULUAN.....</b>                           | 1   |
| 1.1 <b>Latar Belakang.....</b>                    | 1   |
| 1.2 <b>Rumusan Masalah .....</b>                  | 2   |
| 1.3 <b>Manfaat.....</b>                           | 2   |
| 1.4 <b>Tujuan.....</b>                            | 2   |
| <b>BAB II .....</b>                               | 3   |
| <b>TINJAUAN PUSTAKA.....</b>                      | 3   |
| 2.1 <b>Hypertext Markup Language (HTML) .....</b> | 3   |
| 2.1.1    Pengertian HTML .....                    | 3   |
| 2.1.2    Fungsi HTML .....                        | 3   |
| 2.2 <b>Cascading Style Sheets (CSS) .....</b>     | 4   |
| 2.2.1    Pengertian CSS .....                     | 4   |
| 2.2.2    Fungsi CSS.....                          | 4   |
| 2.3 <b>PHP: Hypertext Preprocessor .....</b>      | 4   |
| 2.3.1    Pengertian PHP .....                     | 4   |
| 2.3.1    Fungsi PHP .....                         | 5   |
| 2.4 <b>Javascript.....</b>                        | 5   |
| 2.4.1    Pengertian Javascript.....               | 5   |
| 2.4.2    Fungsi Javascript.....                   | 5   |
| 2.5 <b>Laragon.....</b>                           | 6   |
| 2.5.1    Pengertian Laragon .....                 | 6   |
| 2.5.2    Komponen Laragon.....                    | 6   |
| 2.6 <b>Nginx .....</b>                            | 7   |
| 2.6.1    Pengertian Nginx.....                    | 7   |
| 2.6.2    Fungsi Nginx.....                        | 7   |
| 2.7 <b>MySQL.....</b>                             | 7   |

|                             |   |    |
|-----------------------------|---|----|
| 2.7.1                       | Pengertian MySQL .....  | 7  |
| 2.7.2                       | Fitur MySQL.....  | 8  |
| <b>2.8</b>                  | <b>Aplikasi Kasir ( Pengolahan Basis Data )</b> .....   | 8  |
| 2.8.1                       | Pengertian Kasir ( Pengolahan Basis Data ) .....  | 8  |
| 2.8.2                       | Fitur Aplikasi Kasir ( Pengolahan Basis Data ) .....  | 8  |
| <b>2.9</b>                  | <b>Laravel</b> .....  | 9  |
| 2.9.1                       | Pengertian Laravel .....  | 9  |
| 2.9.2                       | Fitur Laravel.....  | 9  |
| <b>2.10</b>                 | <b>Metronic</b> .....   | 10 |
| 2.10.1                      | Pengertian Metronic .....   | 10 |
| 2.10.2                      | Fitur Metronic .....  | 10 |
| <b>BAB III</b>              | .....   | 12 |
| <b>METODOLOGI</b>           | .....   | 12 |
| <b>3.1</b>                  | <b>Alat dan Bahan</b> .....   | 12 |
| 3.1.1                       | Alat.....   | 12 |
| 3.1.2                       | Bahan.....  | 13 |
| <b>3.2</b>                  | <b>Prosedur</b> .....   | 13 |
| <b>BAB IV</b>               | .....   | 15 |
| <b>HASIL DAN PEMBAHASAN</b> | .....   | 15 |
| <b>4.1</b>                  | <b>Hasil</b> .....  | 15 |
| 4.1.1                       | Membuat daftar kebutuhan teknis dan spesifikasi perangkat yang dibutuhkan. .....  | 15 |
| 4.1.2                       | Menyiapkan perlatan dan bahan yang diperlukan sesuai dengan daftar peralatan.....   | 17 |
| 4.1.3                       | Menginstal text editor .....  | 19 |
| 4.1.4                       | Menginterpretasikan spesifikasi program yang diberikan .....  | 24 |
| 4.1.5                       | Membuat desain User Interface .....   | 25 |
| 4.1.6                       | Menerapkan pemrograman menggunakan basis Framework Laravel  |    |
|                             | 27  |    |
| 4.1.7                       | Proses Pembuatan Code Program untuk User Interface .....  | 40 |
| 4.1.8                       | Membuat database ukk_kasir yang digunakan untuk menambah, mengedit, menghapus, dan melihat data pelanggan, produk, penjualan, dan detail penjualan di kasir sesuai gambar kerja ..... | 61 |
| 4.1.9                       | Melakukan running dan testing.....  | 64 |

|                             |  |           |
|-----------------------------|--|-----------|
| 4.1.10                      | Buatlah dokumentasi kode program dan upload ke github..... | 66        |
| <b>4.2</b>                  | <b>Pembahasan.....</b>                                     | <b>71</b> |
| <b>BAB V.....</b>           |  | <b>73</b> |
| <b>PENUTUP .....</b>        |  | <b>73</b> |
| <b>5.1</b>                  | <b>Kesimpulan.....</b>                                     | <b>73</b> |
| <b>5.2</b>                  | <b>Saran .....</b>   | <b>73</b> |
| <b>DAFTAR PUSTAKA.....</b>  |  | <b>75</b> |
| <b>TENTANG PENULIS.....</b> |  | <b>76</b> |

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Penyelenggaraan Ujian Kompetensi Keahlian (UKK) oleh sekolah bertujuan untuk mengevaluasi kemampuan praktik siswa dalam bidang kejuruan serta memantau perkembangan mereka selama bersekolah. UKK memerlukan bukti otentik, termasuk kehadiran penguji eksternal yang sesuai dengan bidang keahlian masing-masing, guna memastikan validitas hasil evaluasi. Latar belakang penyusunan laporan UKK ini adalah sebagai bagian dari tugas akhir setelah pelaksanaan UKK. Pada ujian ini, terdapat tiga mata pelajaran produktif yang diujikan, yakni Pemrograman Web dan Perangkat Bergerak, Basis Data, dan Pemrograman Berorientasi Objek, yang dilaksanakan dalam satu hari. Materi yang telah diajarkan sebelumnya menjadi dasar pengujian praktik, di mana beberapa kompetensi dasar harus dikuasai oleh siswa dalam setiap mata pelajaran produktif. UKK memberikan hasil evaluasi yang spesifik, yang akan menjadi bagian dari penilaian akhir sekolah.

Dengan laporan ini, kami bertujuan untuk memberikan gambaran lengkap tentang proses pengembangan aplikasi kasir berbasis web yang dirancang untuk mempermudah pengelolaan transaksi, pelanggan, dan stok produk. Kami berkomitmen untuk menciptakan sebuah sistem kasir yang efisien, mudah digunakan, dan relevan dengan kebutuhan bisnis di era digital saat ini. Semoga laporan ini memberikan wawasan yang bermanfaat bagi pelaku usaha, pengembang teknologi, serta semua pihak yang terlibat dalam optimalisasi sistem manajemen penjualan.

## **1.2 Rumusan Masalah**

Tujuan penyelenggaraan Ujian Kompetensi Keahlian (UKK) ini meliputi :

1. Memfasilitasi siswa untuk mendapatkan sertifikat kompetensi.
2. Memberikan penilaian praktik kompetensi kejuruan kepada siswa yang telah menerima materi produktif selama dua sampai tiga tahun.
3. Memberikan gambaran lengkap tentang proses pengembangan aplikasi kasir.
4. Untuk memberikan penjelasan Langkah-;Langkah pembuatan aplikasi kasir sesuai SOP yang baik dan benar.

## **1.3 Manfaat**

Manfaat penyelenggaraan Ujian Kompetensi Keahlian (UKK) :

1. Mengevaluasi kemampuan praktik siswa dalam bidang kejuruan.
2. Memastikan validitas hasil evaluasi dengan kehadiran penguji eksternal.
3. Meningkatkan kemandirian siswa setelah lulus dari SMK agar dapat bersaing dalam dunia kerja atau melanjutkan pendidikan yang lebih tinggi.

## **1.4 Tujuan**

Tujuan dari Ujian Kompetensi Keahlian (UKK) :

1. Memberikan kesempatan siswa kesempatan untuk memperoleh sertifikat kompetensi.
2. Menilai Kemampuan Praktis untuk mengevaluasi keterampilan siswa.
3. Meningkatkan pemahaman dalam pengembangan aplikasi sesuai soal yang telah diberikan
4. Membimbing siswa dalam implementasi Standar Operasional Prosedur (SOP)

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Hypertext Markup Language (HTML)**

##### **2.1.1 Pengertian HTML**

Hypertext Markup Language (HTML) adalah bahasa markup yang digunakan untuk membuat struktur dan konten halaman web. HTML menggunakan tag yang diapit tanda kurung siku (< dan >) untuk mendefinisikan elemen-elemen pada halaman web, seperti heading, paragraf, gambar, dan tautan. Tag tersebut tidak menentukan tampilan visual halaman web, melainkan fungsinya untuk memberikan struktur dan makna pada konten.

##### **2.1.2 Fungsi HTML**

- **Mendefinisikan Struktur Halaman Web:** HTML menyediakan kerangka dasar untuk membangun halaman web dengan mendefinisikan elemen-elemen yang akan ditampilkan, seperti judul, paragraf, daftar, tabel, formulir, dan lainnya.
- **Menyisipkan Konten:** HTML memungkinkan Anda untuk memasukkan berbagai jenis konten ke dalam halaman web, seperti teks, gambar, audio, video, dan lainnya.
- **Membuat Tautan:** HTML memungkinkan Anda untuk membuat tautan ke halaman web lain atau bagian lain dari halaman web yang sama.

## **2.2 Cascading Style Sheets (CSS)**

### **2.2.1 Pengertian CSS**

Cascading Style Sheets (CSS) adalah bahasa stylesheet yang digunakan untuk menata tampilan visual halaman web yang dibuat dengan HTML. CSS tidak mempengaruhi struktur atau konten halaman web, tetapi memungkinkan Anda untuk mengontrol bagaimana elemen-elemen yang didefinisikan dalam HTML akan ditampilkan di browser web.

### **2.2.2 Fungsi CSS**

- **Menentukan Gaya Elemen:** CSS memungkinkan Anda untuk mengatur gaya elemen HTML, seperti warna font, ukuran font, margin, padding, background, dan lainnya.
- **Menciptakan Layout:** CSS dapat digunakan untuk mengatur layout halaman web, seperti memposisikan elemen, mengatur lebar dan tinggi, dan membuat struktur responsif.
- **Meningkatkan Aksesibilitas:** CSS dapat digunakan untuk meningkatkan aksesibilitas halaman web dengan mengatur kontras warna, ukuran font, dan fokus elemen.

## **2.3 PHP: Hypertext Preprocessor**

### **2.3.1 Pengertian PHP**

PHP (Hypertext Preprocessor) adalah bahasa pemrograman scripting server-side yang digunakan untuk membuat konten web dinamis. Berbeda dengan HTML dan CSS yang berfokus pada struktur dan tampilan visual, PHP dapat menjalankan kode di server web sebelum halaman dikirim ke browser.

### **2.3.1 Fungsi PHP**

- **Menghasilkan Konten Dinamis:** PHP dapat digunakan untuk menghasilkan konten web secara dinamis berdasarkan informasi tertentu, seperti data dari database, input pengguna, atau waktu dan tanggal.
- **Bekerja dengan Database:** PHP dapat terhubung ke database untuk mengambil, memanipulasi, dan menyimpan data.
- **Mengelola Sesi dan Keamanan:** PHP dapat digunakan untuk mengelola sesi pengguna, menangani formulir, dan menerapkan langkah-langkah keamanan dasar.
- **Interaksi dengan Sistem Eksternal:** PHP dapat berinteraksi dengan sistem eksternal, seperti layanan web dan API, untuk memperluas fungsionalitas website.

## **2.4 Javascript**

### **2.4.1 Pengertian Javascript**

JavaScript adalah bahasa pemrograman scripting tingkat tinggi yang dijalankan di browser web. Berbeda dengan PHP yang dieksekusi di server, JavaScript dieksekusi langsung di client-side (perangkat pengguna), sehingga memungkinkan penambahan interaktivitas pada halaman web.

### **2.4.2 Fungsi Javascript**

- **Menambah Interaktivitas:** JavaScript dapat digunakan untuk menanggapi event pengguna seperti klik mouse, gerakan mouse, dan submit formulir. Hal ini memungkinkan creation of dynamic and interactive web pages, for instance, creating image carousels, validating form input, or adding real-time updates.
- **DOM Manipulation:** JavaScript dapat memanipulasi Document Object Model (DOM), yang merupakan representasi berstruktur

dari halaman web. Hal ini memungkinkan **pengubahan konten, gaya, dan struktur halaman web secara dinamis**.

- **Asynchronous Communication:** JavaScript dapat melakukan **komunikasi asynchronous** dengan server web, memungkinkan **pertukaran data tanpa perlu memuat ulang seluruh halaman**.

## 2.5 Laragon

### 2.5.1 Pengertian Laragon

Laragon adalah perangkat lunak web development stack yang gratis dan open-source. Fungsinya untuk menyediakan lingkungan pengembangan web lokal yang lengkap, sehingga developer dapat membangun dan menguji website di komputer mereka sendiri tanpa perlu setting server yang kompleks.

### 2.5.2 Komponen Laragon

- **Apache:** Server web open-source yang berfungsi untuk menjalankan website.
- **MySQL:** Sistem manajemen database (DBMS) open-source yang digunakan untuk menyimpan data website.
- **PHP:** Bahasa pemrograman scripting server-side yang digunakan untuk membuat konten web dinamis.
- **MariaDB:** Cabang (fork) dari MySQL yang dikenal dengan performanya yang baik dan sering dipaketkan dengan Laragon.
- **Xdebug:** Debugger PHP yang membantu developer menemukan dan memperbaiki kesalahan dalam kode PHP mereka.
- **Adminer:** Alat web untuk mengelola database MySQL/MariaDB.
- **Composer:** Package manager untuk PHP yang membantu menginstal dan memperbarui libraries dan framework yang dibutuhkan.

## 2.6 Nginx

### 2.6.1 Pengertian Nginx

Nginx adalah web server open-source dan perangkat lunak reverse proxy yang dikenal dengan efisiensi, skalabilitas, dan keandalannya. Efisiensi sumber daya, skalabilitas, dan keandalannya menjadikannya pilihan yang populer untuk website dengan traffic tinggi dan aplikasi web dinamis.

### 2.6.2 Fungsi Nginx

- **Server Web:** Nginx dapat berfungsi sebagai **server web standalone** untuk **menghidangkan website secara langsung**.
- **Reverse Proxy:** Nginx dapat bertindak sebagai **reverse proxy** untuk **meneruskan permintaan ke server web lain**, seperti Apache atau server aplikasi Node.js, dan **mendistribusikan beban** di antara beberapa server.
- **Static File Serving:** Nginx dapat **menyajikan file statis** dengan cepat dan efisien, seperti file HTML, CSS, JavaScript, dan gambar.
- **Load Balancing:** Nginx dapat **menyeimbangkan beban** di antara beberapa server web belakang (backend) untuk **meningkatkan ketersediaan dan performa website**.

## 2.7 MySQL

### 2.7.1 Pengertian MySQL

MySQL adalah sistem manajemen database (DBMS) open-source yang populer. Ia digunakan untuk menyimpan, mengelola, dan mengambil data yang terstruktur dalam format relational. MySQL banyak digunakan untuk mendukung website dan aplikasi web dinamis.

## 2.7.2 Fitur MySQL

- **Open-source dan Gratis:** Siapa saja dapat menggunakan dan memodifikasi kode sumber MySQL secara gratis.
- **Mudah Digunakan:** MySQL memiliki interface yang user-friendly dan query language (SQL) yang relatif mudah dipelajari.
- **Performa Tinggi:** MySQL dikenal dengan performanya yang baik dan skalabilitasnya yang tinggi, menjadikannya pilihan yang cocok untuk berbagai jenis website dan aplikasi.
- **Keamanan:** MySQL menawarkan fitur keamanan untuk melindungi data, seperti user authentication, authorization, dan encryption.
- **Dukungan Luas:** MySQL didukung oleh berbagai platform dan bahasa pemrograman, membuatnya mudah diintegrasikan dengan berbagai teknologi web lainnya.

## 2.8 Aplikasi Kasir ( Pengolahan Basis Data )

### 2.8.1 Pengertian Kasir ( Pengolahan Basis Data )

Aplikasi kasir atau *Point of Sale* (POS) adalah perangkat lunak yang digunakan untuk memproses transaksi penjualan di tempat usaha. Aplikasi ini menggantikan sistem kasir manual dengan sistem digital yang lebih efisien dan akurat.

Dengan aplikasi POS, Anda dapat melakukan berbagai tugas penting seperti mencatat penjualan, mengelola inventaris, melacak pembayaran, dan bahkan memantau kinerja bisnis secara *real-time*.

### 2.8.2 Fitur Aplikasi Kasir ( Pengolahan Basis Data )

- **Menambahkan Data :** Fitur ini bertujuan untuk memastikan bahwa semua data yang dibutuhkan dalam proses transaksi dapat dengan mudah dimasukkan ke dalam sistem.

- **Mencatat Riwayat Transaksi** : Fitur ini memungkinkan pengguna untuk mencatat dan menyimpan semua riwayat transaksi yang telah terjadi.
- **Mengubah dan Menghapus Data** : Pengguna diberikan opsi untuk mengubah atau menghapus data jika sudah tidak diperlukan atau terjadi kesalahan input.
- **Filter Data** : Fitur ini memungkinkan pengguna untuk memfilter data berdasarkan kriteria tertentu sehingga lebih mudah untuk mencari atau mengelompokkan data yang relevan.

## 2.9 Laravel

### 2.9.1 Pengertian Laravel

Laravel adalah kerangka kerja aplikasi web dengan sintaks yang ekspresif dan elegan. Kerangka kerja web menyediakan struktur dan titik awal untuk membuat aplikasi Anda, sehingga Anda dapat fokus untuk membuat sesuatu yang luar biasa sementara kami mengerjakan detailnya.

Laravel berusaha untuk memberikan pengalaman pengembang yang luar biasa sambil menyediakan fitur-fitur canggih seperti injeksi ketergantungan yang menyeluruh, lapisan abstraksi basis data yang ekspresif, antrian dan pekerjaan terjadwal, pengujian unit dan integrasi, dan banyak lagi.

### 2.9.2 Fitur Laravel

- **Blade Templating Engine** : Blade adalah mesin template Laravel yang memungkinkan pengembang untuk menggunakan sintaksis yang sederhana dan bersih dalam membangun tampilan aplikasi.
- **Artisan Command Line Interface** : Dengan Artisan, pengembang dapat menjalankan berbagai tugas seperti membuat controller, model, migration, dan masih banyak lagi tanpa menulis kode secara manual.

- **Migration & Schema Builder** : Laravel menyediakan sistem migration yang memudahkan pengelolaan skema database. Dengan migration, pengembang dapat membuat, mengubah, atau menghapus tabel dan kolom database menggunakan PHP.
- **Authentication & Authorization** : Laravel menyediakan fitur authentication dan authorization yang sangat lengkap dan mudah diimplementasikan. Pengembang dapat menggunakan sistem login, register, dan pengelolaan pengguna dengan beberapa baris kode.

## 2.10 Metronic

### 2.10.1 Pengertian Metronic

**Metronic** adalah sebuah toolkit komprehensif yang menyediakan berbagai komponen antarmuka pengguna (UI) yang dibangun menggunakan **Tailwind CSS**. Dengan menggunakan Metronic, pengembang dapat dengan mudah membangun aplikasi web modern yang tidak hanya menarik secara visual, tetapi juga dapat diskalakan dengan cepat dan efisien.

Toolkit ini mencakup berbagai elemen desain dan komponen yang dirancang untuk mempercepat proses pengembangan, memberikan kontrol penuh atas tampilan dan fungsionalitas aplikasi, serta memastikan aplikasi dapat berjalan dengan baik di berbagai perangkat dan ukuran layar.

### 2.10.2 Fitur Metronic

- **Desain Responsif dan Modern** : Metronic dirancang dengan antarmuka pengguna (UI) yang bersih, modern, dan responsif. Template ini dapat beradaptasi dengan berbagai ukuran layar, mulai dari desktop hingga perangkat mobile, memastikan pengalaman pengguna yang optimal.
- **Komponen UI yang Lengkap** : Metronic menyediakan berbagai elemen dan komponen siap pakai yang dapat digunakan langsung dalam proyek.

- **Sistem Layout yang Fleksibel** : Metronic menyediakan berbagai opsi layout yang bisa disesuaikan.
- **Performa Tinggi dan Optimasi** : Dikembangkan dengan kode yang bersih dan ringan, Metronic dirancang agar cepat dalam memuat halaman dan tidak membebani server.

## **BAB III**

### **METODOLOGI**

#### **3.1 Alat dan Bahan**

##### **3.1.1 Alat**

| No. | Nama Alat dan Bahan                                  | Spesifikasi  | Jumlah | Keterangan |
|-----|--|--|--------|------------|
| 1   | 2  | 3  | 4      | 5          |
| 1.  | Komputer (PC/Laptop) yang difungsikan sebagai server | Prosesor : Dual Core 2,4 GHz (2,93GHz untuk yang memilih <i>platform mobile</i> )<br><br>RAM : 2 GB (4GB untuk yang memilih <i>platform mobile</i> )<br><br>Keyboard<br><br>Mouse<br><br>Monitor | 1 Unit | Baik       |
| 2.  | Komputer (PC/Laptop) yang difungsikan sebagai client | Prosesor : Dual Core 2,4 GHz (2,93GHz untuk yang memilih <i>platform mobile</i> )<br><br>RAM : 2 GB (4GB untuk yang memilih <i>platform mobile</i> )<br><br>Keyboard<br><br>Mouse<br><br>Monitor | 1 unit | Baik       |
| 3.  | Smart Phone  | Android / IOS  | 1 unit | Baik       |
| 4.  | Koneksi Internet                                     | Minimal 1 Mbps   | 1 unit | Baik       |

### **3.1.2      Bahan**

| No. | Bahan           | Nama               |
|-----|-----------------|--------------------|
| 1.  | IDE/Text Editor | Visual Studio Code |
| 2.  | Local Server    | Laragon            |
| 3.  | Browser         | Chrome             |

### **3.2 Prosedur**

1. Membuat daftar kebutuhan teknis dan spesifikasi perangkat yang dibutuhkan pada lembar yang disediakan.
2. Menyiapkan perlatan dan bahan yang diperlukan sesuai dengan daftar peralatan.
3. Identifikasi Tugas yang telah diberikan dengan seksama.

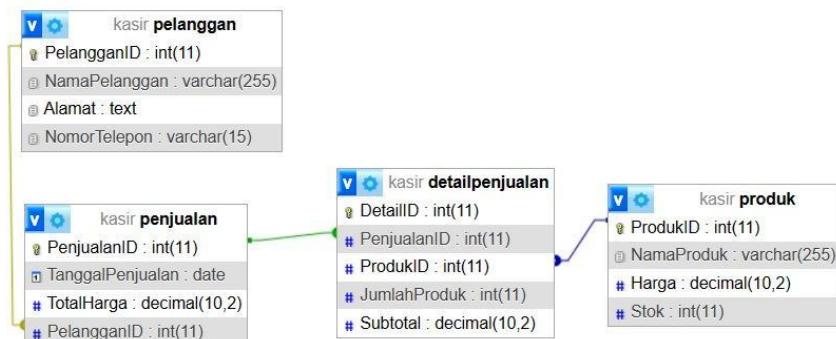
Judul Tugas : Pengelolaan Basis Data

Skenario :

Anda seorang *database administrator* yang diminta oleh sebuah perusahaan untuk mengelola data transaksi, pelanggan, dan stok produk. Anda diminta untuk membuat basis data yang mencakup entitas Penjualan, Pelanggan, Produk, dan Detail Penjualan, memastikan sistem mampu mencatat semua transaksi secara akurat. Anda membuat tabel-tabel, menentukan hubungan antar entitas, dan mengimplementasikan basis data menggunakan langkah-langkah terstruktur. Hasilnya, perusahaan tersebut memiliki sistem yang efektif untuk memantau penjualan, mengelola data pelanggan, dan mengontrol stok produk. Anda diminta untuk melakukan langkah kerja seperti dibawah ini:

- 1). Analisis Kebutuhan: Identifikasi informasi yang perlu disimpan untuk Penjualan, Pelanggan, Produk, dan Detail Penjualan.
- 2). Perancangan Skema: Buat diagram entitas-relasi (ERD) untuk menentukan tabel, atribut, dan relasi antar entitas.
- 3). Pembuatan Tabel: Implementasikan tabel untuk setiap entitas (Penjualan, Pelanggan, Produk, Detail Penjualan) sesuai dengan skema.

- 4). Definisi Relasi: Tentukan relasi antar tabel dengan kunci utama (*Primary Key*) dan kunci asing (*Foreign Key*).
- 5). Input Data: Masukkan data sampel ke dalam tabel untuk pengujian.
- 6). Pengujian *Query*: Uji *query* untuk fungsi seperti pencatatan transaksi, pencarian pelanggan, dan pengecekan stok.
- 7). Optimasi: Pastikan indeksasi dan struktur tabel mendukung efisiensi pencarian data.
- 8). Dokumentasi: Buat dokumentasi sistem untuk penggunaan dan pemeliharaan basis data.
4. Menginstall Web Server, IDE, Browser dan Software pendukung lainnya yang digunakan untuk memprogram aplikasi kasir ( Pengolahan Basis Data )
5. Mengimplementasikan Skema Database dan menetukan relasi antar table.



5. Membuat desain User Interface
  1. Membuat desain dengan memperhatikan kaidah UI/UX
  2. Menerapkan pemrograman menggunakan basis Framework Laravel
  4. Membuat fitur yang digunakan untuk menambah, mengedit, dan menghapus data pelanggan, produk, penjualan, dan detail penjualan di aplikasi kasir sesuai gambar kerja.
  5. Running dan testing hasil program.
  6. Membuat dokumentasi kode program dan mengupload portofolio pada github.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Hasil**

##### **4.1.1 Membuat daftar kebutuhan teknis dan spesifikasi perangkat yang dibutuhkan.**

- Kebutuhan Teknis

| No. | Kebutuhan<br>Teknis       | Bahan /<br><i>Software</i> | Fungsi  |
|-----|---------------------------|----------------------------|---|
| 1.  | Bahasa<br>Pemrograman     | PHP                        | Memberikan instruksi kepada komputer agar dapat memproses dan menampilkan konten web dengan cara yang diinginkan  |
| 2.  | Penyedia<br>Server Lokal  | Laragon                    | Menyediakan lingkungan pengembangan web lokal yang lengkap dengan server, database, dan alat pengembangan         |
| 3.  | Web Server                | Nginx                      | Melayani permintaan HTTP dari klien dan menyampaikan halaman web serta konten statis atau dinamis kepada pengguna |
| 4.  | DBMS                      | MySQL                      | Mengelola, menyimpan, dan mengatur akses terhadap data dalam basis data   |
| 5.  | Browser                   | Chrome                     | Mengakses, menampilkan, dan menguji tampilan serta fungsionalitas dari situs web                                  |
| 6.  | Integrated<br>Development | Visual<br>Studio Code      | Untuk menulis, mengedit, dan mengelola kode dalam pengembangan web  |

|     | Environment<br>(IDE) |   |  |
|-----|----------------------|---|--|
| 7.  | Version Control      | Git   | Melacak perubahan dalam kode sumber dan berkas proyek pengembangan web   |
| 8.  | Dokumentasi          | Dokumentasi kode,<br>petunjuk instalasi,<br>panduan pengguna. | Memberikan panduan yang jelas kepada pengguna tentang cara menggunakan aplikasi web dengan benar dan efisien.  |
| 9.  | Framework            | Laravel   | erangka kerja aplikasi web dengan sintaks yang ekspresif dan elegan. Kerangka kerja web menyediakan struktur dan titik awal untuk membuat aplikasi Anda, sehingga Anda dapat fokus untuk membuat sesuatu yang luar biasa sementara kami mengerjakan detailnya.                     |
| 10. | Template             | Metronic  | Sebuah template atau tema premium yang sangat populer dan digunakan dalam pembuatan aplikasi web dan dashboard. Template ini dirancang untuk mempermudah pengembangan antarmuka pengguna (UI) dengan menyediakan berbagai elemen desain, komponen, dan layout yang siap digunakan. |

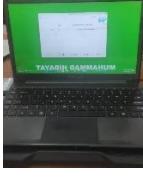
- Spesifikasi Perangkat

| No. | Perangkat        | Spesifikasi  |
|-----|------------------|--|
| 1.  | Laptop           | Prosesor : AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz<br><br>RAM: 8 GB |
| 2.  | OS               | Windows 11   |
| 3.  | Smart Phone      | Prosesor: Octa-core Max 2.30 GHz<br><br>Ram: 8 GB                                  |
| 4.  | Koneksi Internet | 50 Mbps  |

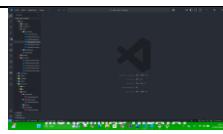
#### 4.1.2 Menyiapkan perlatan dan bahan yang diperlukan sesuai dengan daftar peralatan

- Alat

| No. | Nama Alat                              | Spesifikasi   | Jumlah | Keterangan | Fungsi   | Gambar  |
|-----|--|---|--------|------------|--|---|
| 1   | 2                                      | 3   | 4      | 5          |  |   |
| 1.  | Laptop yang difungsikan sebagai server | Prosesor : 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 3.00 GHz<br><br>RAM: 8 GB | 1 Unit | Baik       | Laptop digunakan untuk menyediakan layanan atau sumber daya kepada perangkat lain di jaringan. |  |

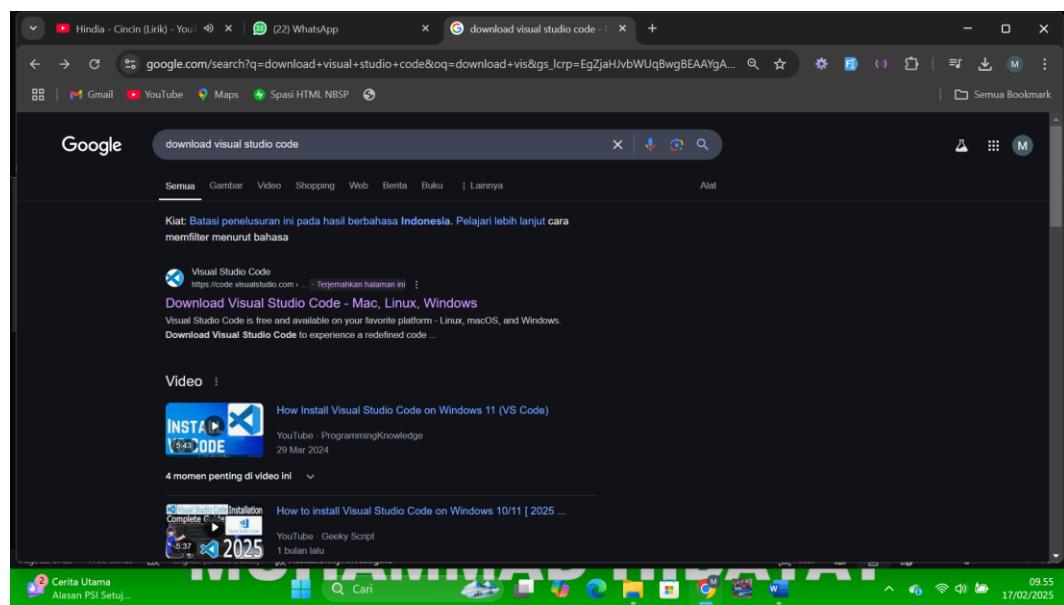
| No. | Nama Alat                              | Spesifikasi   | Jumlah | Keterangan | Fungsi   | Gambar  |
|-----|--|---|--------|------------|--|---|
| 1   | 2                                      | 3   | 4      | 5          |  |   |
| 2.  | Laptop yang difungsikan sebagai client | Prosesor : 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 3.00 GHz<br><br>RAM: 8 GB | 1 unit | Baik       | laptop menggunakan layanan yang disediakan oleh server lain di jaringan. |    |
| 3.  | Smart Phone                            | Android   | 1 unit | Baik       | Menguji responsifitas aplikasi web.                                      |    |
| 4.  | Koneksi Internet                       | 50 Mbps   | 1 unit | Baik       | Menyediakan akses ke internet.   |  |

- Bahan

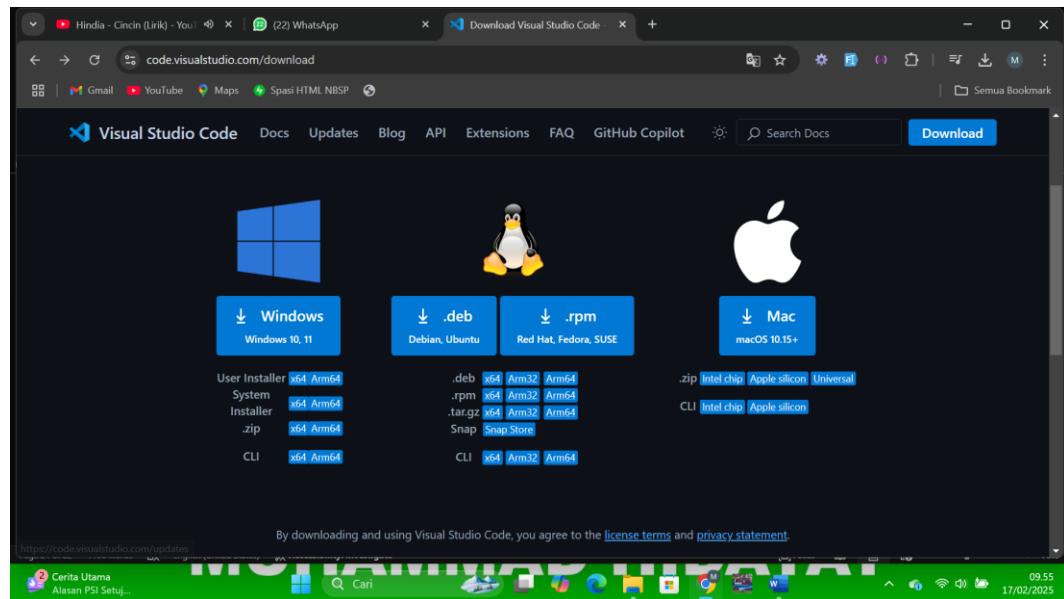
| No. | Nama Bahan                           | Keterangan | Gambar   | Fungsi   |
|-----|--------------------------------------|------------|--|--|
| 1.  | Visual Studio Code (IDE/text editor) | Siap       |  | Perangkat lunak untuk membuat, mengedit, dan memformat teks. |
| 2.  | Larago (Local Server)                | Siap       |  | Mengakses dan menampilkan konten web dari internet.          |
| 3.  | Chrome (Browser)                     | Siap       |  | Mengakses dan menampilkan konten web dari internet.          |

### 4.1.3 Menginstal text editor

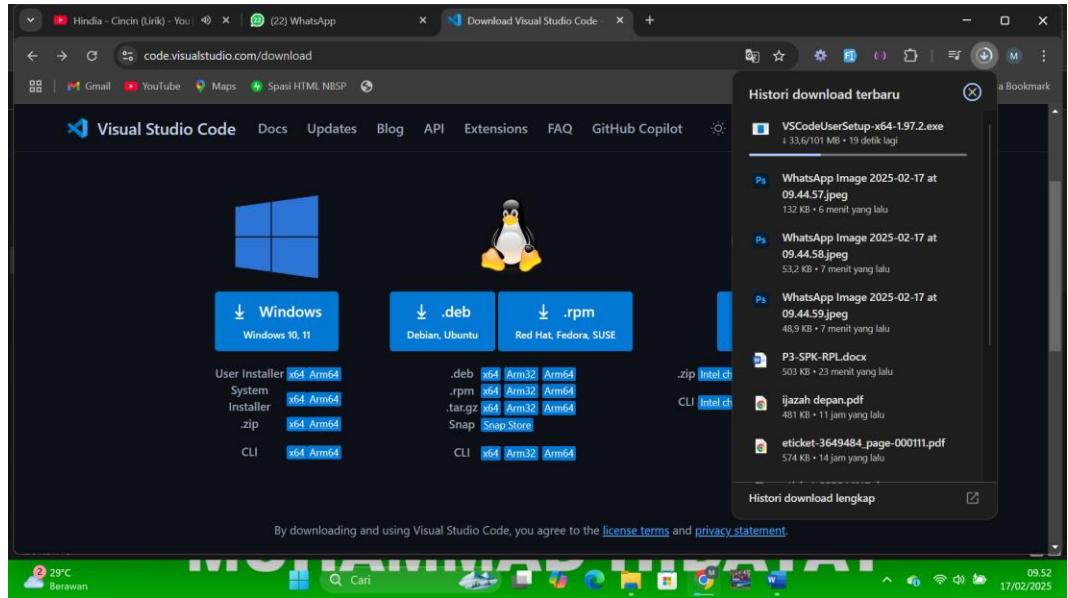
- Buka Browser (Chrome), dan ketik di *search bar* “download vscode”.



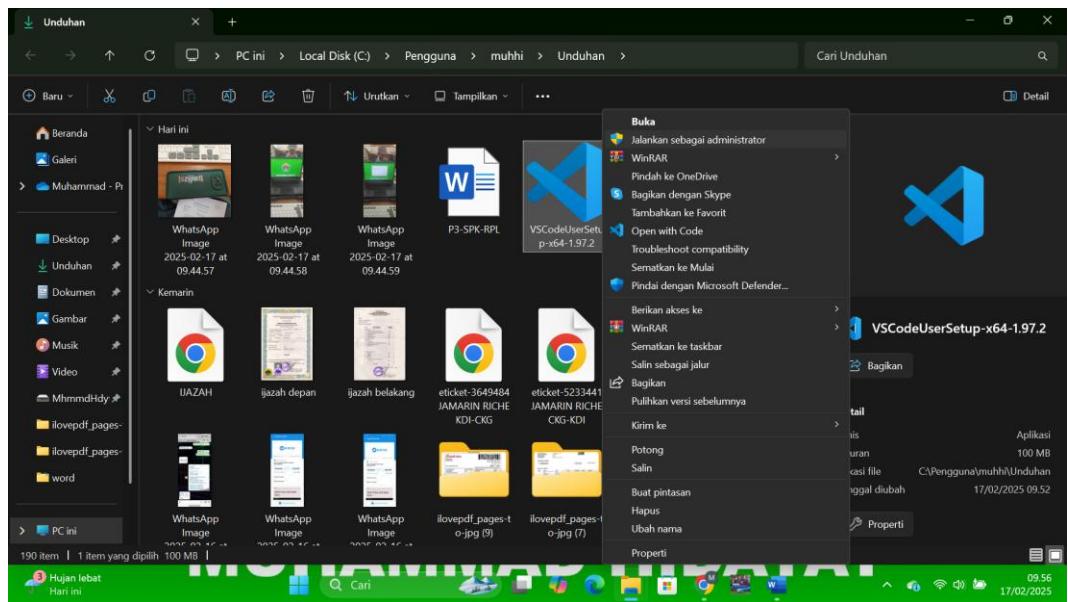
- Masuk ke website VSCode, di halaman download klik di bagian Windows 10, 11.



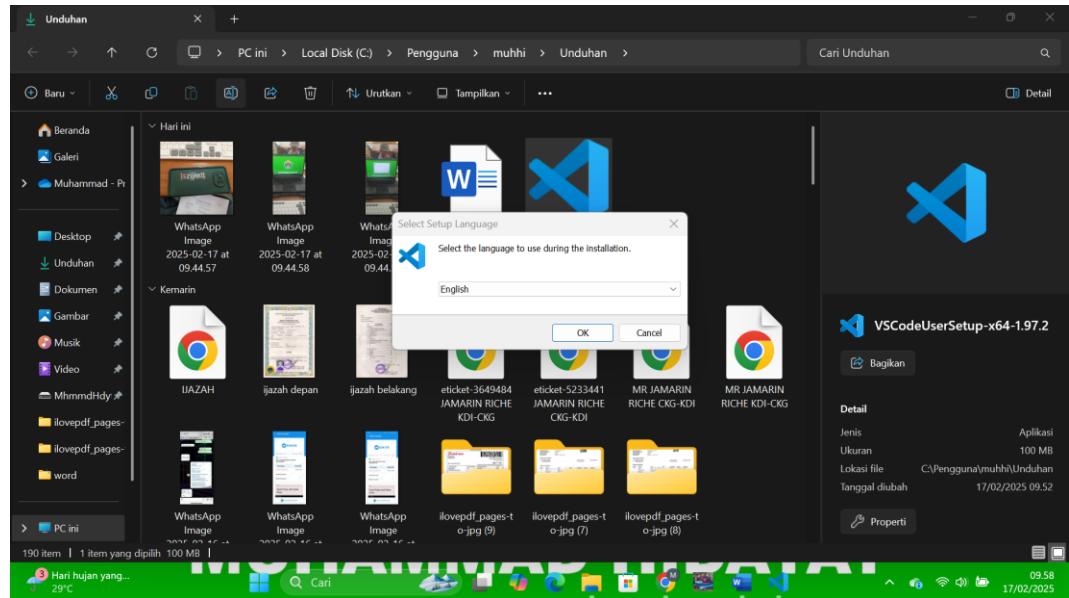
- Tunggu hingga proses download selesai.



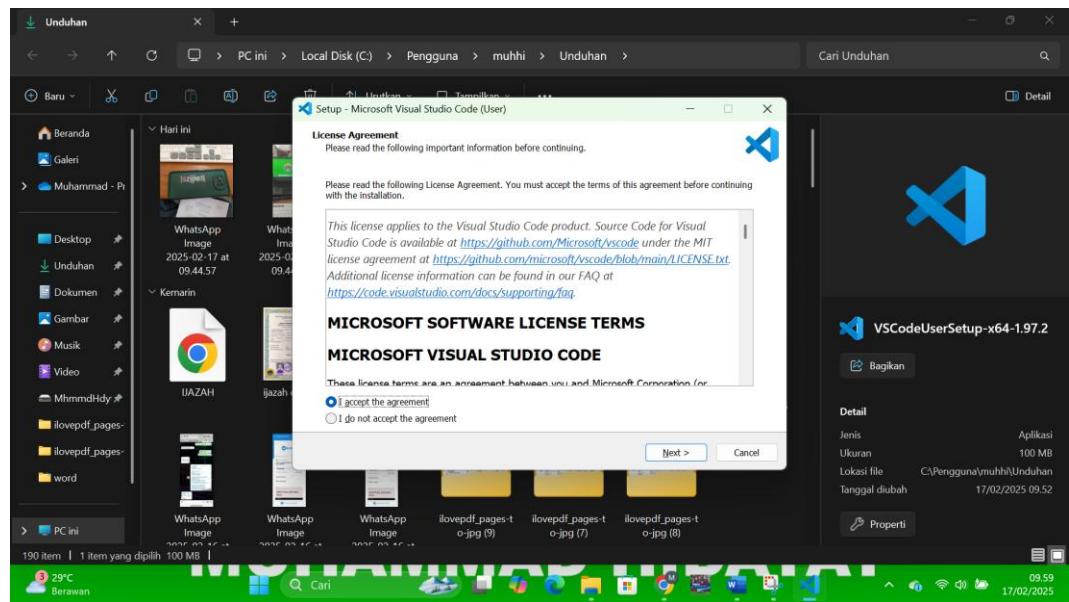
- Selanjutnya lakukan proses instalasi, klik installer VSCode.



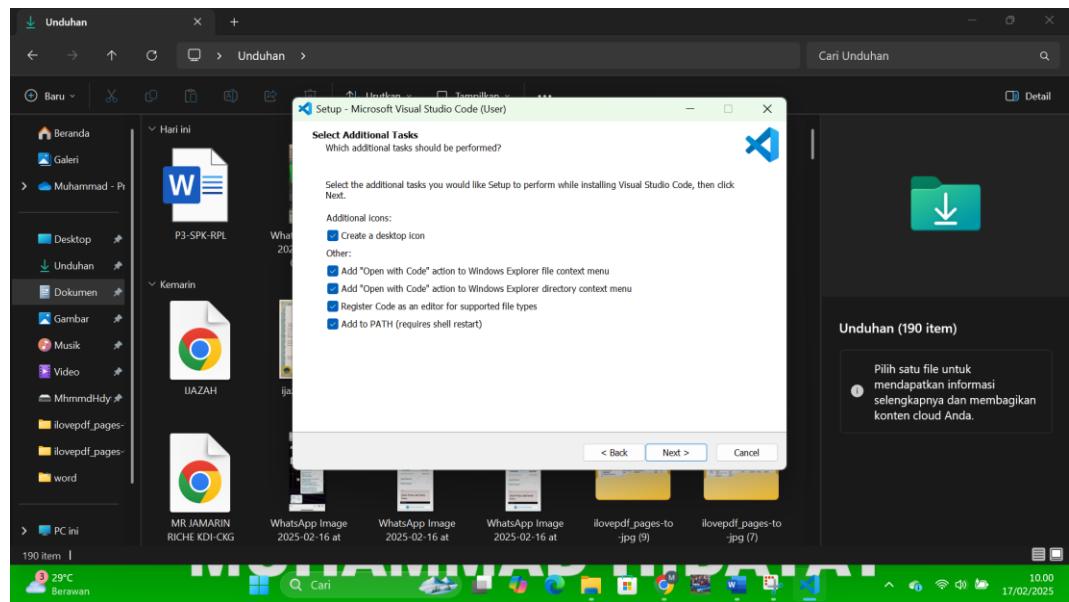
- Klik ‘Yes’ untuk mengizinkan penginstallan VSCode.



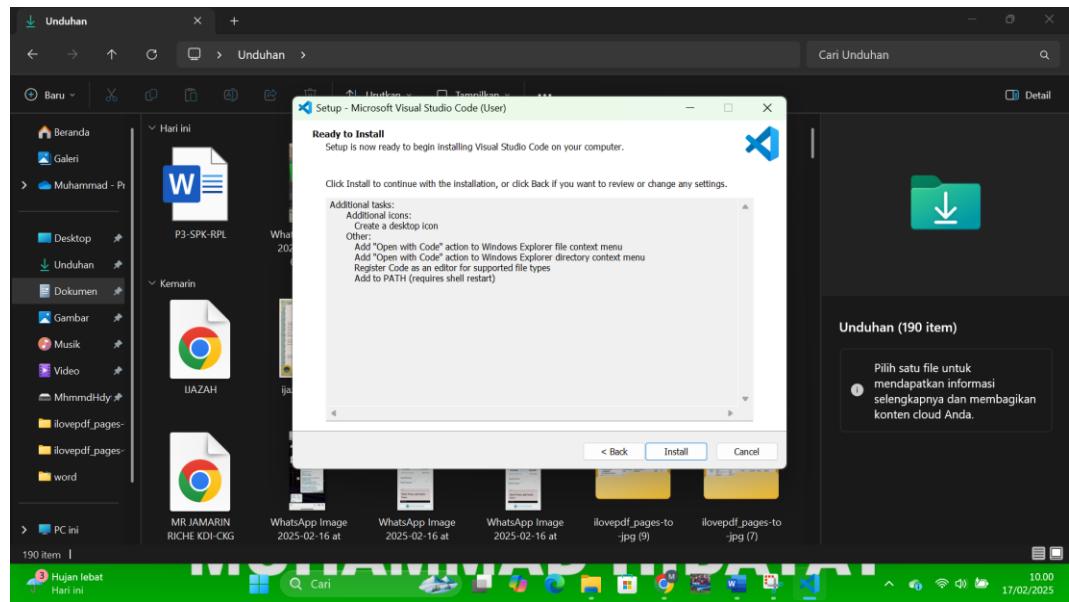
- Pilih opsi “I accept the agreement” lalu klik next.



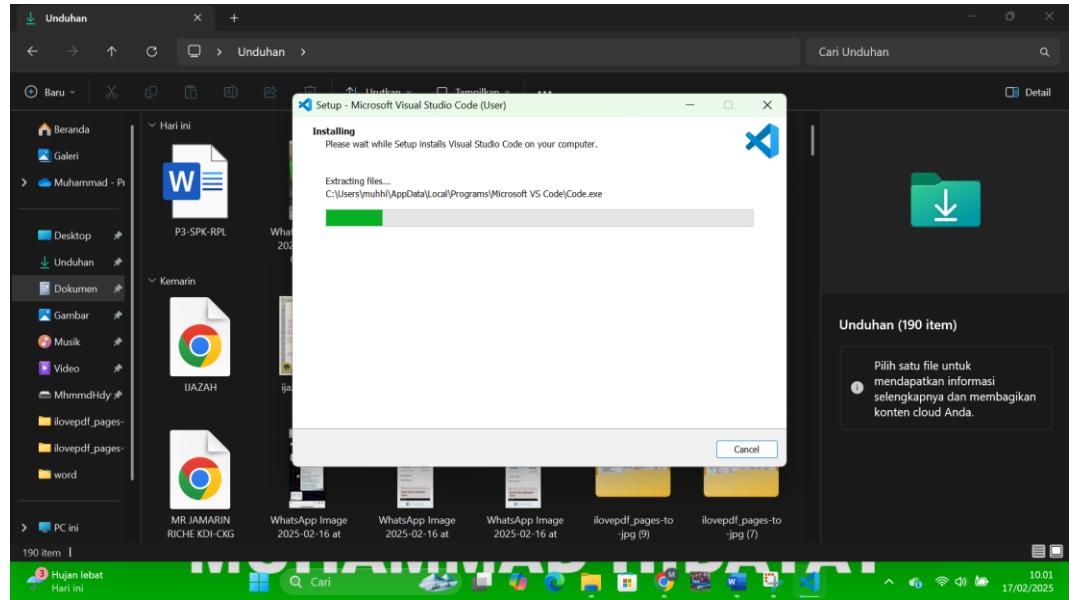
- Centang semua opsi lalu klik next.



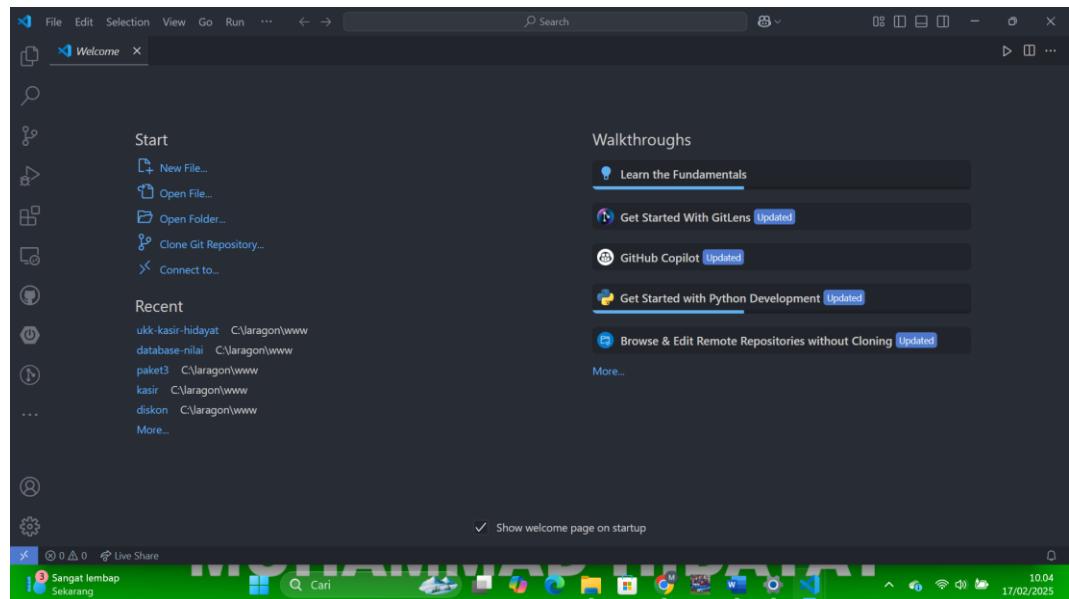
- Klik Install.



- Tunggu hingga proses instalasi selesai



- Instalasi selesai, aplikasi VSCode dapat digunakan.

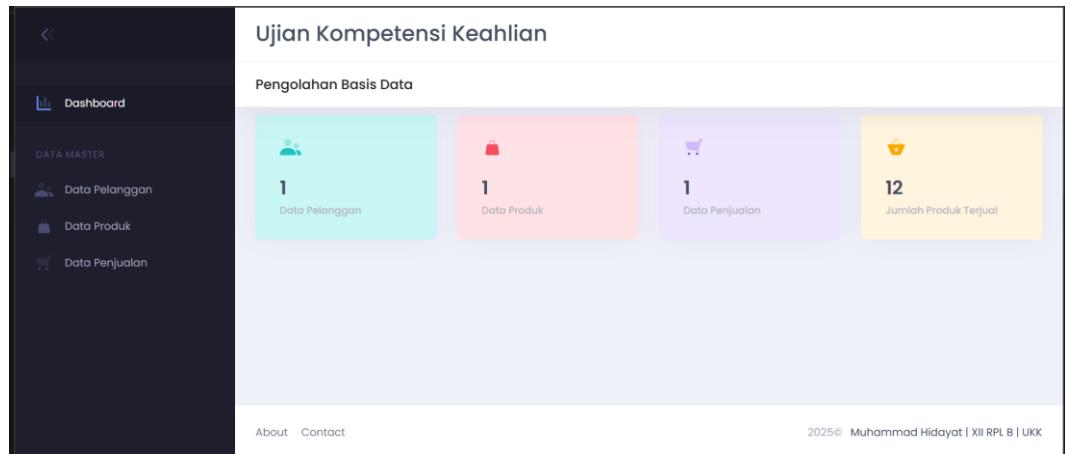


#### **4.1.4 Menginterpretasikan spesifikasi program yang diberikan**

| No | Spesifikasi Program                            | Deskripsi  |
|----|--|--|
| 1. | Manajemen Data Pelanggan                       | Fitur ini memungkinkan pengguna untuk menambahkan, mengedit, menghapus, dan melihat daftar pelanggan.  |
| 2. | Daftar Data Produk                             | Menyediakan daftar produk yang tersedia dalam sistem, termasuk informasi seperti nama produk, kategori, harga, stok yang tersedia, dan deskripsi produk. |
| 3. | Pencatatan Transaksi Penjualan                 | Memungkinkan pengguna untuk mencatat setiap transaksi penjualan yang terjadi.  |
| 4. | Riwayat Detail Penjualan                       | Menampilkan daftar transaksi yang telah dilakukan.   |
| 5. | Pencarian Data Penjualan Berdasarkan Tanggal   | Memungkinkan pengguna untuk mencari dan menampilkan transaksi penjualan berdasarkan rentang tanggal tertentu.  |
| 6. | Pencarian Data Penjualan Berdasarkan Pelanggan | Memudahkan pencarian riwayat transaksi berdasarkan pelanggan tertentu.   |

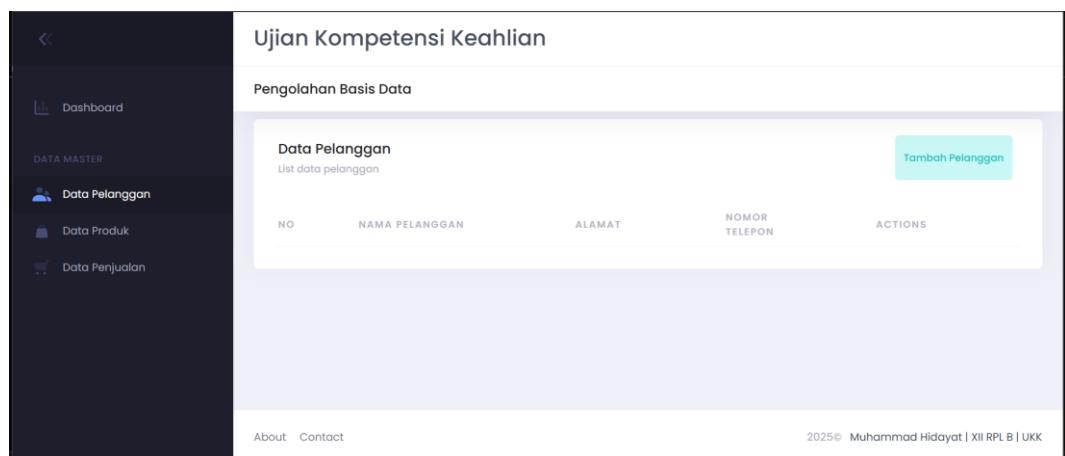
#### 4.1.5 Membuat desain User Interface

- Halaman dashboard



Desain dashboard yang menjadi halaman utama dari semua data.

- Halaman Pelanggan



Desain halaman pelanggan yang dapat dilihat semua user

- Halaman Detail Penjualan

The screenshot shows a user interface for a sales application. On the left is a dark sidebar menu with options: Dashboard, DATA MASTER (Data Pelanggan, Data Produk), and Data Penjualan. The main content area has a title "Ujian Kompetensi Keahlian" and a sub-section "Pengolahan Basis Data". Below this is a section titled "Detail Penjualan 1" showing purchase details: Nama Pelanggan: Muhammad Hidayat, Tanggal: 2025-02-17, Total Harga: Rp 0. A table lists "Produk yang Terjual" with columns: Nama Produk, Jumlah Produk, Harga Satuan, Subtotal, and Aksi. Below the table is a "Tambah Produk" section with fields for "Produk" (Quos quis rerum erro (Stok: 2)) and "Jumlah".

Desain halaman detail penjualan yang akan menampilkan semua data yang terkait dengan penjualan tertentu

#### 4.1.6 Menerapkan pemrograman menggunakan basis Framework Laravel

```

APP_NAME=Kasir
APP_ENV=local
APP_KEY=bax64:n0mdDUG//rjRLVx1oMh8CZ0ErHfXtFaximlhPpAgFE
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT:3306
DB_DATABASE=ukk-kasir
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=smtp
MAIL_HOST=mailpit
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null

```

Gambar 1.1.1 Code program .env untuk laravel

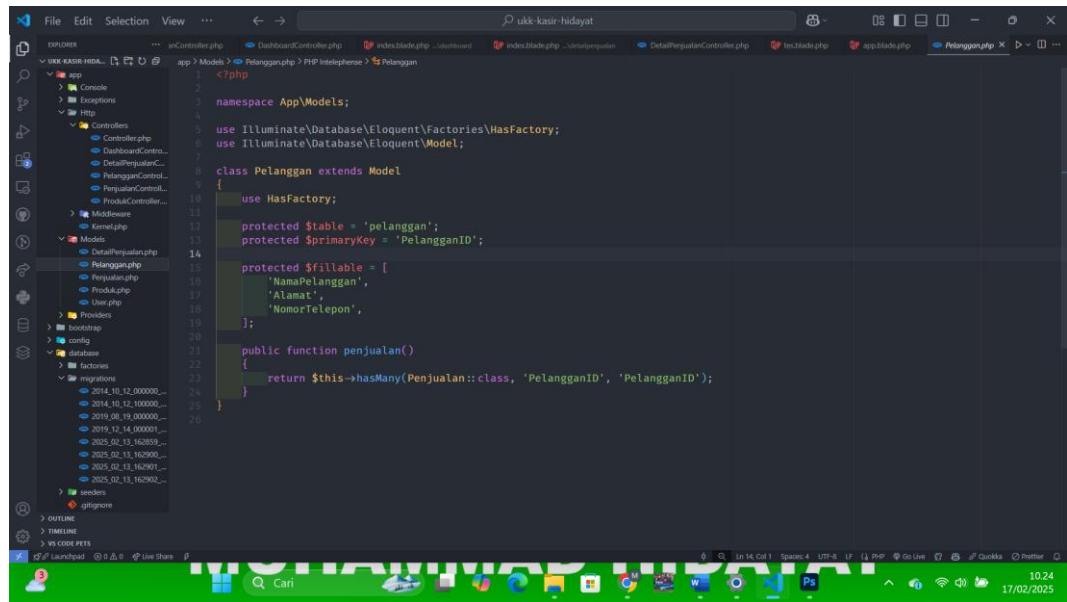
```

use App\Http\Controllers\DashboardController;
use App\Http\Controllers\DetailPenjualanController;
use App\Http\Controllers\PelangganController;
use App\Http\Controllers\PenjualanController;
use App\Http\Controllers\ProdukController;
use App\Models\DetailPenjualan;
use Illuminate\Support\Facades\Route;
use SebastianBergmann\CodeCoverage\Report\Html\Dashboard;

Route::get('/', [DashboardController::class, 'index']);
Route::resource('pelanggan', PelangganController::class);
Route::resource('produk', ProdukController::class);
Route::resource('penjualan', PenjualanController::class);
Route::resource('new:detail-penjualan', controller: DetailPenjualanController::class);

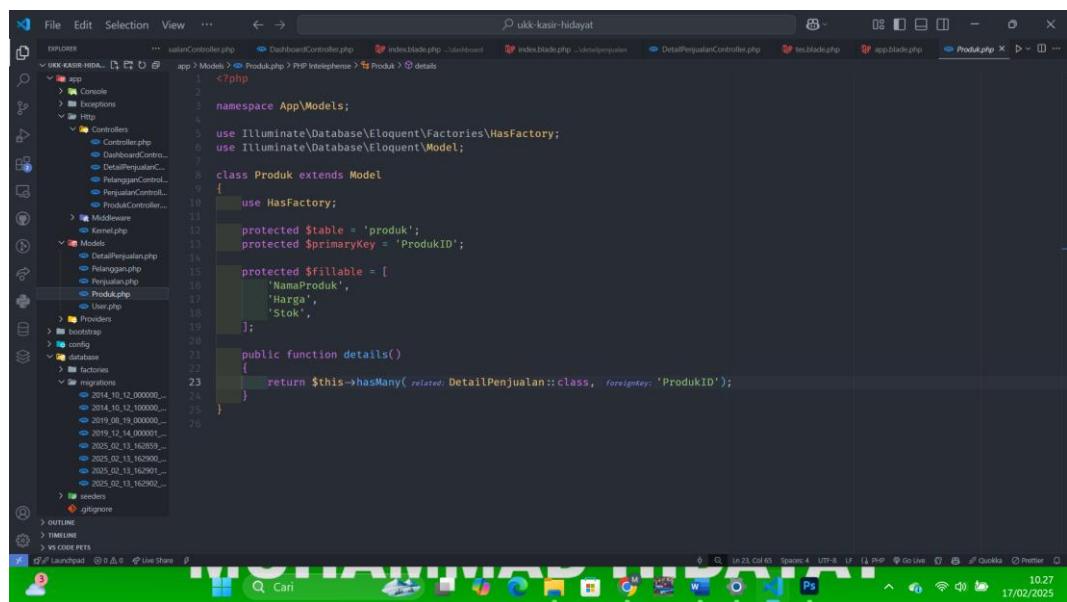
```

Gambar 1.1.2 Code program web.php untuk routing



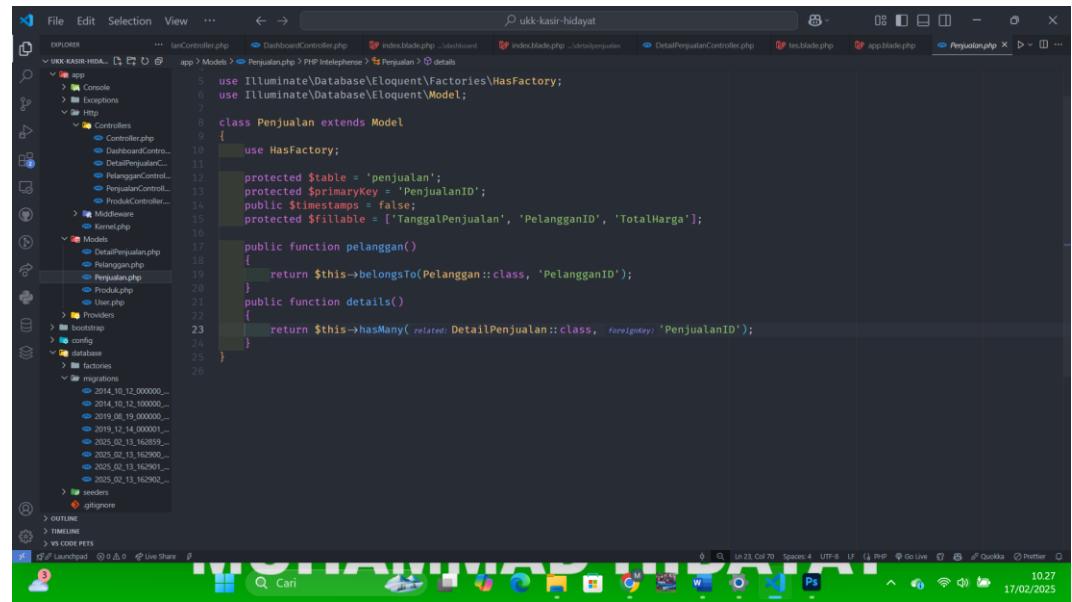
```
<?php  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
  
class Pelanggan extends Model  
{  
    use HasFactory;  
  
    protected $table = 'pelanggan';  
    protected $primaryKey = 'PelangganID';  
  
    protected $fillable = [  
        'NamaPelanggan',  
        'Alamat',  
        'NomorTelepon',  
    ];  
  
    public function penjualan()  
    {  
        return $this->hasMany(Penjualan::class, 'PelangganID', 'PelangganID');  
    }  
}  
;
```

Gambar 1.1.3 Code program model pelanggan.php



```
<?php  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
  
class Produk extends Model  
{  
    use HasFactory;  
  
    protected $table = 'produk';  
    protected $primaryKey = 'ProdukID';  
  
    protected $fillable = [  
        'NamaProduk',  
        'Harga',  
        'Stok',  
    ];  
  
    public function details()  
    {  
        return $this->hasMany(related: DetailPenjualan::class, foreignKey: 'ProdukID');  
    }  
}  
;
```

Gambar 1.1.4 Code program Laravel model produk.php



```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

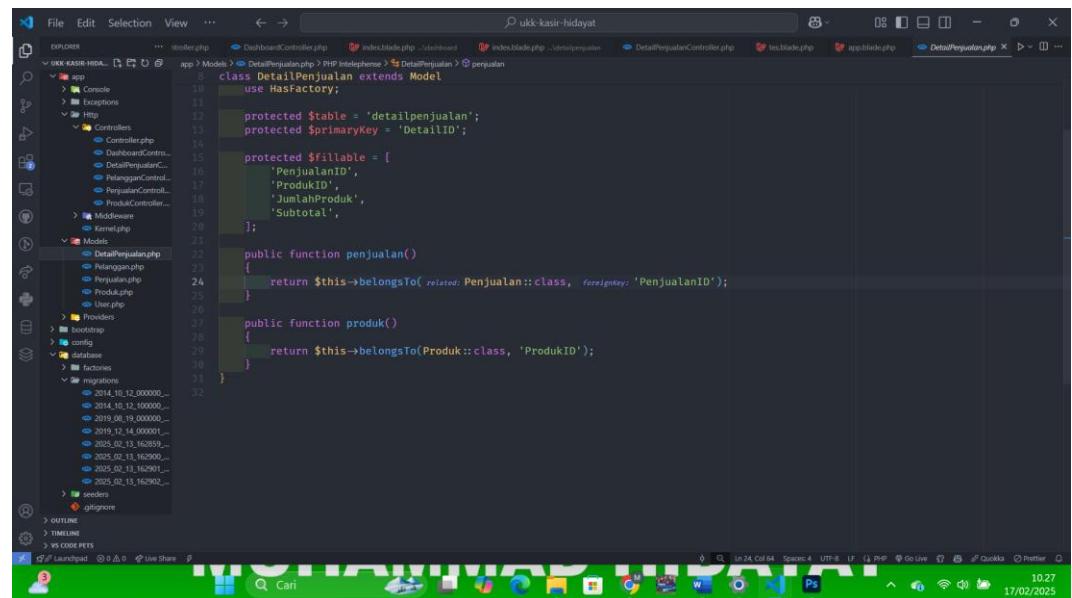
class Penjualan extends Model
{
    use HasFactory;

    protected $table = 'penjualan';
    protected $primaryKey = 'PenjualanID';
    public $timestamps = false;
    protected $fillable = ['TanggalPenjualan', 'PelangganID', 'TotalHarga'];

    public function pelanggan()
    {
        return $this->belongsTo(Pelanggan::class, 'PelangganID');
    }

    public function details()
    {
        return $this->hasMany(DetailPenjualan::class, 'foreignkey: 'PenjualanID');
    }
}
```

Gambar 1.1.5 Code program Laravel model penjualan.php



```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class DetailPenjualan extends Model
{
    use HasFactory;

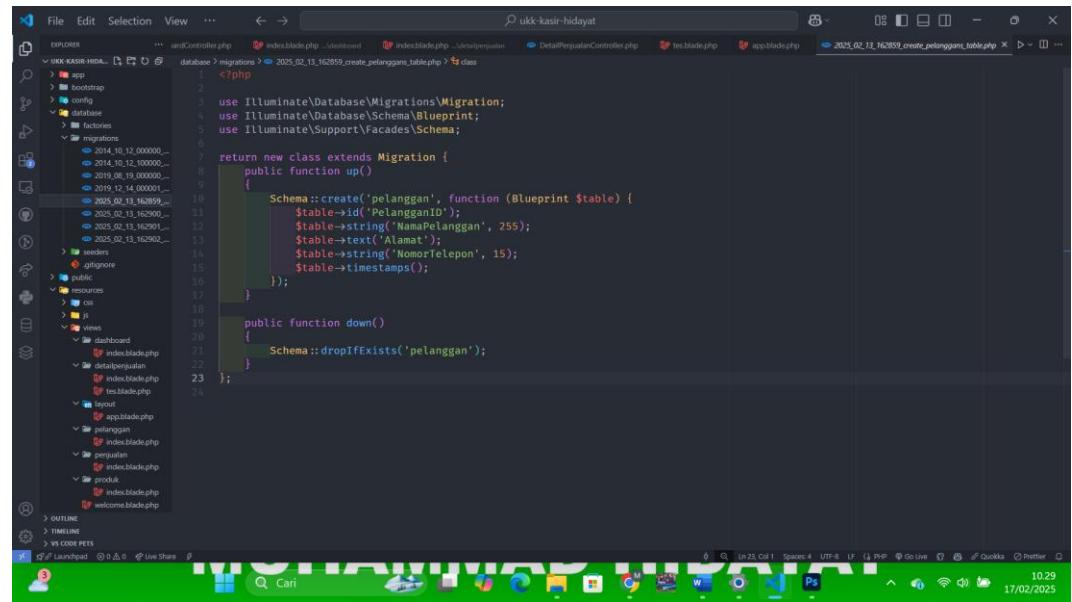
    protected $table = 'detailpenjualan';
    protected $primaryKey = 'DetailID';

    protected $fillable = [
        'PenjualanID',
        'ProdukID',
        'JumlahProduk',
        'Subtotal',
    ];

    public function penjualan()
    {
        return $this->belongsTo(Penjualan::class, 'foreignkey: 'PenjualanID');
    }

    public function produk()
    {
        return $this->belongsTo(Produk::class, 'ProdukID');
    }
}
```

Gambar 1.1.6 Code program Laravel model detailpenjualan.php



The screenshot shows the Visual Studio Code interface with the file `2025_02_13_162859_create_pelanggan_table.php` open in the editor. The code defines a database migration for creating a `'pelanggan'` table. It uses the `Blueprint` class to define columns for `PelangganID` (string, 255), `NamaPelanggan` (text), `Alamat` (text), and `NomorTelepon` (string, 15). The migration also includes timestamps.

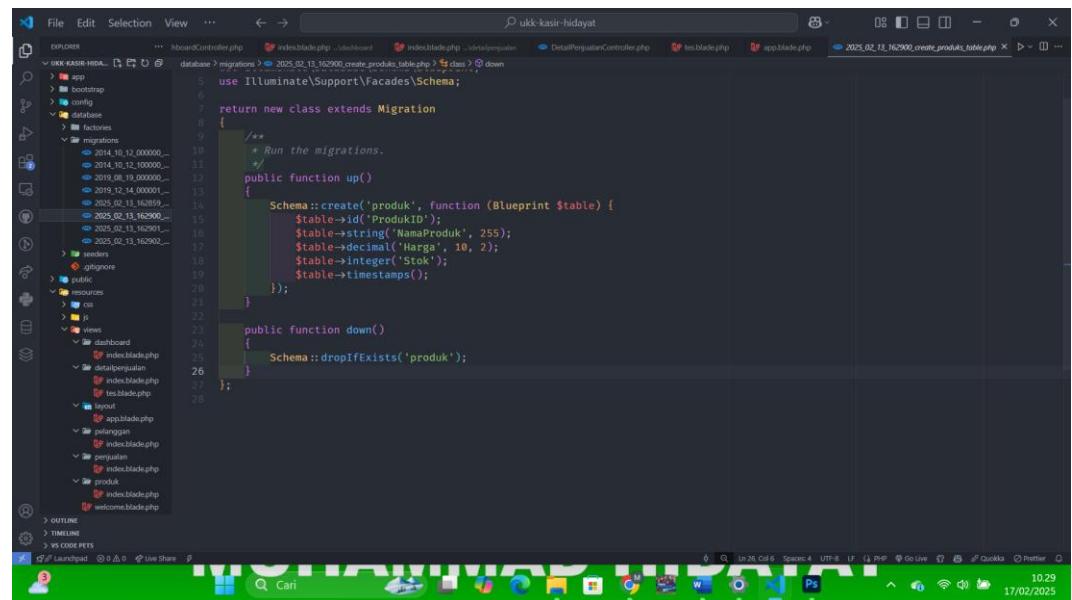
```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration {
    public function up()
    {
        Schema::create('pelanggan', function (Blueprint $table) {
            $table->id('PelangganID');
            $table->string('NamaPelanggan', 255);
            $table->text('Alamat');
            $table->string('NomorTelepon', 15);
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('pelanggan');
    }
};
```

Gambar 1.2.3 Code program Laravel migration data pelanggan



The screenshot shows the Visual Studio Code interface with the file `2025_02_13_162900_create_produk_table.php` open in the editor. The code defines a database migration for creating a `'produk'` table. It uses the `Blueprint` class to define columns for `ProdukID` (string, 255), `NamaProduk` (text), `Harga` (decimal, 10, 2), and `Stok` (integer). The migration also includes timestamps.

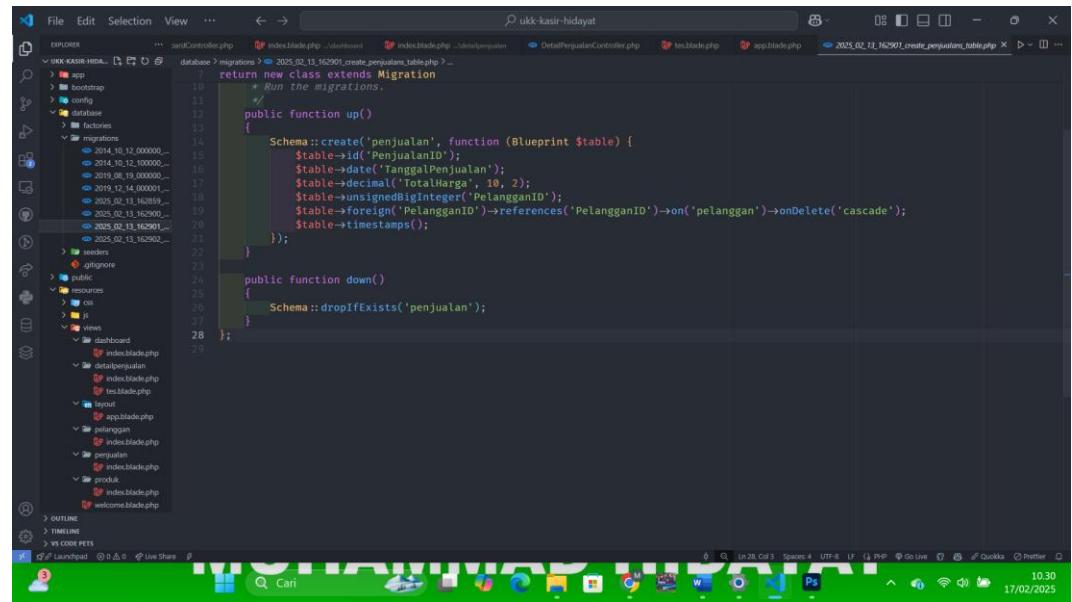
```
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Migrations\Migration;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up()
    {
        Schema::create('produk', function (Blueprint $table) {
            $table->id('ProdukID');
            $table->string('NamaProduk', 255);
            $table->decimal('Harga', 10, 2);
            $table->integer('Stok');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('produk');
    }
};
```

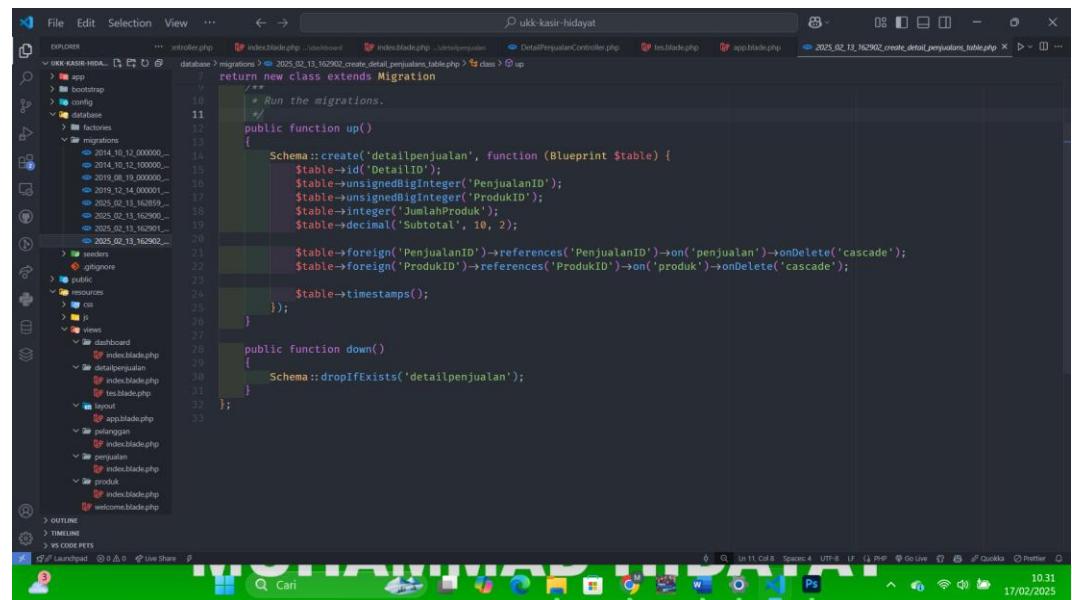
Gambar 1.2.4 Code program laravel migration data produk



```
File Edit Selection View ... < - > | ukk-kasir-hidayat
EXPLORER ... searchController.php index.blade.php ..._viewdropdown index.blade.php ..._viewdropdownpenjualan DetailPenjualanController.php tes.blade.php app.blade.php 2025_02_13_162901_create_penjualan_table.php ...
database > migrations > 2025_02_13_162901_create_penjualan_table.php ...
return new class extends Migration
{
    * Run the migrations.
    */
    public function up()
    {
        Schema::create('penjualan', function (Blueprint $table) {
            $table->id('PenjualanID');
            $table->date('TanggalPenjualan');
            $table->decimal('TotalHarga', 10, 2);
            $table->unsignedBigInteger('PelangganID');
            $table->foreign('PelangganID')->references('PelangganID')->onDelete('cascade');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down()
    {
        Schema::dropIfExists('penjualan');
    }
};
```

Gambar 1.2.5 Code program Laravel migration data penjualan

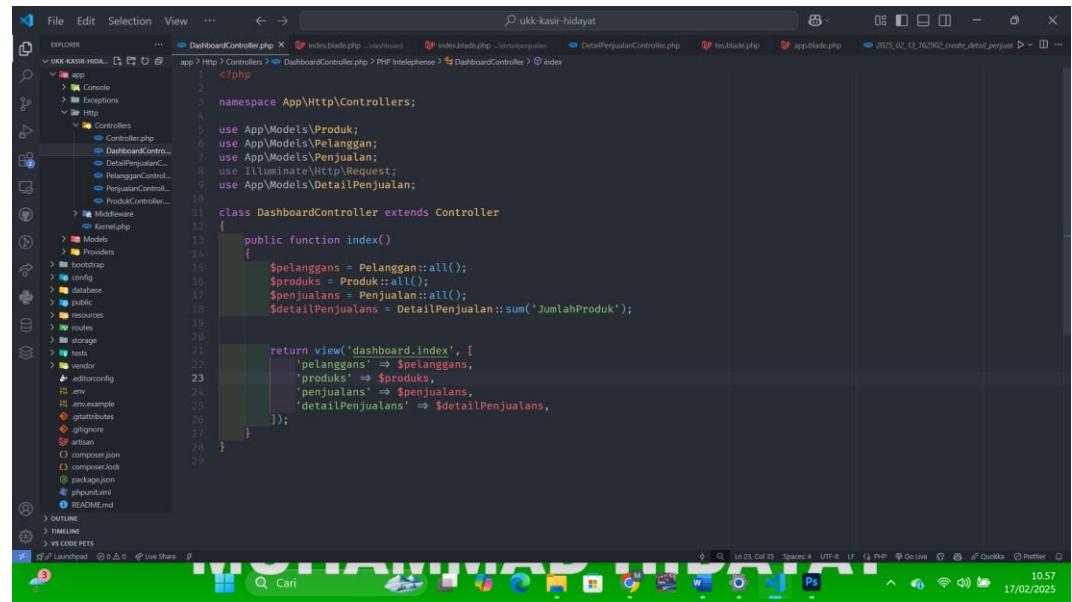


```
File Edit Selection View ... < - > | ukk-kasir-hidayat
EXPLORER ... controller.php index.blade.php ..._viewdropdown index.blade.php ..._viewdropdownpenjualan DetailPenjualanController.php tes.blade.php app.blade.php 2025_02_13_162902_create_detail_penjualan_table.php ...
database > migrations > 2025_02_13_162902_create_detail_penjualan_table.php ...
return new class extends Migration
{
    * Run the migrations.
    */
    public function up()
    {
        Schema::create('detailpenjualan', function (Blueprint $table) {
            $table->id('DetailID');
            $table->unsignedBigInteger('PenjualanID');
            $table->unsignedBigInteger('ProdukID');
            $table->integer('JumlahProduk');
            $table->decimal('Subtotal', 10, 2);

            $table->foreign('PenjualanID')->references('PenjualanID')->onDelete('cascade');
            $table->foreign('ProdukID')->references('ProdukID')->onDelete('cascade');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down()
    {
        Schema::dropIfExists('detailpenjualan');
    }
};
```

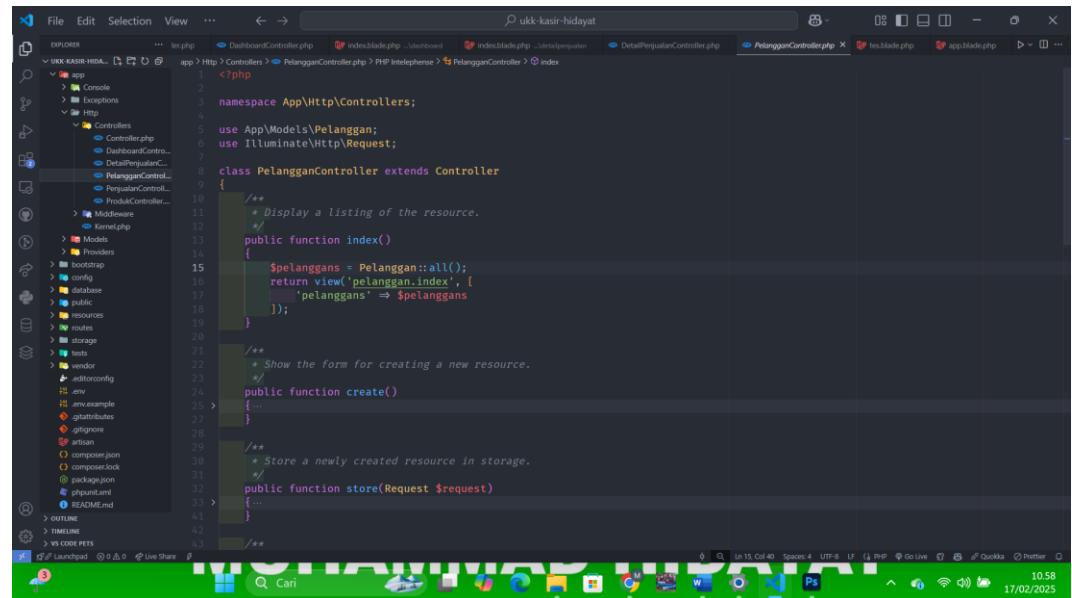
Gambar 1.2.6 Code program Laravel migration data detailpenjualan



The screenshot shows the Visual Studio Code interface with the file `DashboardController.php` open in the editor. The code defines a `DashboardController` class that extends `Controller`. The `index()` method retrieves all `Pelanggan`, `Produk`, and `Penjualan` records, and calculates the total value of sales. The controller then returns a view named `dashboard.index` with the data passed to it.

```
<?php  
namespace App\Http\Controllers;  
  
use App\Models\Produk;  
use App\Models\Pelanggan;  
use Illuminate\Http\Request;  
use App\Models\Penjualan;  
use App\Models\DetailPenjualan;  
  
class DashboardController extends Controller  
{  
    public function index()  
    {  
        $pelanggans = Pelanggan::all();  
        $produk = Produk::all();  
        $penjualans = Penjualan::all();  
        $detailPenjualans = DetailPenjualan::sum('JumlahProduk');  
  
        return view('dashboard.index', [  
            'pelanggans' => $pelanggans,  
            'produk' => $produk,  
            'penjualans' => $penjualans,  
            'detailPenjualans' => $detailPenjualans,  
        ]);  
    }  
}
```

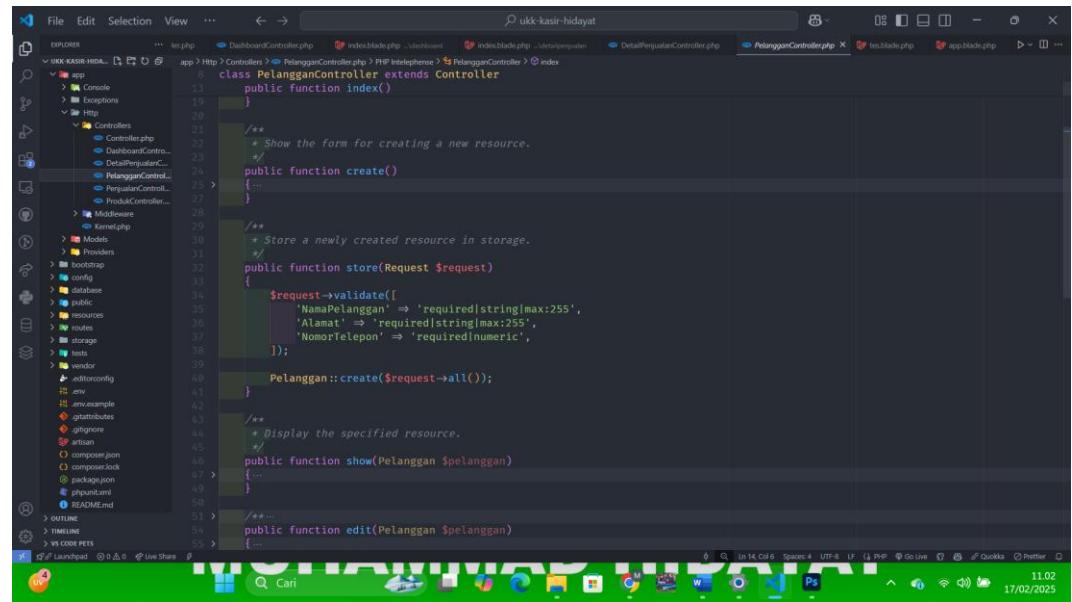
Gambar 1.3.1 Code program Laravel menampilkan tampilan dashboard



The screenshot shows the Visual Studio Code interface with the file `PelangganController.php` open in the editor. The code defines a `PelangganController` class that extends `Controller`. It includes methods for listing all `Pelanggan` records, creating a new record, and storing a newly created record in storage.

```
<?php  
namespace App\Http\Controllers;  
  
use App\Models\Pelanggan;  
use Illuminate\Http\Request;  
  
class PelangganController extends Controller  
{  
    /**  
     * Display a listing of the resource.  
     */  
    public function index()  
    {  
        $pelanggans = Pelanggan::all();  
        return view('pelanggan.index', [  
            'pelanggans' => $pelanggans  
        ]);  
    }  
  
    /**  
     * Show the form for creating a new resource.  
     */  
    public function create()  
    {  
    }  
  
    /**  
     * Store a newly created resource in storage.  
     */  
    public function store(Request $request)  
    {  
    }  
}
```

Gambar 1.3.2 Code program Laravel menampilkan tampilan pelanggan



The screenshot shows the Visual Studio Code interface with the file `PelangganController.php` open in the editor. The code is a PHP class extending the `Controller` class. It contains methods for creating, storing, displaying, editing, updating, and destroying a `Pelanggan` resource. The code uses validation rules and database storage.

```
class PelangganController extends Controller
{
    /**
     * Show the Form for creating a new resource.
     */
    public function index()
    {
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        $request->validate([
            'NamaPelanggan' => 'required|string|max:255',
            'Alamat' => 'required|string|max:255',
            'NomorTelepon' => 'required|numeric',
        ]);

        Pelanggan::create($request->all());
    }

    /**
     * Display the specified resource.
     */
    public function show(Pelanggan $pelanggan)
    {
    }

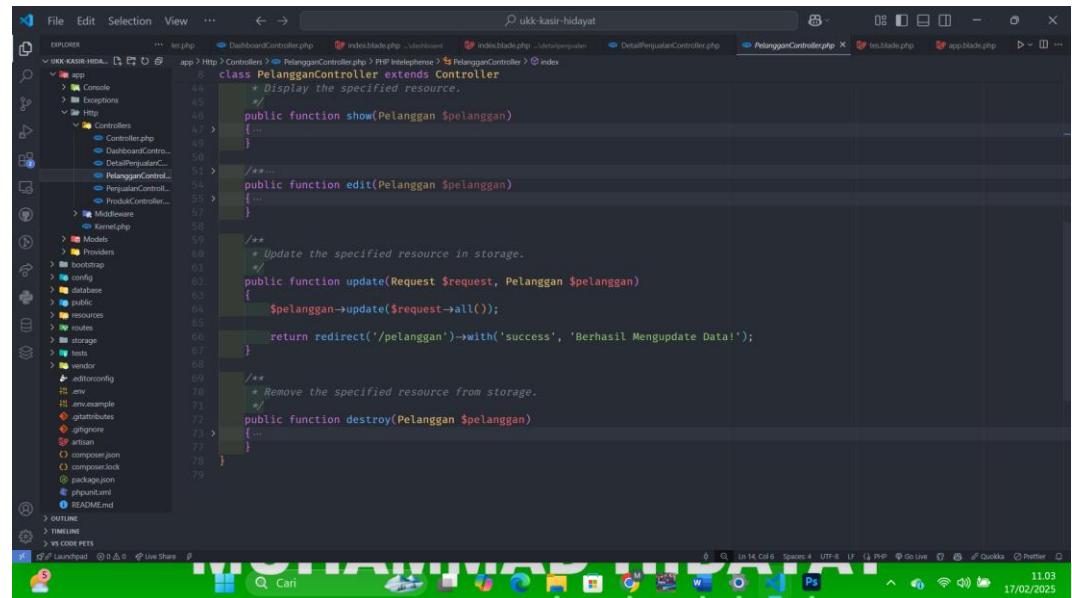
    /**
     * Edit the specified resource.
     */
    public function edit(Pelanggan $pelanggan)
    {
    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, Pelanggan $pelanggan)
    {
        $pelanggan->update($request->all());

        return redirect('/pelanggan')->with('success', 'Berhasil Mengupdate Data!');
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy(Pelanggan $pelanggan)
    {
    }
}
```

Gambar 1.4.1 Code program Laravel untuk menambahkan data pelanggan



The screenshot shows the Visual Studio Code interface with the file `PelangganController.php` open in the editor. The code is a PHP class extending the `Controller` class. It contains methods for displaying, editing, updating, and destroying a `Pelanggan` resource. The code uses validation rules and database storage.

```
class PelangganController extends Controller
{
    /**
     * Display the specified resource.
     */
    public function show(Pelanggan $pelanggan)
    {
    }

    /**
     * Edit the specified resource.
     */
    public function edit(Pelanggan $pelanggan)
    {
    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, Pelanggan $pelanggan)
    {
        $pelanggan->update($request->all());

        return redirect('/pelanggan')->with('success', 'Berhasil Mengupdate Data!');
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy(Pelanggan $pelanggan)
    {
    }
}
```

Gambar 1.5.1 Code program Laravel untuk mengupdate data pelanggan

The screenshot shows the Visual Studio Code interface with the file `PelangganController.php` open. The code is as follows:

```
class PelangganController extends Controller
{
    /**
     * Show the specified resource.
     *
     * @param  \App\Models\Pelanggan  $pelanggan
     * @return \Illuminate\Http\Response
     */
    public function show(Pelanggan $pelanggan)
    {
        //
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \App\Models\Pelanggan  $pelanggan
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, Pelanggan $pelanggan)
    {
        //
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  \App\Models\Pelanggan  $pelanggan
     * @return \Illuminate\Http\RedirectResponse
     */
    public function destroy(Pelanggan $pelanggan)
    {
        $pelanggan->delete();

        return redirect('/pelanggan')->with('success', 'Berhasil Menghapus Data!');
    }
}
```

Gambar 1.5.1 Code program Laravel untuk menghapus data pelanggan

The screenshot shows the Visual Studio Code interface with the file `ProdukController.php` open. The code is as follows:

```
<?php

namespace App\Http\Controllers;

use App\Models\Produk;
use Illuminate\Http\Request;

class ProdukController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $produk = Produk::all();
        return view('produk.index', compact('produk'));
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        //
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \App\Models\Produk  $produk
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, Produk $produk)
    {
        //
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  \App\Models\Produk  $produk
     * @return \Illuminate\Http\Response
     */
    public function destroy(Produk $produk)
    {
        //
    }
}
```

Gambar 1.6.1 Code program Laravel untuk menampilkan tampilan daftar produk

The screenshot shows the Visual Studio Code interface with the file 'ProductController.php' open in the editor. The code is a PHP class named 'ProdukController' extending 'Controller'. It contains methods for listing products ('index'), creating a new product ('store'), updating an existing product ('update'), and deleting a product ('destroy'). Validation rules are defined using Laravel's Validator facade. The code is well-formatted with color-coded syntax highlighting.

```
class ProdukController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        ...
    }

    public function store(Request $request)
    {
        $request->validate([
            'NamaProduk' => 'required|string|max:255',
            'Harga' => 'required|numeric',
            'Stok' => 'required|integer',
        ]);

        Produk::create($request->all());
    }

    public function update(Request $request, Produk $produk)
    {
        ...
    }

    public function destroy(Produk $produk)
    {
        ...
    }
}
```

Gambar 1.7.1 Code program Laravel untuk menambahkan data produk

This screenshot is identical to the one above, showing the 'ProductController.php' file in Visual Studio Code. The code remains the same, defining methods for listing, storing, updating, and destroying products. The validation logic for the 'store' and 'update' methods is clearly visible.

```
class ProdukController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        ...
    }

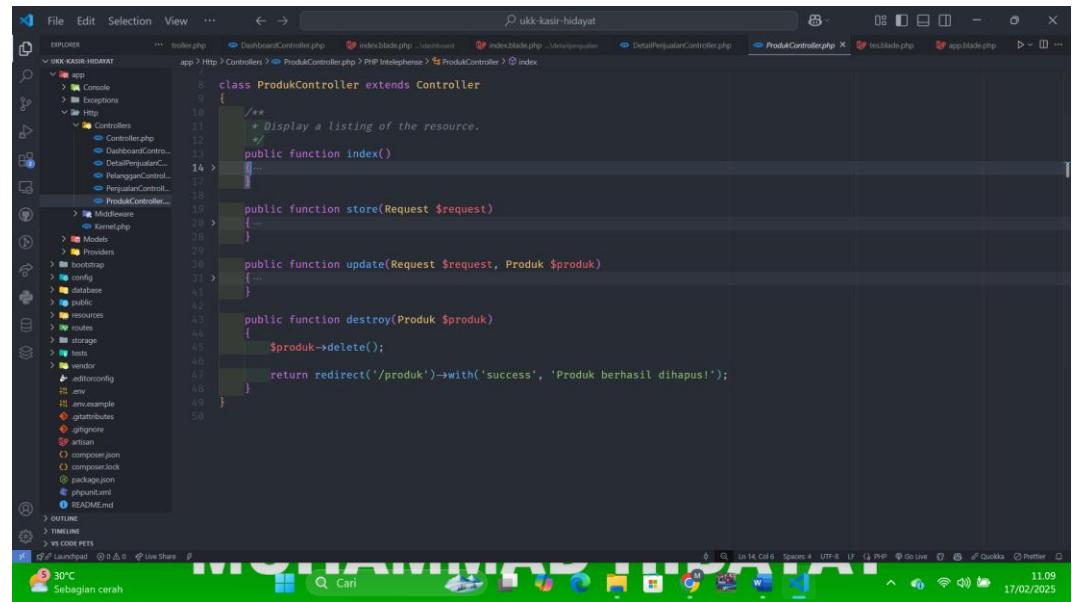
    public function store(Request $request)
    {
        $request->validate([
            'NamaProduk' => 'required|string|max:255',
            'Harga' => 'required|numeric',
            'Stok' => 'required|integer',
        ]);

        $produk->update($request->all());
    }

    public function update(Request $request, Produk $produk)
    {
        ...
    }

    public function destroy(Produk $produk)
    {
        ...
    }
}
```

Gambar 1.8.1 Code program Laravel untuk mengupdate data



The screenshot shows the VS Code interface with the file `ProdukController.php` open in the editor. The code implements a `ProdukController` class that extends `Controller`. It contains methods for listing products (`index`), storing new products (`store`), updating existing products (`update`), and deleting products (`destroy`). The `destroy` method uses the `delete` method of the `Produk` model to remove the product from the database and then redirects back to the product index page with a success message.

```
class ProdukController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        //
    }

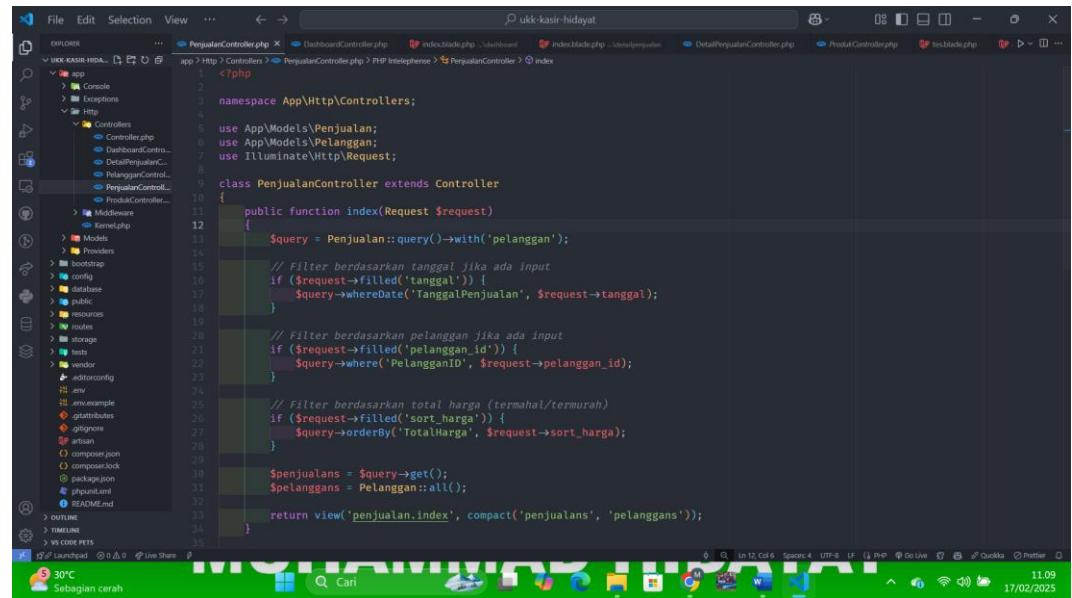
    public function store(Request $request)
    {
        //
    }

    public function update(Request $request, Produk $produk)
    {
        $produk->delete();

        return redirect('/produk')->with('success', 'Produk berhasil dihapus!');
    }

    public function destroy(Produk $produk)
    {
        //
    }
}
```

Gambar 1.8.2 Code program Laravel untuk menghapus data produk



The screenshot shows the VS Code interface with the file `PenjualanController.php` open in the editor. The code implements a `PenjualanController` class that extends `Controller`. The `index` method queries the `Penjualan` model and applies filters based on optional input parameters: `tanggal` and `pelanggan_id`. It also sorts the results by total price (`totalHarga`). Finally, it returns a view named `penjualan.index` with the filtered data.

```
<?php

namespace App\Http\Controllers;

use App\Models\Penjualan;
use App\Models\Pelanggan;
use Illuminate\Http\Request;

class PenjualanController extends Controller
{
    public function index(Request $request)
    {
        $query = Penjualan::query()->with('pelanggan');

        // Filter berdasarkan tanggal jika ada input
        if ($request->filled('tanggal')) {
            $query->whereDate('TanggalPenjualan', $request->tanggal);
        }

        // Filter berdasarkan pelanggan jika ada input
        if ($request->filled('pelanggan_id')) {
            $query->where('PelangganID', $request->pelanggan_id);
        }

        // Filter berdasarkan total harga (termahal/termurah)
        if ($request->filled('sort_harga')) {
            $query->orderBy('totalHarga', $request->sort_harga);
        }
    }

    $penjualans = $query->get();
    $pelanggans = Pelanggan::all();

    return view('penjualan.index', compact('penjualans', 'pelanggans'));
}
```

Gambar 1.9.1 Code program Laravel untuk menampilkan data penjualan

```
<?php  
namespace App\Http\Controllers;  
  
use App\Models\Penjualan;  
use App\Models\Pelanggan;  
use Illuminate\Http\Request;  
  
class PenjualanController extends Controller  
{  
    public function index(Request $request)  
    {  
        $request->validate([  
            'TanggalPenjualan' => 'required|date',  
            'PelangganID' => 'required|exists:pelanggan,PelangganID',  
        ]);  
  
        Penjualan::create([  
            'TanggalPenjualan' => $request->TanggalPenjualan,  
            'PelangganID' => $request->PelangganID,  
            'TotalHarga' => 0, // total dinitung dari detail penjualan nanti  
        ]);  
  
        return redirect('/penjualan')->with('success', 'Penjualan berhasil ditambahkan!');  
    }  
  
    public function update(Request $request, Penjualan $penjualan)  
    {  
        $request->validate([  
            'TanggalPenjualan' => 'required|date',  
            'PelangganID' => 'required|exists:pelanggan,PelangganID',  
        ]);  
  
        $penjualan->update($request->all());  
  
        return redirect('/penjualan')->with('success', 'Penjualan berhasil diperbarui!');  
    }  
  
    public function destroy(Penjualan $penjualan)  
    {  
        ...  
    }  
}
```

**Gambar 1.10.1** Code program Laravel untuk menambahkan transaksi penjualan

```
<?php  
namespace App\Http\Controllers;  
  
use App\Models\Penjualan;  
use App\Models\Pelanggan;  
use Illuminate\Http\Request;  
  
class PenjualanController extends Controller  
{  
    public function index(Request $request)  
    {  
        ...  
    }  
  
    public function store(Request $request)  
    {  
        ...  
    }  
  
    public function update(Request $request, Penjualan $penjualan)  
    {  
        $request->validate([  
            'TanggalPenjualan' => 'required|date',  
            'PelangganID' => 'required|exists:pelanggan,PelangganID',  
        ]);  
  
        $penjualan->update($request->all());  
  
        return redirect('/penjualan')->with('success', 'Penjualan berhasil diperbarui!');  
    }  
  
    public function destroy(Penjualan $penjualan)  
    {  
        ...  
    }  
}
```

**Gambar 1.10.2** Code program Laravel untuk mengubah transaksi penjualan

```
File Edit Selection View ... Explorer File Explorer Search Search: ukk-kasir-hidayat app > Http > Controllers > PenjualanController.php > PHP Intelephense > PenjualanController > index
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Penjualan;
6 use App\Models\Pelanggan;
7 use Illuminate\Http\Request;
8
9 class PenjualanController extends Controller
10 {
11     public function index(Request $request)
12     {
13         ...
14     }
15
16     public function store(Request $request)
17     {
18         ...
19     }
20
21     public function update(Request $request, Penjualan $penjualan)
22     {
23         ...
24     }
25
26     public function destroy(Penjualan $penjualan)
27     {
28         $penjualan->delete();
29
30         return redirect('/penjualan')->with('success', 'Penjualan berhasil dihapus!');
31     }
32 }
```

Gambar 1.10.2 Code program Laravel untuk menghapus transaksi penjualan

```
File Edit Selection View ... Explorer File Explorer Search Search: ukk-kasir-hidayat app > Http > Controllers > DetailPenjualanController.php > PHP Intelephense > DetailPenjualanController
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\DetailPenjualan;
7 use App\Models\Penjualan;
8 use App\Models\Produk;
9 use Illuminate\Support\Facades\DB;
10
11 class DetailPenjualanController extends Controller
12 {
13
14     public function edit($id)
15     {
16         $penjualan = Penjualan::with('details.produk')->findOrFail($id);
17         $produk = Produk::all();
18
19         return view('detailpenjualan.index', [
20             'penjualan' => $penjualan,
21             'produk' => $produk
22         ]);
23     }
24
25     public function update(Request $request, $id)
26     {
27         ...
28     }
29
30     public function destroy($id)
31     {
32         ...
33     }
34 }
```

Gambar 1.10.3 Code program Laravel untuk melihat detail penjualan tententu

```

class DetailPenjualanController extends Controller
{
    public function edit($id)
    {
        $request->validate([
            'ProdukID' => 'required|exists:produk,ProdukID',
            'JumlahProduk' => 'required|integer|min:1',
        ]);

        DB::transaction(function () use ($request, $id) {
            $produk = Produk::findOrFail($request->ProdukID);

            if ($produk->Stok < $request->JumlahProduk) {
                return back()->withErrors(['msg' => 'Stok tidak mencukupi']);
            }

            // Cek apakah produk sudah ada di detail penjualan
            $detail = DetailPenjualan::where('PenjualanID', $id)
                ->where('ProdukID', $request->ProdukID)
                ->first();

            $subtotal = $produk->Harga * $request->JumlahProduk;

            if ($detail) {
                // Jika ada detail, update jumlah dan subtotal
                $detail->increment('JumlahProduk', $request->JumlahProduk);
                $detail->increment('Subtotal', $subtotal);
            } else {
                // Tambahkan detail baru
                $detailPenjualan = create([
                    'PenjualanID' => $id,
                    'ProdukID' => $request->ProdukID,
                    'JumlahProduk' => $request->JumlahProduk,
                    'Subtotal' => $subtotal,
                ]);
            }

            // Mengurangi stok produk
            $produk->decrement('Stok', $request->JumlahProduk);

            // Update total harga di penjualan
            $penjualan = Penjualan::findOrFail($id);
            $penjualan->increment('TotalHarga', $subtotal);
        });
    }

    public function update(Request $request, $id)
    {
        ...
    }

    public function destroy($id)
    {
        DB::transaction(function () use ($id) {
            $detail = DetailPenjualan::findOrFail($id);
            $penjualan = Penjualan::findOrFail($detail->PenjualanID);
            $produk = Produk::findOrFail($detail->ProdukID);

            $produk->increment('Stok', $detail->JumlahProduk);
            $penjualan->decrement('TotalHarga', $detail->Subtotal);
            $detail->delete();
        });

        return back()->with('success', 'Produk dihapus dari detail penjualan!');
    }
}

```

**Gambar 1.10.4** Code program untuk mengupdate atau mengubah detail dari penjualan tersebut

```

use App\Models\Penjualan;
use App\Models\Produk;
use Illuminate\Support\Facades\DB;

class DetailPenjualanController extends Controller
{
    public function edit($id)
    {
        ...
    }

    public function update(Request $request, $id)
    {
        ...
    }

    public function destroy($id)
    {
        DB::transaction(function () use ($id) {
            $detail = DetailPenjualan::findOrFail($id);
            $penjualan = Penjualan::findOrFail($detail->PenjualanID);
            $produk = Produk::findOrFail($detail->ProdukID);

            $produk->increment('Stok', $detail->JumlahProduk);
            $penjualan->decrement('TotalHarga', $detail->Subtotal);
            $detail->delete();
        });

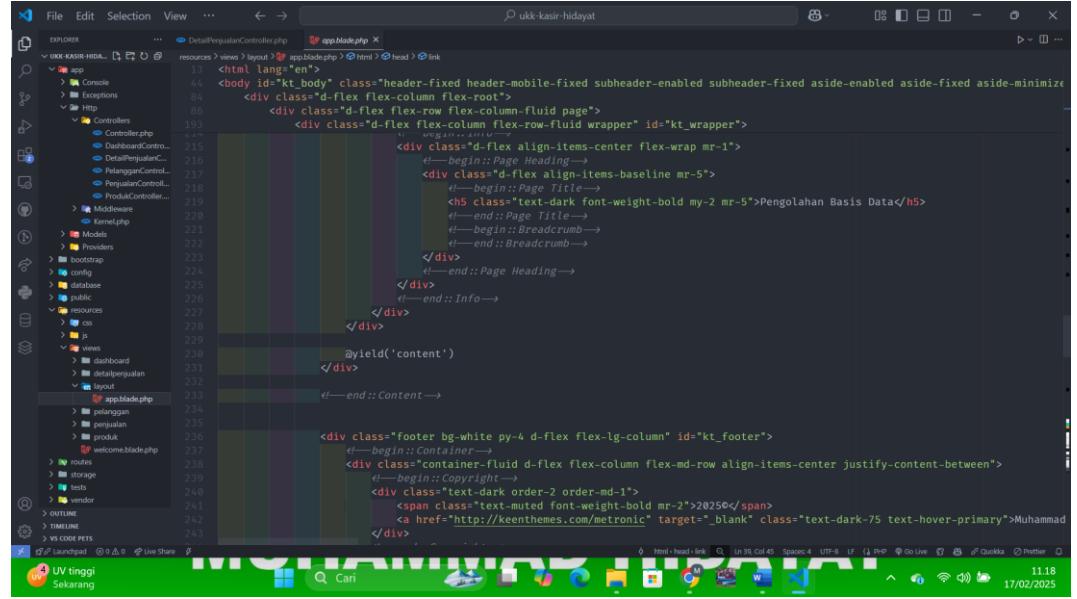
        return back()->with('success', 'Produk dihapus dari detail penjualan!');
    }
}

```

**Gambar 1.11.1** Code program Laravel untuk menghapus data dari detail penjualan

#### 4.1.7 Proses Pembuatan Code Program untuk User Interface

- Code

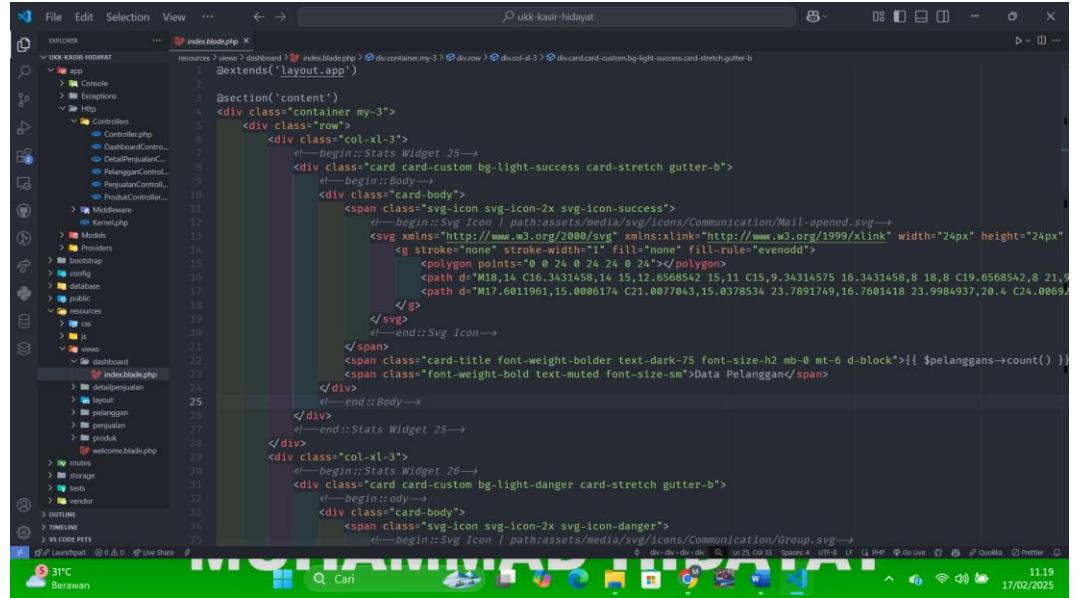


```

<html lang="en">
  <body id="kt_body" class="header-fixed header-mobile-fixed subheader-enabled subheader-fixed aside-enabled aside-fixed aside-minimize aside-minimize-v2">
    <div class="d-flex flex-column flex-root">
      <div class="d-flex flex-row flex-column-fluid wrapper" id="kt_wrapper">
        <div class="d-flex align-items-center flex-wrap mr-5">
          <!--begin::Page Heading-->
          <div class="d-flex align-items-baseline mr-5">
            <!--begin::Page Title-->
            <h5 class="text-dark font-weight-bold my-2 mr-5">Pengolahan Basis Data</h5>
            <!--end::Page Title-->
            <!--begin::Breadcrumb-->
            <div class="breadcrumb [<!-->]">
              <!--begin::Item-->
              <div class="item [<!-->]">
                <!--begin::Text-->
                <span class="text-muted font-weight-bold mr-2">20250</span>
                <a href="http://keenthemes.com/metronic" target="_blank" class="text-dark-75 text-hover-primary">Muhammad</a>
              <!--end::Text-->
            </div>
            <!--end::Item-->
          </div>
          <!--end::Page Heading-->
        </div>
        <!--begin::Info-->
        <div class="d-flex align-items-center flex-wrap ml-5">
          <!--begin::Section-->
          <div class="section [<!-->]">
            <!--begin::Section-->
            <div class="content [<!-->]">
              <!--begin::Section-->
              <div class="inner [<!-->]">
                <!--begin::Section-->
                <div class="inner [<!-->]">
                  <!--begin::Section-->
                  <div class="inner [<!-->]">
                    <!--begin::Section-->
                    <div class="inner [<!-->]">
                      <!--begin::Section-->
                      <div class="inner [<!-->]">
                        <!--begin::Section-->
                        <div class="inner [<!-->]">
                          <!--begin::Section-->
                          <div class="inner [<!-->]">
                            <!--begin::Section-->
                            <div class="inner [<!-->]">
                              <!--begin::Section-->
                              <div class="inner [<!-->]">
                                <!--begin::Section-->
                                <div class="inner [<!-->]">
                                  <!--begin::Section-->
                                  <div class="inner [<!-->]">
                                    <!--begin::Section-->
                                    <div class="inner [<!-->]">
                                      <!--begin::Section-->
                                      <div class="inner [<!-->]">
                                        <!--begin::Section-->
                                        <div class="inner [<!-->]">
                                          <!--begin::Section-->
                                          <div class="inner [<!-->]">
                                            <!--begin::Section-->
                                            <div class="inner [<!-->]">
                                              <!--begin::Section-->
                                              <div class="inner [<!-->]">
                                                <!--begin::Section-->
                                                <div class="inner [<!-->]">
                                                  <!--begin::Section-->
                                                  <div class="inner [<!-->]">
                                                    <!--begin::Section-->
                                                    <div class="inner [<!-->]">
                                                      <!--begin::Section-->
                                                      <div class="inner [<!-->]">
                                                        <!--begin::Section-->
                                                        <div class="inner [<!-->]">
                                                          <!--begin::Section-->
                                                          <div class="inner [<!-->]">
                                                            <!--begin::Section-->
                                                            <div class="inner [<!-->]">
                                                              <!--begin::Section-->
                                                              <div class="inner [<!-->]">
                                                                <!--begin::Section-->
                                                                <div class="inner [<!-->]">
                                                                  <!--begin::Section-->
                                                                  <div class="inner [<!-->]">
                                                                    <!--begin::Section-->
                                                                    <div class="inner [<!-->]">
                                                                      <!--begin::Section-->
                                                                      <div class="inner [<!-->]">
                                                                        <!--begin::Section-->
                                                                        <div class="inner [<!-->]">
                                                                          <!--begin::Section-->
                                                                          <div class="inner [<!-->]">
                                                                            <!--begin::Section-->
                                                                            <div class="inner [<!-->]">
                                                                              <!--begin::Section-->
                                                                              <div class="inner [<!-->]">
                                                                                <!--begin::Section-->
                                                                                <div class="inner [<!-->]">
                                                                                  <!--begin::Section-->
                                                                                  <div class="inner [<!-->]">
                                                                                    <!--begin::Section-->
                                                                                    <div class="inner [<!-->]">
                                                                                      <!--begin::Section-->
                                                                                      <div class="inner [<!-->]">
                                                                                        <!--begin::Section-->
                                                                                        <div class="inner [<!-->]">
                                                                                          <!--begin::Section-->
                                                                                          <div class="inner [<!-->]">
                                                                                            <!--begin::Section-->
                                                                                            <div class="inner [<!-->]">
                                                                                              <!--begin::Section-->
                                                                                              <div class="inner [<!-->]">
                                                                                                <!--begin::Section-->
                                                                                                <div class="inner [<!-->]">
                                                                                                  <!--begin::Section-->
                                                                                                  <div class="inner [<!-->]">
                                                                                                    <!--begin::Section-->
                                                                                                    <div class="inner [<!-->]">
                                                                                                      <!--begin::Section-->
                                                                                                      <div class="inner [<!-->]">
                                                                                                        <!--begin::Section-->
                                                                                                        <div class="inner [<!-->]">
                                                                                                          <!--begin::Section-->
................................................................

```

Gambar 2.1.1 Code program halaman layout (layout/app.blade.php)



```

<div class="d-flex flex-column flex-root">
  <div class="d-flex flex-row flex-column-fluid wrapper" id="kt_wrapper">
    <div class="d-flex align-items-center flex-wrap mr-5">
      <!--begin::Section-->
      <div class="section [<!-->]">
        <!--begin::Section-->
        <div class="content [<!-->]">
          <!--begin::Section-->
          <div class="inner [<!-->]">
            <!--begin::Section-->
            <div class="inner [<!-->]">
              <!--begin::Section-->
              <div class="inner [<!-->]">
                <!--begin::Section-->
                <div class="inner [<!-->]">
                  <!--begin::Section-->
                  <div class="inner [<!-->]">
                    <!--begin::Section-->
                    <div class="inner [<!-->]">
                      <!--begin::Section-->
                      <div class="inner [<!-->]">
                        <!--begin::Section-->
                        <div class="inner [<!-->]">
                          <!--begin::Section-->
                          <div class="inner [<!-->]">
                            <!--begin::Section-->
                            <div class="inner [<!-->]">
                              <!--begin::Section-->
................................................................

```

Gambar 2.1.2 Code program halaman dashboard (dashboard/index.blade.php)

```

<section>'content'>
    <div class="container my-3">
        @if (session('success'))
            <div class="alert alert-custom alert-primary" role="alert">
                <div class="alert-icon">
                    <i class="flaticon-warning"></i>
                </div>
                <div class="alert-text">{{ session('success') }}</div>
            </div>
        @endif
        <div class="card card-custom">
            <div class="card-header flex-wrap border-0 pt-6 pb-0">
                <div class="card-title">
                    <h3 class="card-label">Data Pelanggan</h3>
                    <span class="text-muted pt-2 font-size-sm d-block">List data pelanggan</span>
                </div>
            </div>
            <div class="card-body">
                <!-- begin: Datatable -->
                <div class="datatable bordered datatables-head-custom datable-default datable-primary datable-loaded" id="kt_datatable">
                    <table class="datatable">
                        <thead>
                            <tr>
                                <th class="datatable-cell"><span style="width: 50px;">No</span></th>
                                <th class="datatable-cell"><span style="width: 170px;">Nama Pelanggan</span></th>
                                <th class="datatable-cell"><span style="width: 110px;">Alamat</span></th>
                                <th class="datatable-cell"><span style="width: 110px;">Nomor Telepon</span></th>
                                <th class="datatable-cell"><span style="width: 115px;">Actions</span></th>
                            </tr>
                        </thead>
                        <tbody class="datatable-body">
                            @foreach ($pelanggans as $pelanggan)
                                <tr class="odd">
                                    <td class="datatable-cell"><span style="width: 50px;">{{ $loop->iteration }}</span></td>
                                    <td class="datatable-cell"><span style="width: 170px;">{{ $pelanggan->NamePelanggan }}</span></td>
                                    <td class="datatable-cell"><span style="width: 110px;">{{ $pelanggan->Alamat }}</span></td>
                                    <td class="datatable-cell"><span style="width: 110px;">{{ $pelanggan->NomorTelepon }}</span></td>
                                    <td class="datatable-cell"><span style="width: 115px;">{{ $pelanggan->NameTalunan }}</span></td>
                                </tr>
                            @endforeach
                        </tbody>
                    </table>
                </div>
                <button class="btn btn-sm btn-light-success font-weight-bold mr-2" data-toggle="modal" data-target="#addModal">Tambah Pelanggan</button>
            </div>
        </div>
    </div>
</section>

```

**Gambar 2.1.3** Code program halaman data pelanggan  
(pelanggan/index.blade.php)

```

<div class="modal-dialog" role="document">
    <div class="modal-content">
        <div class="modal-header">
            <h5 class="modal-title" id="addModallabel">Tambah Pelanggan</h5>
            <button type="button" class="close" data-dismiss="modal" aria-label="Close"><i aria-hidden="true" class="ki ki-close"></i></button>
        </div>
        <div class="modal-body">
            <form id="addForm">
                <csrf>
                    <div class="form-group">
                        <label for="addNamaPelanggan">Nama Pelanggan</label>
                        <input type="text" class="form-control" id="addNamaPelanggan" required>
                    </div>
                    <div class="form-group">
                        <label for="addAlamat">Alamat</label>
                        <textarea class="form-control" id="addAlamat" rows="3" required></textarea>
                    </div>
                    <div class="form-group">
                        <label for="addNomorTelepon">Nomor Telepon</label>
                        <input type="text" class="form-control" id="addNomorTelepon" required pattern="[0-9]*" title="Hanya angka diperbolehkan" required>
                    </div>
                </form>
            <div class="modal-footer">
                <button type="button" class="btn btn-light-primary" data-dismiss="modal">Tutup</button>
                <button type="button" class="btn btn-light-success font-weight-bold mr-2" id="savePelanggan">Simpan</button>
            </div>
        </div>
    </div>

```

**Gambar 2.1.4** Code program modal untuk menambahkan data pelanggan

The screenshot shows the Visual Studio Code interface with the file `index.blade.php` open in the editor. The code displays the structure for a modal dialog used for editing customer data. It includes sections for the modal header, form fields for name, address, and phone number, and a footer with save and close buttons. The code uses Bootstrap classes like `modal-fade`, `modal-dialog`, and `modal-content`.

```
</div>
    <!-- Modal Edit -->
    <div class="modal fade" id="editModal" tabindex="-1" role="dialog" aria-labelledby="editModalLabel" aria-hidden="true">
        <div class="modal-dialog" role="document">
            <div class="modal-header">
                <h5 class="modal-title" id="editModalLabel">Edit Pelanggan</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&amptimes</span>
                </button>
            </div>
            <form id="editForm" method="POST">
                @csrf
                @method('PUT')
                <div class="modal-body">
                    <input type="hidden" id="editPelangganID" name="PelangganID">
                    <div class="form-group">
                        <label for="editNamaPelanggan">Nama Pelanggan</label>
                        <input type="text" class="form-control" id="editNamaPelanggan" name="NamaPelanggan" required>
                    </div>
                    <div class="form-group">
                        <label for="editAlamat">Alamat</label>
                        <input type="text" class="form-control" id="editAlamat" name="Alamat" required>
                    </div>
                    <div class="form-group">
                        <label for="editNomorTelepon">Nomor Telepon</label>
                        <input type="text" class="form-control" id="editNomorTelepon" name="NomorTelepon" required>
                    </div>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                    <button type="submit" class="btn btn-primary">Save changes</button>
                </div>
            </form>
        </div>
    </div>
```

Gambar 2.1.5 Code program modal untuk mengedit data pelanggan

The screenshot shows the Visual Studio Code interface with the file `index.blade.php` open in the editor. The code displays the structure for a modal dialog used for deleting customer data. It includes a confirmation message, input fields for the customer ID, and a footer with cancel and delete buttons. The code uses Bootstrap classes like `modal-fade`, `modal-dialog`, and `modal-content`.

```
</div>
    <!-- Modal Hapus -->
    <div class="modal fade" id="deleteModal" tabindex="-1" role="dialog" aria-labelledby="deleteModalLabel" aria-hidden="true">
        <div class="modal-dialog" role="document">
            <div class="modal-header">
                <h5 class="modal-title" id="deleteModalLabel">Konfirmasi Hapus</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&amptimes</span>
                </button>
            </div>
            <form id="deleteForm" method="POST">
                @csrf
                @method('DELETE')
                <div class="modal-body">
                    <p>Apakah Anda yakin menghapus <b><input type="hidden" id="deletePelangganID" name="PelangganID"></b>?</p>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary" data-dismiss="modal">Batal</button>
                    <button type="submit" class="btn btn-danger">Hapus</button>
                </div>
            </form>
        </div>
    </div>
    <!-- Modal Tambah Data -->
    <div class="modal fade" id="addModal" tabindex="-1" role="dialog" aria-labelledby="addModalLabel" aria-hidden="true">
```

Gambar 2.1.6 Code program modal untuk menghapus data pelanggan

```

$(document).ready(function() {
    $('#btn-edit').click(function() {
        var name = $(this).data('name');
        var alamat = $(this).data('alamat');
        var telepon = $(this).data('telepon');

        $('#editForm').attr('action', '/pelanggan/' + id); // Update action form
        $('#editModal').modal('show');
    });

    // Memanajikan model baru dengan data pelanggan yang dipilih
    $('#btn-delete').click(function() {
    });

    $('#savePelanggan').click(function() {
        var data = {
            NamaPelanggan: $('#addNamaPelanggan').val(),
            Alamat: $('#addAlamat').val(),
            NomorTelepon: $('#addNomorTelepon').val(),
            _token: '{{ csrf_token() }}'
        };

        $.ajax({
            url: '/pelanggan',
            type: 'POST',
            data: data,
            success: function(response) {
                location.reload();
            }
        });
    });
});

```

**Gambar 2.1.7** Code program javascript menggunakan jquery untuk halaman data pelanggan

```

@extends('layout.app')

@section('content')
    @if (session('success'))
        

{{ session('success') }}


    @endif

### Daftar Produk



Tambah Produk



| #                       | Nama Produk              | Harga                 | Stock                 | Actions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------|--------------------------|-----------------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {{ \$loop->iteration }} | {{ \$produk->ProdukID }} | {{ \$produk->Harga }} | {{ \$produk->Stock }} | <a data-harga="{{ \$produk-&gt;Harga }}" data-id="{{ \$produk-&gt;ProdukID }}" data-name="{{ \$produk-&gt;NamaProduk }}" data-stok="{{ \$produk-&gt;Stock }}" data-target="#editModal" data-toggle="modal" href="#"><i class="fa fa-edit"></i></a> <a data-harga="{{ \$produk-&gt;Harga }}" data-id="{{ \$produk-&gt;ProdukID }}" data-name="{{ \$produk-&gt;NamaProduk }}" data-stok="{{ \$produk-&gt;Stock }}" data-target="#deleteModal" data-toggle="modal" href="#"><i class="fa fa-trash"></i></a> |


```

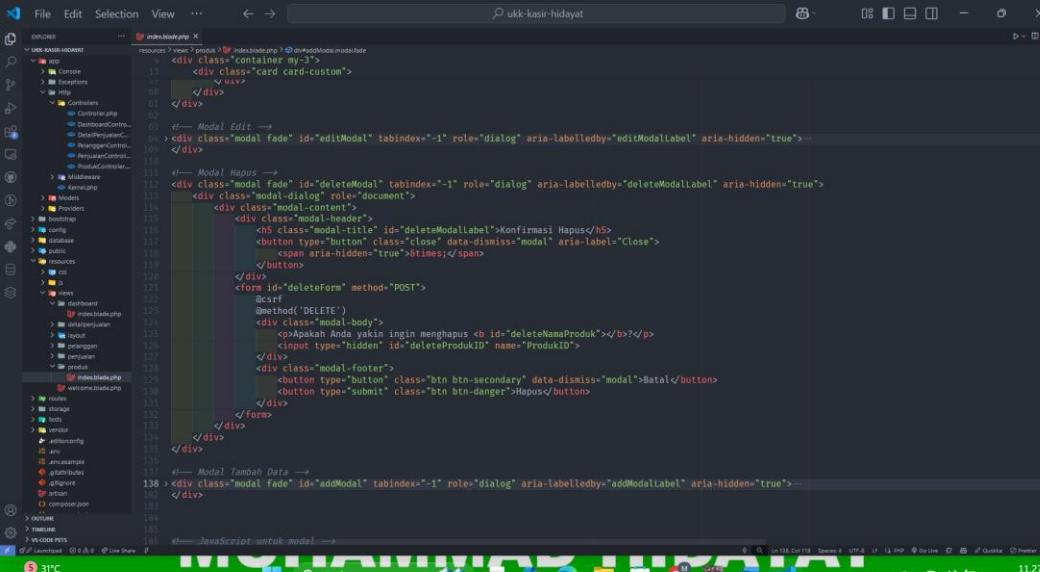
**Gambar 2.2.1** Code program halaman data produk  
(produk/index.blade.php)

```
<!-- Modal Tambah Data -->
<div class="modal fade" id="addModal" tabindex="-1" role="dialog" aria-labelledby="addModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="addModalLabel">Tambah Produk</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <i class="ki ki-close"></i>
                </button>
            </div>
            <div class="modal-body">
                <form id="addForm">
                    <div class="form-group">
                        <label for="addNamaProduk">Nama Produk</label>
                        <input type="text" class="form-control" id="addNamaProduk" required>
                    </div>
                    <div class="form-group">
                        <label for="addHarga">Harga</label>
                        <div class="input-group bootstrap-touchspin bootstrap-touchspin-injected">...</div>
                    </div>
                    <div class="form-group">
                        <label for="addStok">Stok</label>
                        <div class="input-group bootstrap-touchspin bootstrap-touchspin-injected">...</div>
                    </div>
                </form>
            <div class="modal-footer">
                <button type="button" class="btn btn-light-primary" data-dismiss="modal">Tutup</button>
                <button type="button" class="btn btn-light-success font-weight-bold mr-2" id="saveProduk">Simpan</button>
            </div>
        </div>
    </div>
</div>
```

Gambar 2.2.2 Code program modal untuk menambahkan data produk

```
<!-- Modal Edit -->
<div class="modal fade" id="editModal" tabindex="-1" role="dialog" aria-labelledby="editModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="editModalLabel">Edit Produk</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <i class="ki ki-close"></i>
                </button>
            </div>
            <div id="editForm" method="POST">
                <input type="hidden" name="ProdukID" value="..."/>
                <div class="modal-body">
                    <input type="hidden" id="editProdukID" name="ProdukID">
                    <div class="form-group">
                        <label for="editNamaProduk">Nama Produk</label>
                        <input type="text" class="form-control" id="editNamaProduk" name="NamaProduk" required>
                    </div>
                    <div class="form-group">
                        <label for="editHarga">Harga</label>
                        <div class="input-group bootstrap-touchspin bootstrap-touchspin-injected">...</div>
                    </div>
                    <div class="form-group">
                        <label for="editStok">Stok</label>
                        <div class="input-group bootstrap-touchspin bootstrap-touchspin-injected">...</div>
                    </div>
                </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                <button type="button" class="btn btn-primary">Save changes</button>
            </div>
        </div>
    </div>
</div>
```

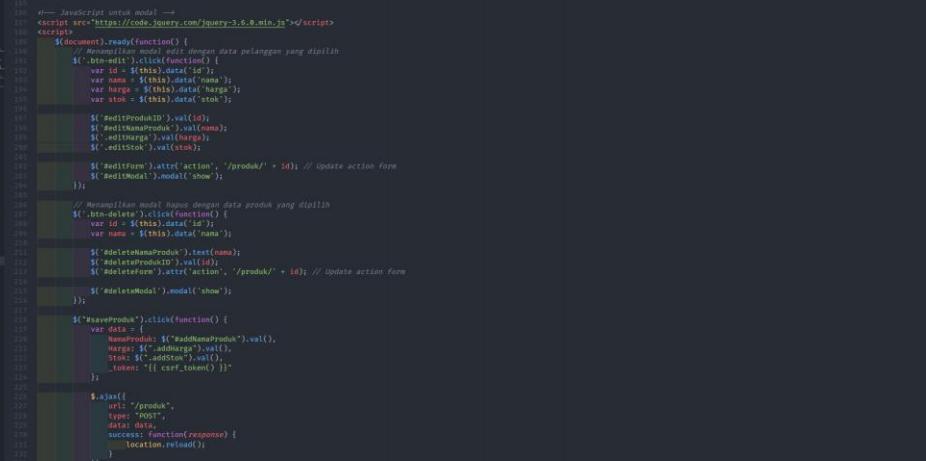
Gambar 2.2.3 Code program modal untuk mengedit data produk



The screenshot shows a Microsoft Visual Studio Code interface with the following details:

- File Path:** `src/resources/viewer/produk/index.blade.php`
- Code Content:** The code is a Blade template for a product management application. It includes sections for adding products, editing products, and deleting products. The modal dialogs are implemented using Bootstrap's modal component.
- Code Snippets:** Several parts of the code are highlighted with red and yellow annotations, likely from an IDE feature like 'Find Usages' or 'Definition'.
- Bottom Status Bar:** Shows file paths like `index.blade.php`, line numbers `1-118 Col 118`, and other standard developer information.
- Bottom Icons:** Includes icons for file operations, search, and navigation.
- Bottom Status Bar:** Shows the date `17/02/2025`.

**Gambar 2.2.4** Code program modal untuk menghapus data produk



```
File Edit Selection View ... ← → ⌘ ukk-kasir-hidayat
src/index.blade.php
...
<!-- Javascript untuk modal -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function() {
    // Mengambil modal edit dengan data pelanggan yang dipilih
    $('#editModal').on('show.bs.modal', function(e) {
        var id = $(this).data('id');
        var name = $(this).data('name');
        var address = $(this).data('address');
        var stok = $(this).data('stok');

        $('#editProdukID').val(id);
        $('#editNameProduk').val(name);
        $('#editAddress').val(address);
        $('#editStok').val(stok);

        $('#editForm').attr('action', '/produk/' + id); // update action form
    });

    // Mengambil modal hapus dengan data produk yang dipilih
    $('#btn-delete').click(function() {
        var id = $(this).data('id');
        var name = $(this).data('name');

        $('#deleteNameProduk').text(name);
        $('#deleteAddress').attr('action', '/produk/' + id); // Update action form

        $('#deleteModal').modal('show');
    });

    $('#saveProduk').click(function() {
        var data = [
            namaProduk: $('#addNameProduk').val(),
            alamat: $('#addAddress').val(),
            stok: $('#addStok').val(),
            _token: '{{ csrf_token() }}'
        ];

        $.ajax({
            url: "/produk",
            type: "POST",
            data: data,
            success: function(response) {
                window.location.reload();
            }
        });
    });
});
</script>
```

**Gambar 2.2.5** Code program javascript menggunakan jquery untuk halaman data produk

```

<section>'content'>
    <div class="container my-3">
        @if (session('success'))
            <div class="alert alert-custom alert-primary" role="alert">
                <div class="alert-icon">
                    <i class="flaticon-warning"></i>
                </div>
                <div class="alert-text">{{ session('success') }}</div>
            </div>
        @endif

        <form method="GET" action="{{ route('penjualan.index') }}">
            <div class="row align-items-center">
                <div class="col-8">
                    <div class="col-12 mb-3">
                        <label>Tanggal:</label>
                        <input type="date" class="form-control" name="tanggal" value="{{ request('tanggal') }}>
                    </div>

                    <!-- Filter Pelanggan -->
                    <div class="col-md-3">
                        <label>Pelanggan:</label>
                        <select class="form-control" name="pelanggan_id">
                            <option value="">> Semua --</option>
                            @foreach ($pelanggans as $pelanggan)
                                <option value="{{ $pelanggan->id }}> {{ $pelanggan->NamaPelanggan }}</option>
                            @endforeach
                        </select>
                    </div>

                    <!-- Urutkan Total Harga -->
                    <div class="col-md-3">
                        <label>Harga:</label>
                        <select class="form-control" name="sort_harga">
                            <option value="">> Pilih --</option>
                            <option value="asc" {{ request('sort_harga') == 'asc' ? 'selected' : '' }}>Termurah</option>
                            <option value="desc" {{ request('sort_harga') == 'desc' ? 'selected' : '' }}>Termahal</option>
                        </select>
                    </div>
                </div>
            </div>
        </form>
    </div>

```

**Gambar 2.2.6** Code program halaman penjualan  
(penjualan\index.blade.php)

```

<div class="modal fade" id="addModal">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5>Tambah Penjualan</h5>
                <button type="button" class="close" data-dismiss="modal">&times;</button>
            </div>
            <form action="{{ route('penjualan.store') }}" method="POST">
                @csrf
                <div class="modal-body">
                    <div class="form-group">
                        <label for="TanggalPenjualan">Tanggal</label>
                        <input type="date" class="form-control" name="TanggalPenjualan" required>
                    </div>
                    <div class="form-group">
                        <label for="PelangganID">Pelanggan</label>
                        <select class="form-control" name="PelangganID">
                            @foreach ($pelanggans as $pelanggan)
                                <option value="{{ $pelanggan->id }}> {{ $pelanggan->NamaPelanggan }}</option>
                            @endforeach
                        </select>
                    </div>
                </div>
                <div class="modal-Footer">
                    <button type="button" class="btn btn-secondary" data-dismiss="modal">Batal</button>
                    <button type="submit" class="btn btn-success">Simpan</button>
                </div>
            </form>
        </div>
    </div>

```

**Gambar 2.3.1** Code program modal untuk menambahkan transaksi penjualan

The screenshot shows a code editor with the file `index.blade.php` open. The code is a modal form for editing a sales transaction. It includes fields for PenjualanID (hidden), TanggalPenjualan (date input), PelangganID (select dropdown with options for each pelanggan), and a footer with cancel and save buttons. The code uses Bootstrap classes like `modal-fade`, `modal-dialog`, and `modal-content`.

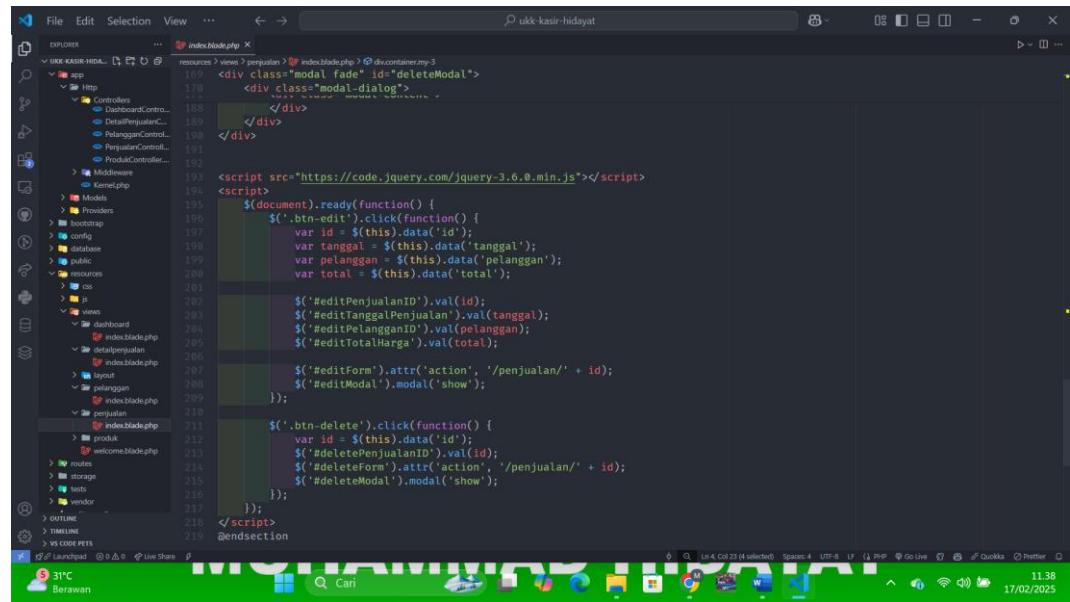
```
<div class="modal fade" id="editModal">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Edit Penjualan</h5>
                <button type="button" class="close" data-dismiss="modal">&times;
            </div>
            <form id="editForm" method="POST">
                @csrf
                @method('PUT')
                <div class="modal-body">
                    <input type="hidden" id="editPenjualanID" name="PenjualanID">
                    <div class="form-group">
                        <label for="TanggalPenjualan">Tanggal</label>
                        <input type="date" class="form-control" id="editTanggalPenjualan" name="TanggalPenjualan" required>
                    </div>
                    <div class="form-group">
                        <label for="editPelangganID">Pelanggan</label>
                        <select class="form-control" id="editPelangganID" name="PelangganID">
                            @foreach ($pelanggans as $pelanggan)
                                <option value="{{ $pelanggan->PelanganID }}>{{ $pelanggan->NamaPelanggan }}</option>
                            @endforeach
                        </select>
                    </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary" data-dismiss="modal">Batal</button>
                    <button type="submit" class="btn btn-primary">Simpan Perubahan</button>
                </div>
            </form>
        </div>
    </div>
</div>
```

Gambar 2.3.2 Code program modal untuk mengedit transaksi penjualan

The screenshot shows a code editor with the file `index.blade.php` open. The code is a modal form for deleting a sales transaction. It asks for confirmation before deletion, has a hidden field for PenjualanID, and a footer with cancel and delete buttons. The code uses Bootstrap classes like `modal-fade`, `modal-dialog`, and `modal-content`.

```
<div class="modal fade" id="deleteModal">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Hapus Penjualan</h5>
                <button type="button" class="close" data-dismiss="modal">&times;
            </div>
            <form id="deleteForm" method="POST">
                @csrf
                @method('DELETE')
                <div class="modal-body">
                    <p>Apakah Anda yakin ingin menghapus transaksi ini?</p>
                    <input type="hidden" id="deletePenjualanID" name="PenjualanID">
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary" data-dismiss="modal">Batal</button>
                    <button type="submit" class="btn btn-danger">Hapus</button>
                </div>
            </form>
        </div>
    </div>
</div>
```

Gambar 2.3.3 Code program modal untuk menghapus transaksi penjualan



```

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function() {
    $('.btn-edit').click(function() {
        var id = $(this).data('id');
        var tanggal = $(this).data('tanggal');
        var pelanggan = $(this).data('pelanggan');
        var total = $(this).data('total');

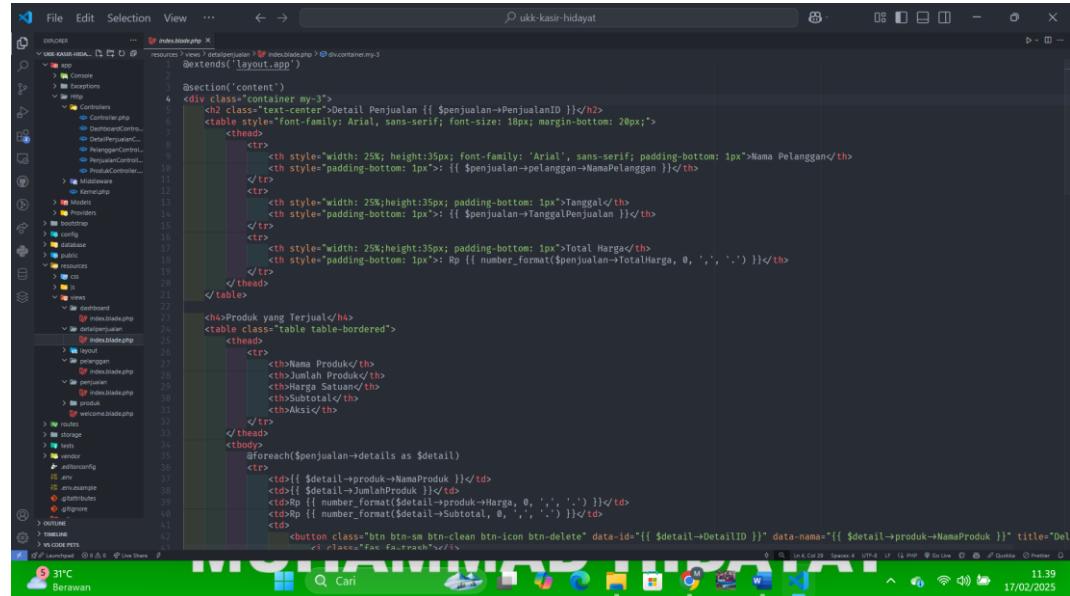
        $('#editPenjualanID').val(id);
        $('#editTanggalPenjualan').val(tanggal);
        $('#editPelangganID').val(pelanggan);
        $('#editTotalHarga').val(total);

        $('#editForm').attr('action', '/penjualan/' + id);
        $('#editModal').modal('show');
    });

    $('.btn-delete').click(function() {
        var id = $(this).data('id');
        $('#deletePenjualanID').val(id);
        $('#deleteForm').attr('action', '/penjualan/' + id);
        $('#deleteModal').modal('show');
    });
});
</script>

```

**Gambar 2.3.4** Code program javascript menggunakan jquery untuk halaman penjualan



```

<@extends('layout.app')>

<@section('content')>
    <div class="container my-3">
        <h2 class="text-center">Detail Penjualan {{ $penjualan->PenjualanID }}

```

**Gambar 2.3.5** Code program halaman detail penjualan (detailpenjualan\index.blade.php)

The screenshot shows the VS Code interface with the file `index.blade.php` open. The code is part of a Laravel application, specifically within the `resources/views/detailpenjualan` directory. The code defines a form for adding a product to a sales detail:

```
<div class="container my-3">
    <table class="table table-bordered">
        ...
    </table>

<h4>Tambah Produk</h4>
<form action="{{ route('detail-penjualan.update', $penjualan->PenjualanID) }}" method="POST">
    @csrf
    @method('PUT')
    <div class="form-group">
        <label>Produk</label>
        <select name="ProdukID" class="form-control">
            @foreach($produk as $produk)
                <option value="{{ $produk->ProdukID }}>{{ $produk->NamaProduk }} (Stok: {{ $produk->Stok }})
            @endforeach
        </select>
    </div>
    <div class="form-group">
        <label>Jumlah</label>
        <input type="number" name="JumlahProduk" class="form-control" min="1" required>
    </div>
    <button type="submit" class="btn btn-primary">Tambah</button>
</form>

<div class="modal fade" id="deleteModal" tabindex="-1" role="dialog" aria-labelledby="deleteModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
```

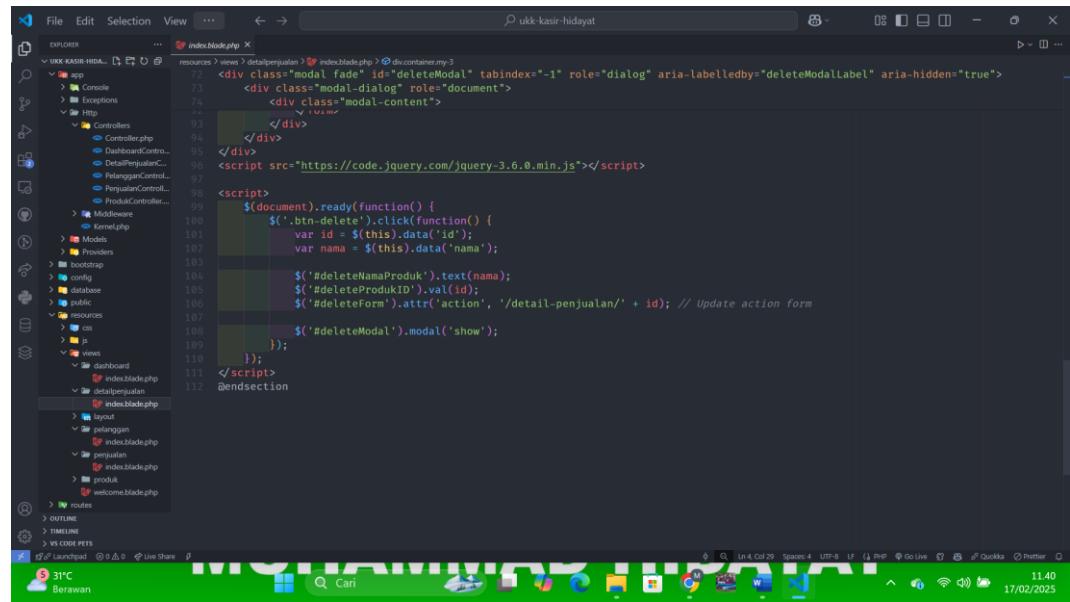
Gambar 2.4.1 Code program form untuk menambahkan produk ke detail penjualan

The screenshot shows the VS Code interface with the file `index.blade.php` open. The code is part of a Laravel application, specifically within the `resources/views/detailpenjualan` directory. It includes a modal for confirming the deletion of a product:

```
<div class="container my-3">
    <form action="{{ route('detail-penjualan.update', $penjualan->PenjualanID) }}" method="POST">
        ...
    </div>

    <div class="modal fade" id="deleteModal" tabindex="-1" role="dialog" aria-labelledby="deleteModalLabel" aria-hidden="true">
        <div class="modal-dialog" role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title" id="deleteModalLabel">Konfirmasi Hapus</h5>
                    <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                        <span aria-hidden="true">&amptimes</span>
                    </button>
                </div>
                <form id="deleteForm" method="POST">
                    @csrf
                    @method('DELETE')
                    <div class="modal-body">
                        <p>Apakah Anda yakin ingin menghapus <b>{{ $produk->NamaProduk }}</b>?</p>
                        <input type="hidden" id="deleteProdukID" name="PelangganID">
                    </div>
                    <div class="modal-footer">
                        <button type="button" class="btn btn-secondary" data-dismiss="modal">Batal</button>
                        <button type="submit" class="btn btn-danger">Hapus</button>
                    </div>
                </form>
            </div>
        <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
        <script>
            $(document).ready(function() {
                ...
            });
        </script>
    </div>
</div>
```

Gambar 2.4.2 Code program modal untuk menghapus produk dari detail penjualan

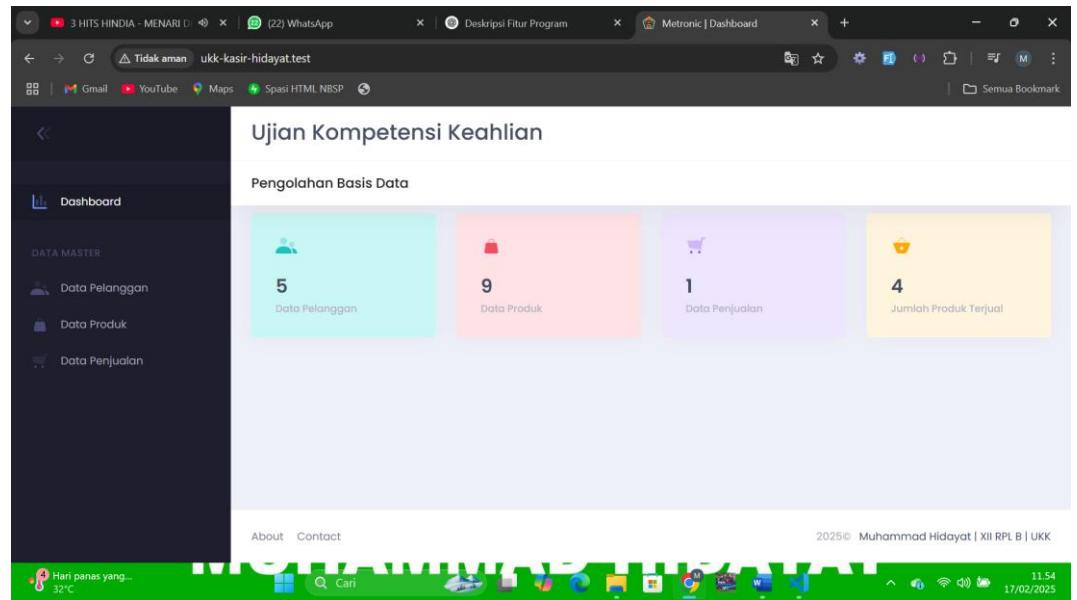


The screenshot shows a code editor window with the file `index.blade.php` open. The code is a part of a Laravel application, specifically for a modal delete feature. The modal is triggered by a button with the class `.btn-delete`. When clicked, it retrieves the product name from the data attribute `nama`, the product ID from `id`, and updates the form action to include the ID. The modal is then shown.

```
resources > views > detail-penjualan > index.blade.php > div.container.my-3
72     <div class="modal fade" id="deleteModal" tabindex="-1" role="dialog" aria-labelledby="deleteModalLabel" aria-hidden="true">
73         <div class="modal-dialog" role="document">
74             <div class="modal-content">
75                 <div class="modal-header">
76                     <button type="button" class="close" data-dismiss="modal" aria-label="Close">&amptimes</button>
77                 </div>
78                 <div class="modal-body">
79                     <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
80
81                     <script>
82                         $(document).ready(function() {
83                             $('.btn-delete').click(function() {
84                                 var id = $(this).data('id');
85                                 var nama = $(this).data('nama');
86
87                                 $('#deleteNamaProduk').text(nama);
88                                 $('#deleteProdukID').val(id);
89                                 $('#deleteForm').attr('action', '/detail-penjualan/' + id); // Update action form
90
91                                 $('#deleteModal').modal('show');
92                             });
93                         });
94                     </script>
95                 </div>
96             </div>
97         </div>
98     </div>
99
100    <script>
101        $('.btn-delete').click(function() {
102            var id = $(this).data('id');
103            var nama = $(this).data('nama');
104
105            $('#deleteNamaProduk').text(nama);
106            $('#deleteProdukID').val(id);
107            $('#deleteForm').attr('action', '/detail-penjualan/' + id); // Update action form
108
109            $('#deleteModal').modal('show');
110        });
111    </script>
112 @endsection
```

**Gambar 2.4.3** Code program javascript menggunakan jquery untuk halaman detail-penjualan

- Hasil



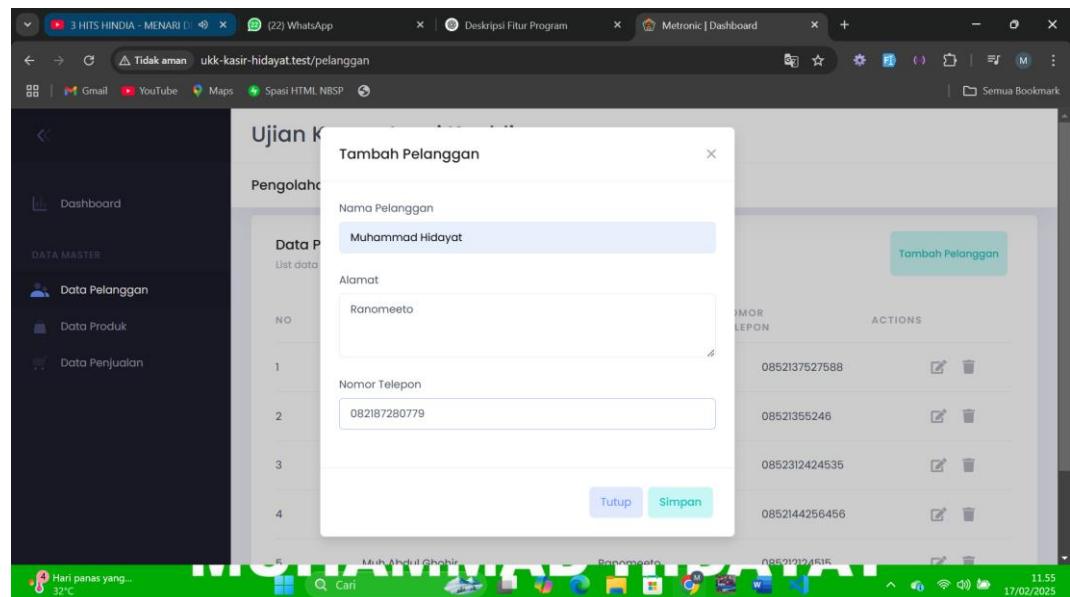
Gambar 3.1 Tampilan Halaman Dashboard

The screenshot shows the "Data Pelanggan" section of the application. The title is "Ujian Kompetensi Keahlian" and the subtitle is "Pengolahan Basis Data". A button "Tambah Pelanggan" is visible. The table lists four customers:

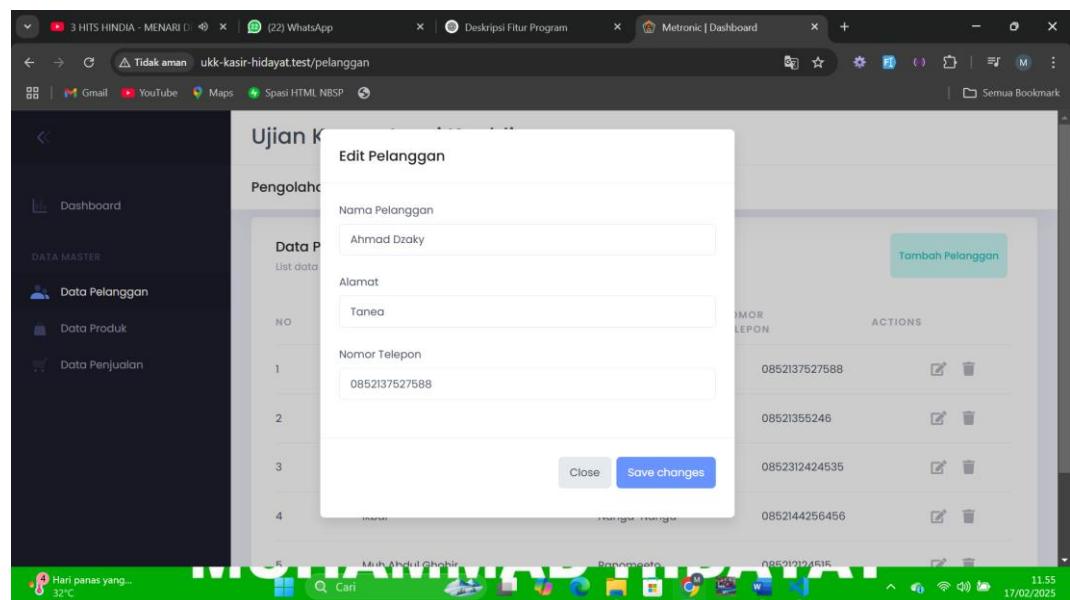
| NO | NAMA PELANGGAN   | ALAMAT      | NOMOR TELEPON | ACTIONS |
|----|------------------|-------------|---------------|---------|
| 1  | Ahmad Dzaky      | Tanea       | 0852137527588 |         |
| 2  | Arika Khairunisa | Ranomeeto   | 08521355246   |         |
| 3  | Andi Bungalia    | Baruga      | 0852312424535 |         |
| 4  | Ikbal            | Nanga-Nanga | 0852144256456 |         |

At the bottom, there's a weather widget showing "Hari panas yang..." and "32°C", along with a system tray showing various icons and the date/time "17/02/2025 11:54".

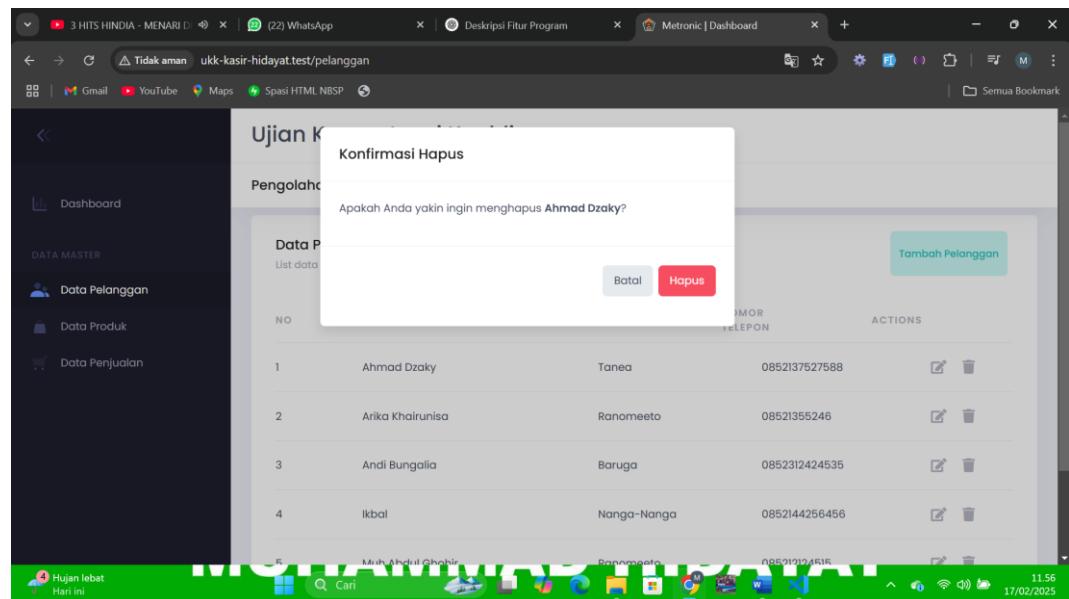
Gambar 3.2 Tampilan halaman data pelanggan



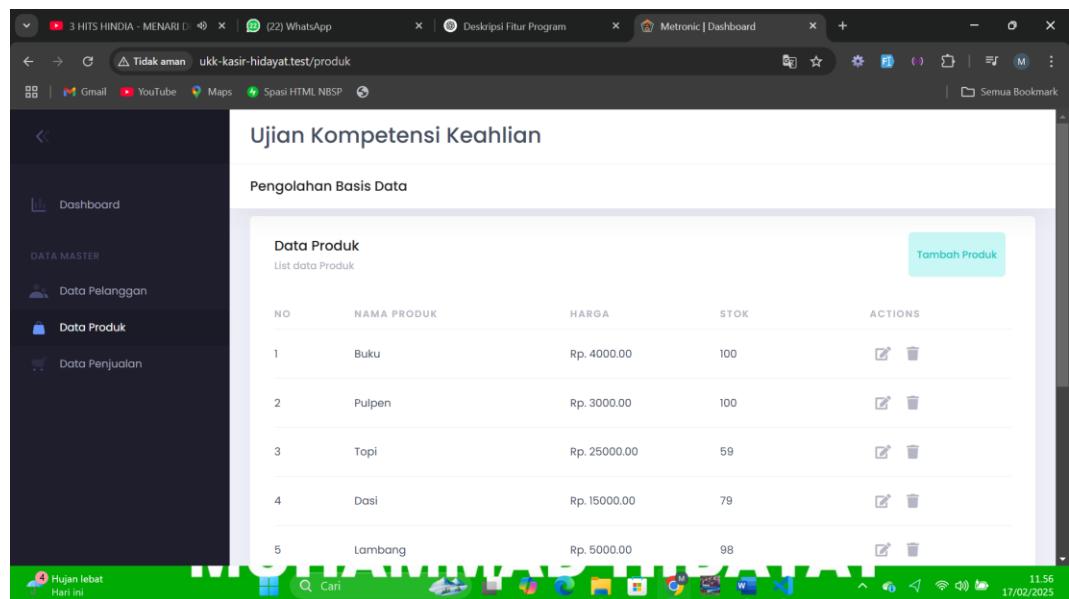
Gambar 3.3 Tampilan modal untuk menambahkan data pelanggan baru



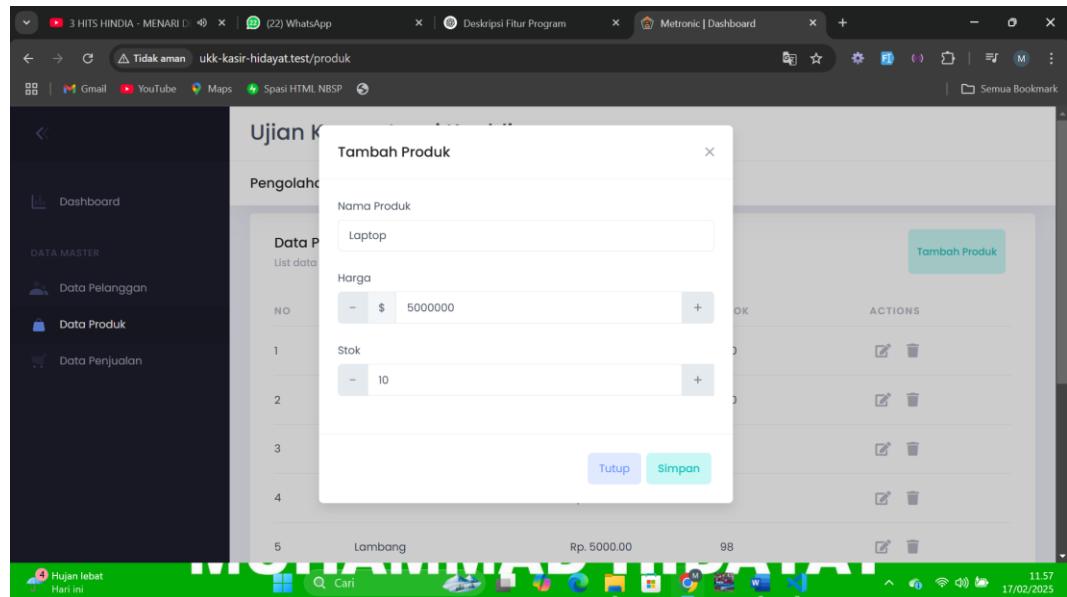
Gambar 3.4 Tampilan modal untuk menedit data pelanggan



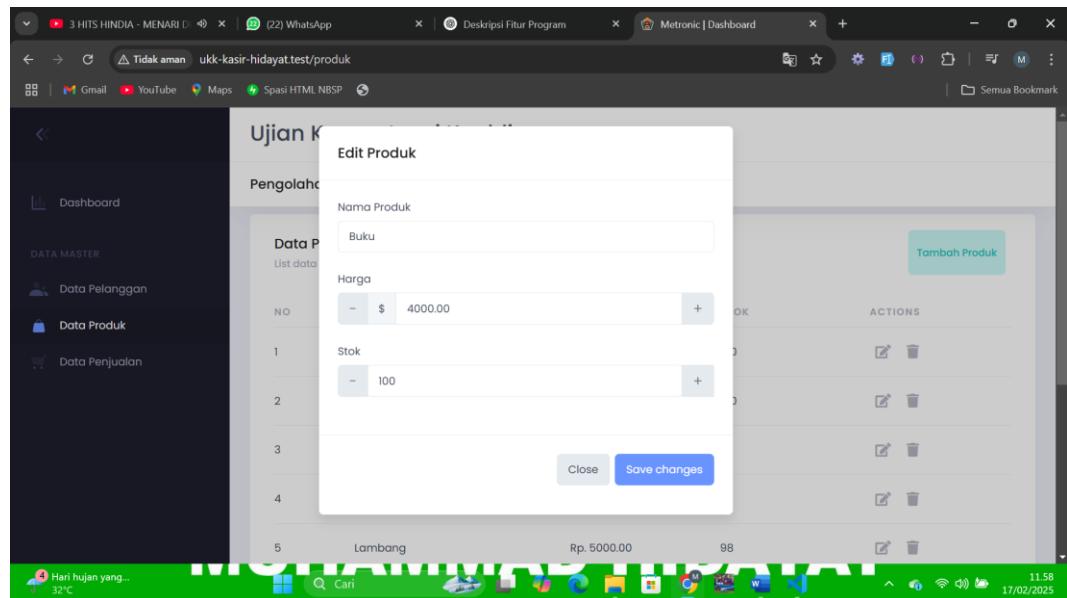
Gambar 3.5 Tampilan modal untuk menghapus data pelanggan



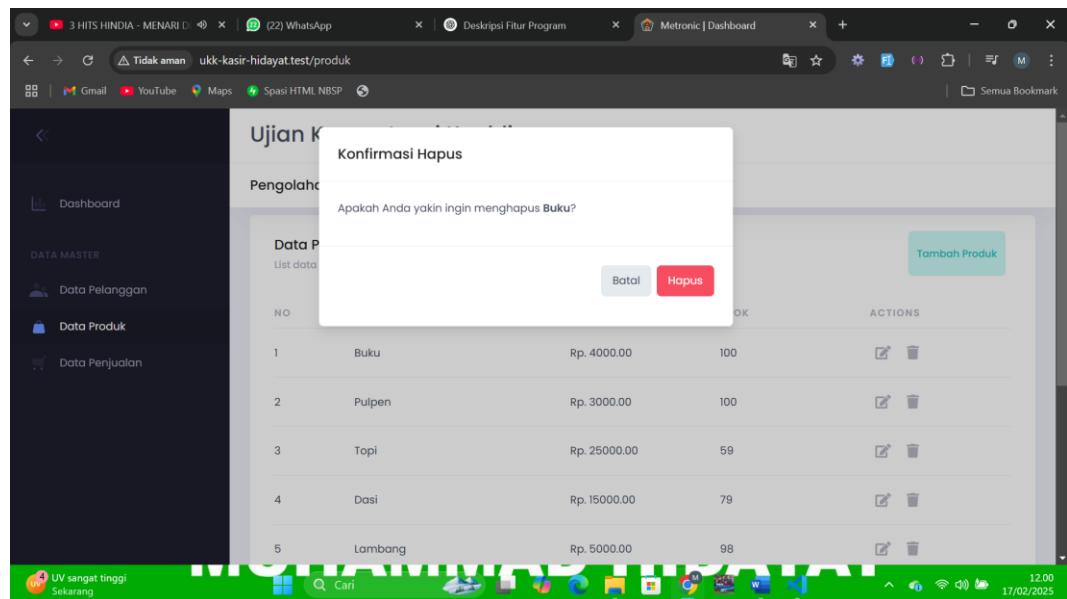
Gambar 3.6 Tampilan halaman data produk



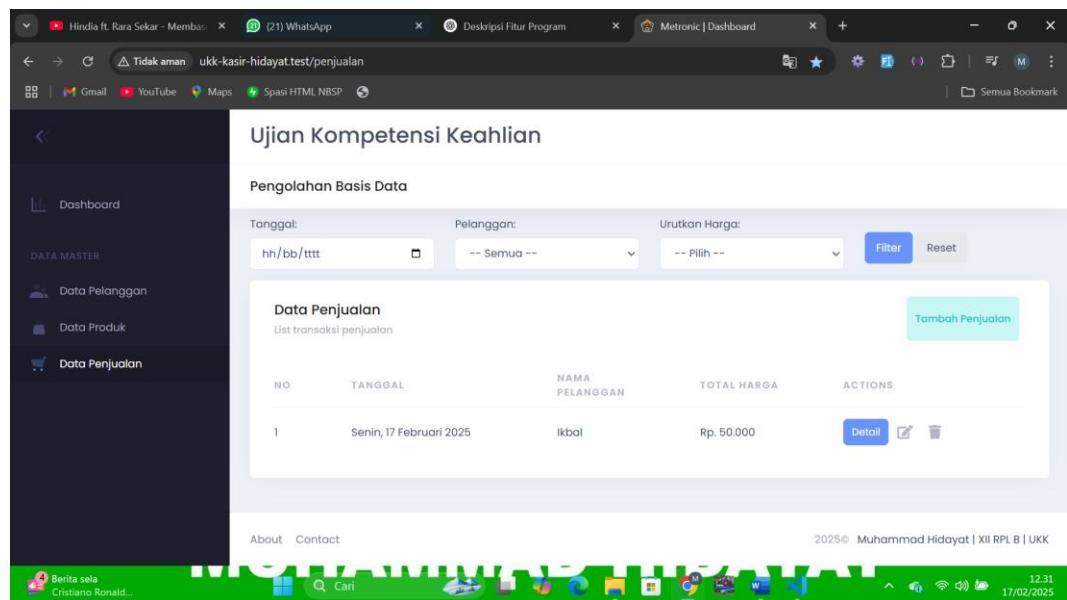
**Gambar 3.7** Tampilan modal untuk menambahkan data produk baru



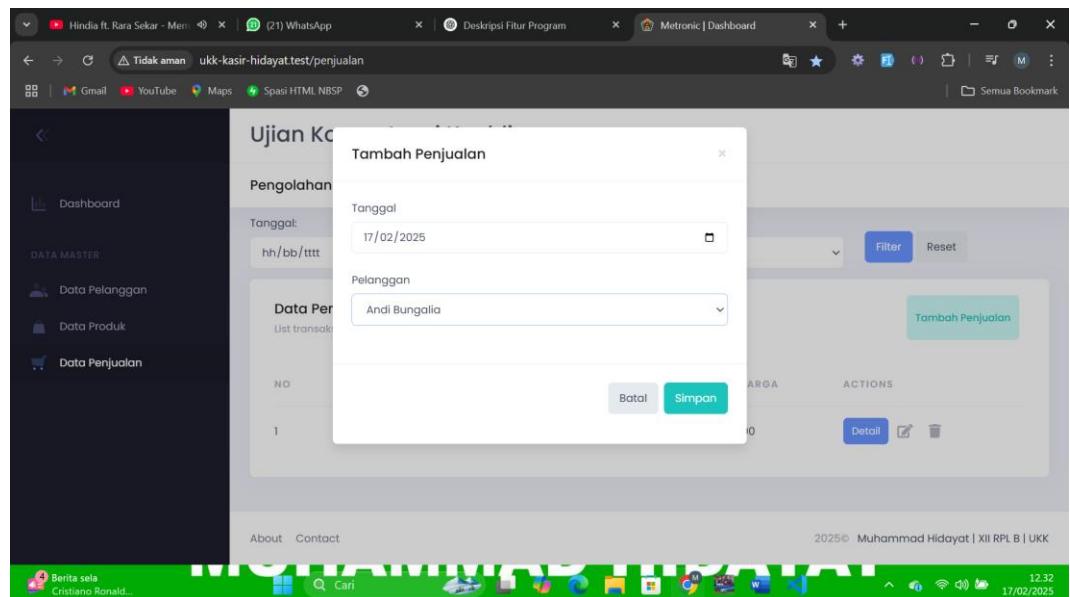
**Gambar 3.8** Tampilan modal untuk mengedit data produk



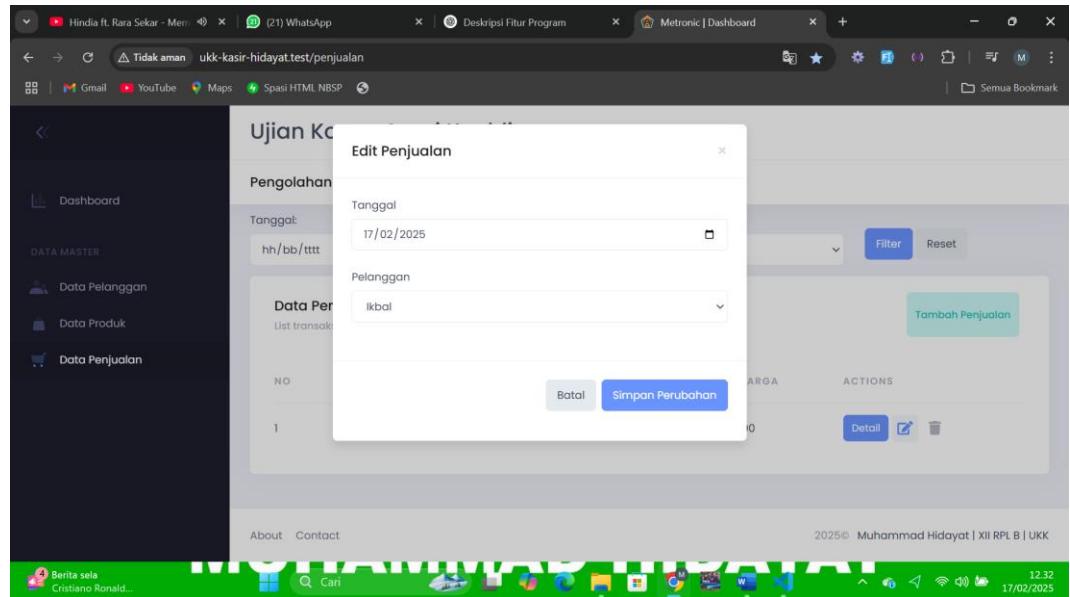
Gambar 3.9 Tampilan modal untuk menghapus data produk



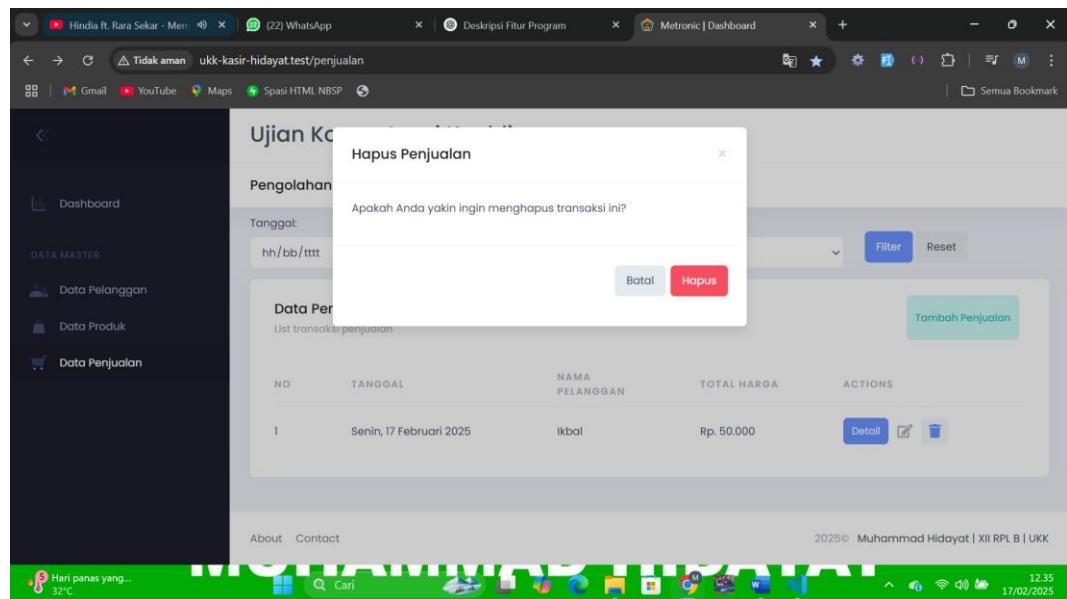
Gambar 3.8 Tampilan halaman transaksi penjualan



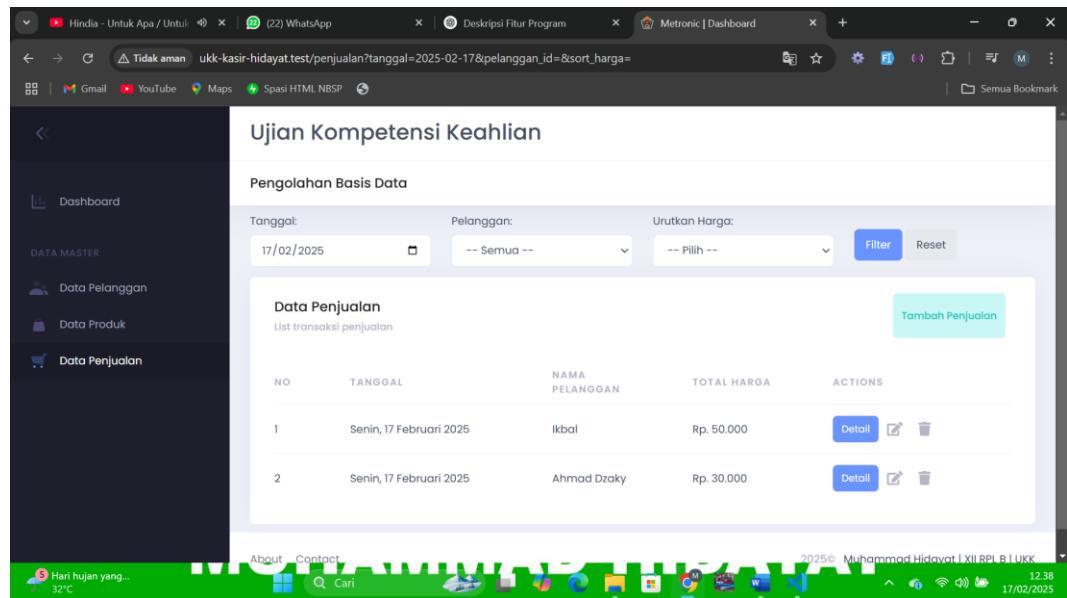
**Gambar 3.9** Tampilan modal untuk menambahkan transaksi penjualan baru



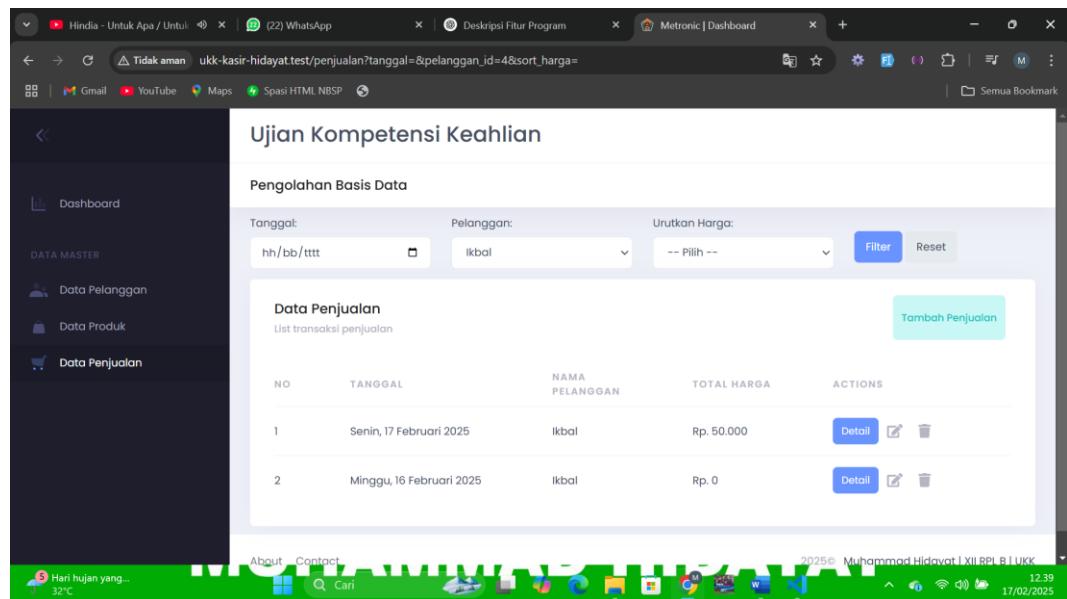
**Gambar 3.8** Tampilan modal untuk mengedit transaksi penjualan



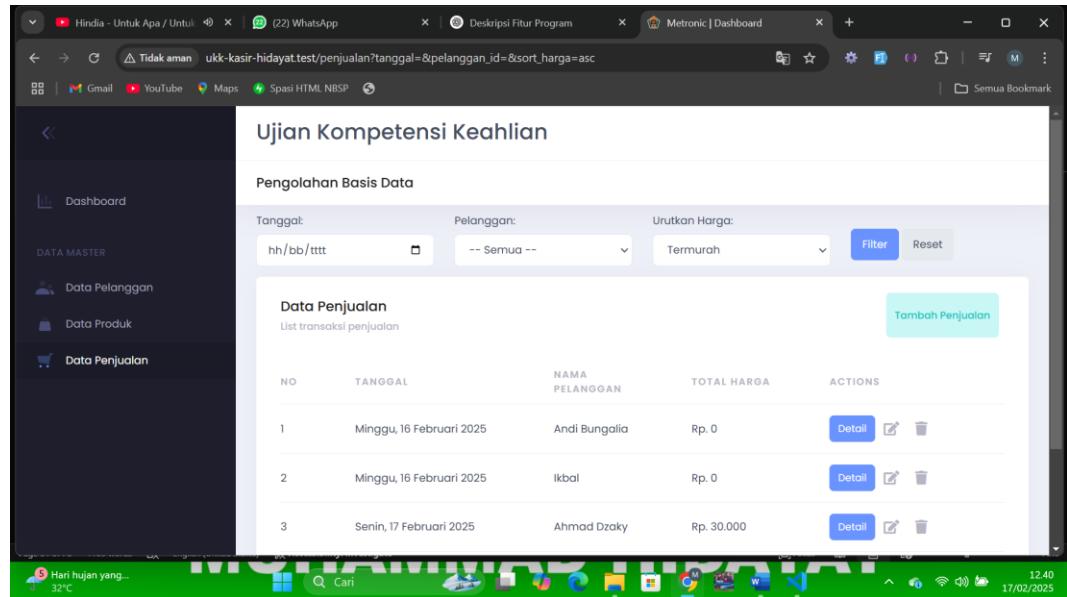
**Gambar 3.9** Tampilan modal untuk menghapus transaksi penjualan



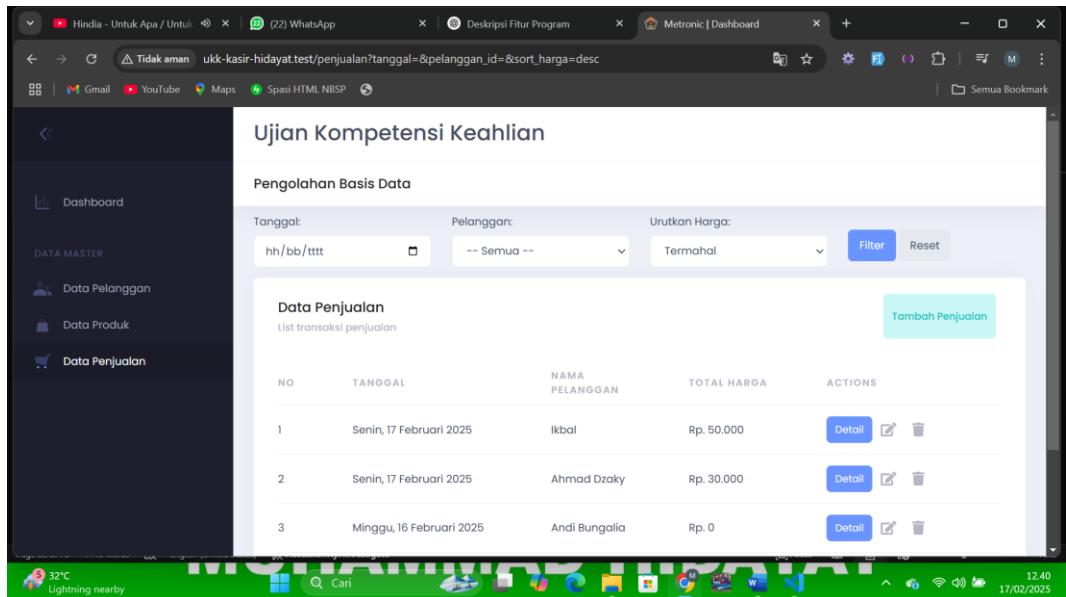
**Gambar 3.8** Tampilan halaman transaksi penjualan dengan filter tanggal tertentu



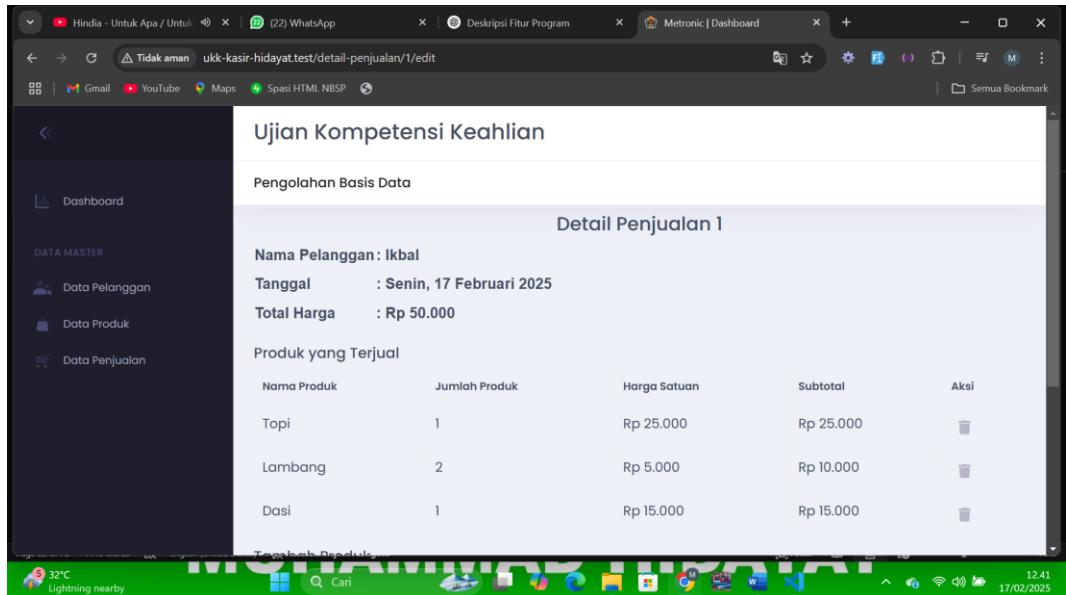
**Gambar 3.9** Tampilan halaman transaksi penjualan dengan filter pelanggan tertentu



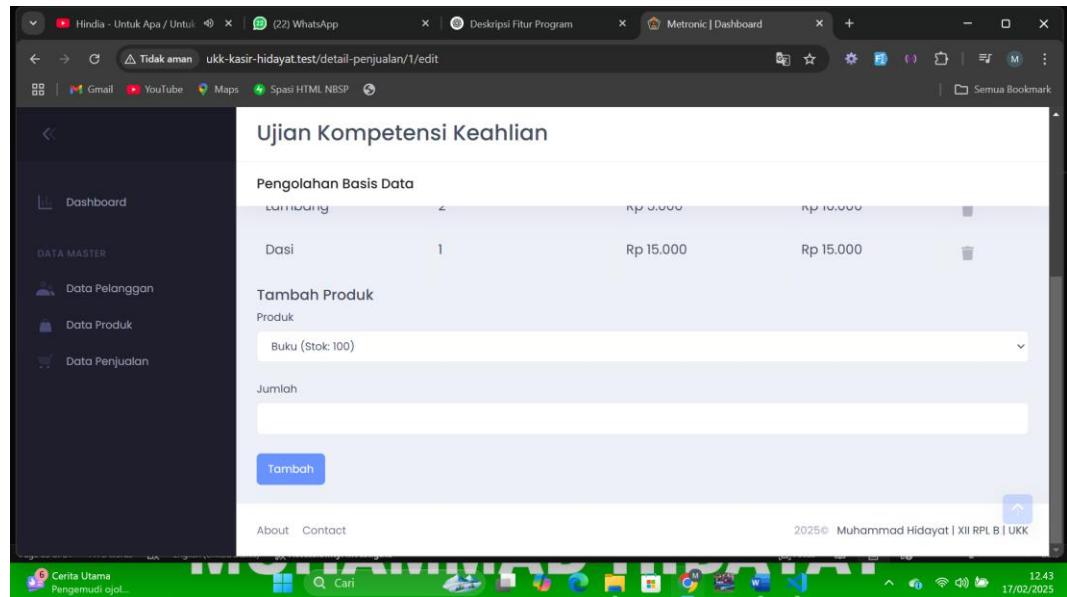
**Gambar 3.8** Tampilan halaman transaksi penjualan dengan filter harga dari yang termurah



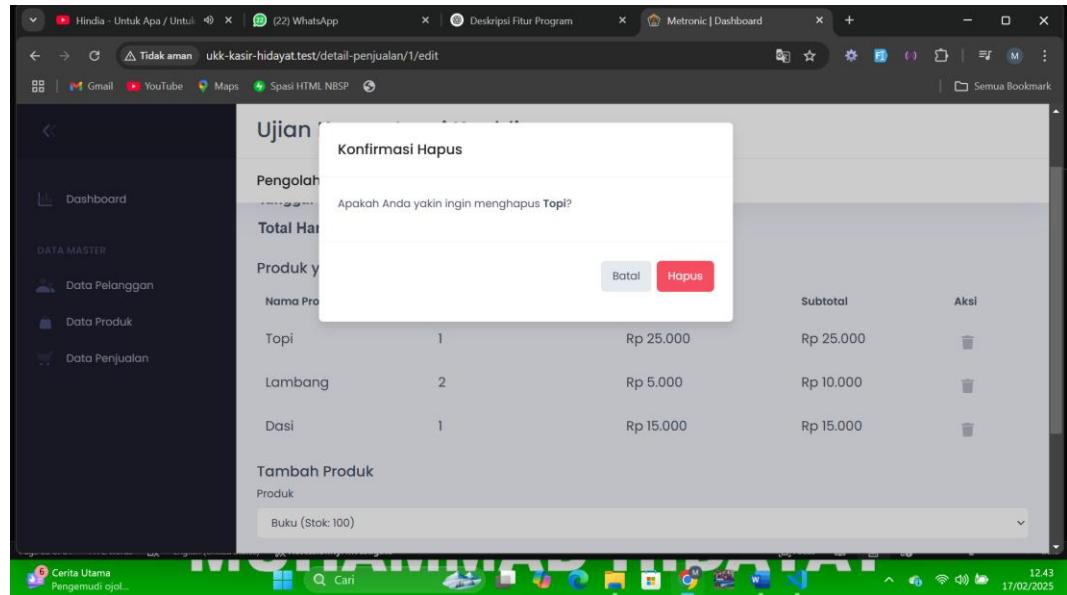
Gambar 3.9 Tampilan halaman transaksi penjualan dengan filter harga dari yang termahal



Gambar 3.8 Tampilan halaman detail penjualan



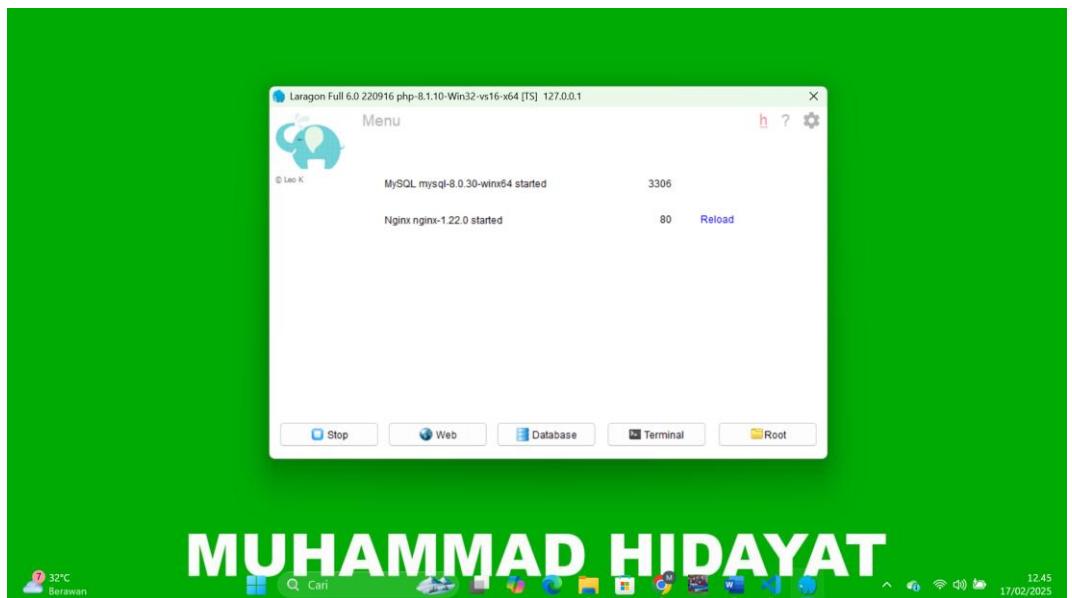
**Gambar 3.9** Tampilan form untuk menambahkan produk ke dalam detail penjualan



**Gambar 3.8** Tampilan modal untuk menghapus produk dari detail penjualan

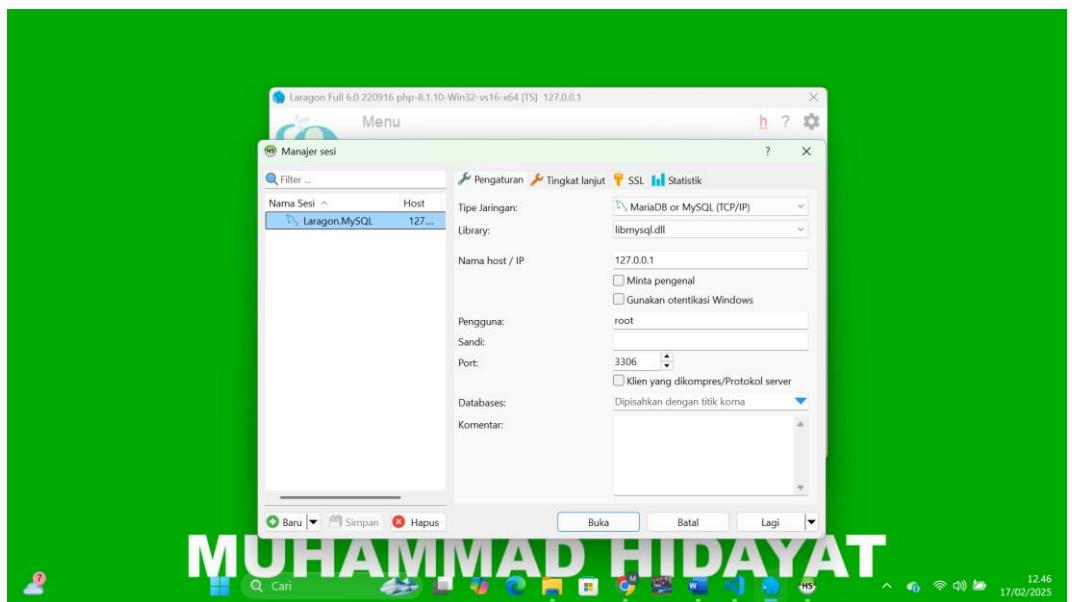
#### **4.1.8 Membuat database ukk\_kasir yang digunakan untuk menambah, mengedit, menghapus, dan melihat data pelanggan, produk, penjualan, dan detail penjualan di kasir sesuai gambar kerja**

- Menyalakan DBMS dan Web Server



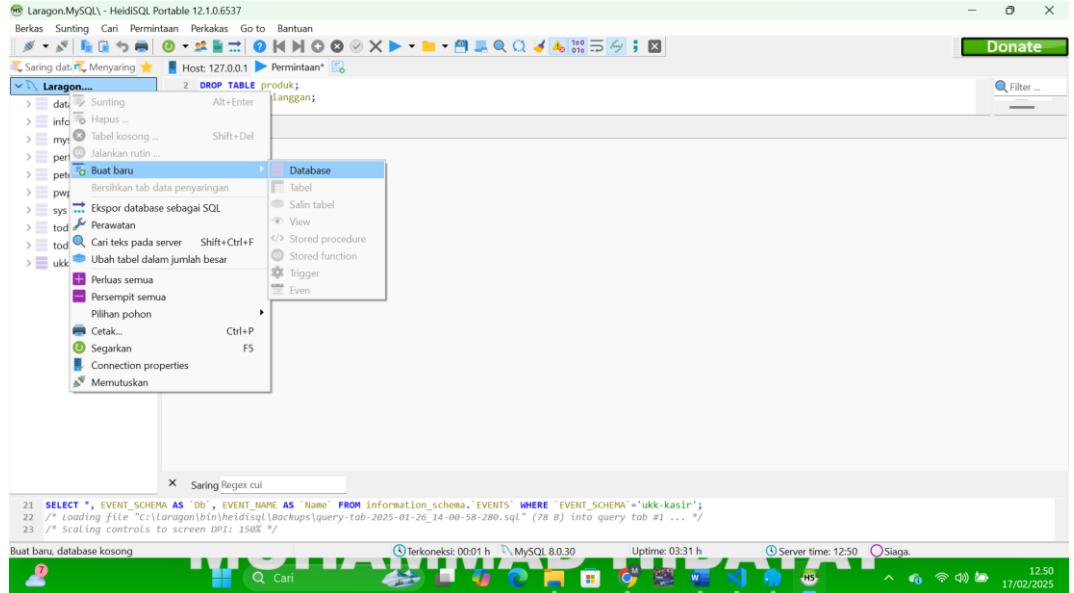
Menggunakan aplikasi penyedia server seperti Laragon untuk mengaktifkan **WebServer** (Nginx) dan **DBMS** (MySQL).

- Akses Aplikasi Manajemen Basis Data



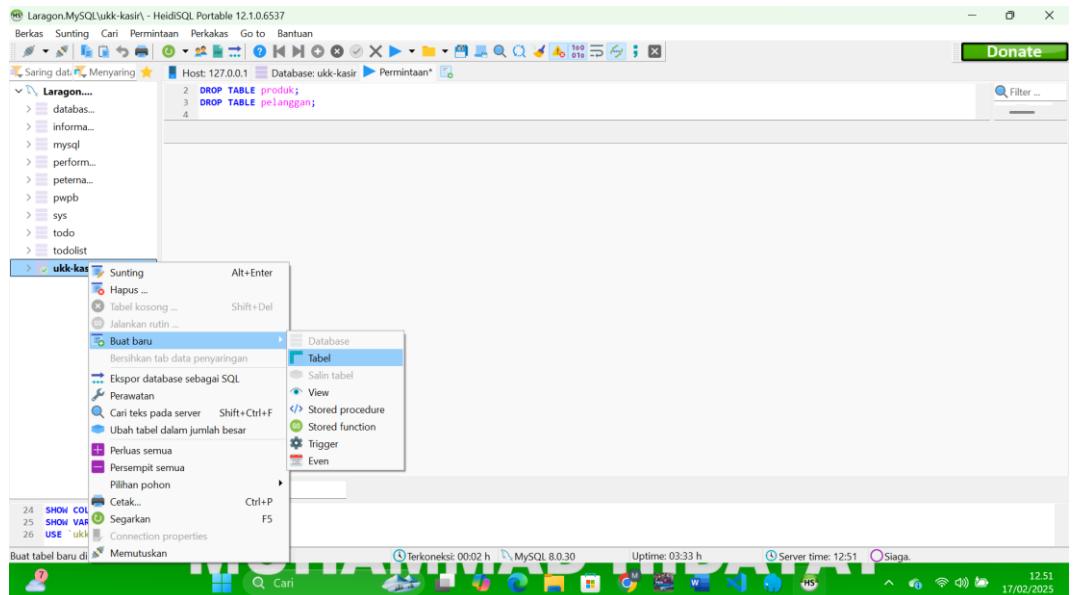
Membuka aplikasi HeidiSQL dengan menekan tombol Database di Laragon

- Membuat database



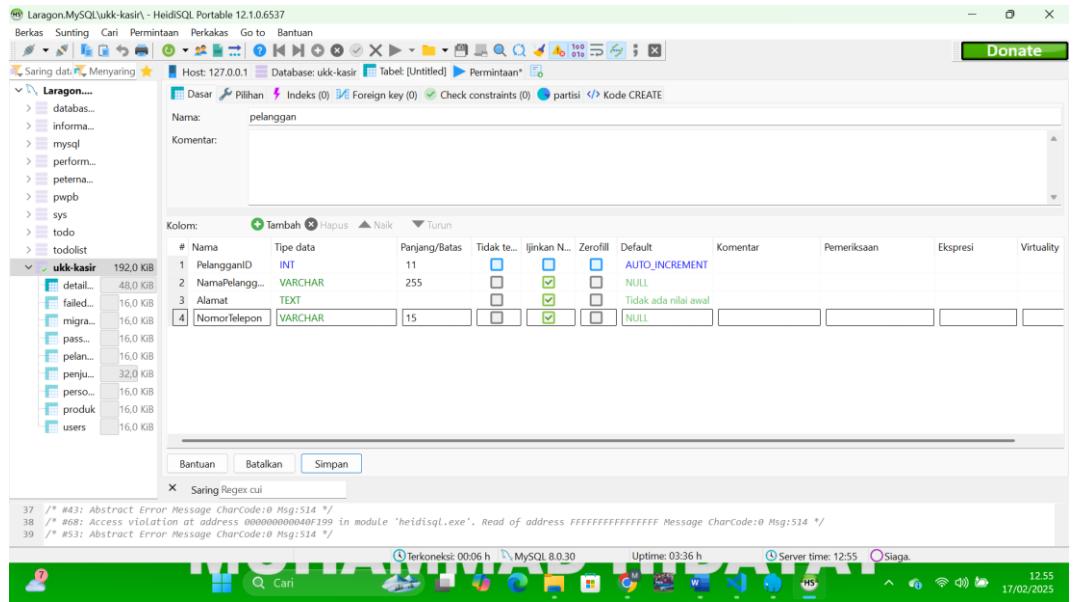
Klik kanan dibagian Side bar, lalu arahkan kursor ke bagian “Buat baru”, lalu pilih database, lalu masukkan nama database ukk\_kasir

- Membuat table



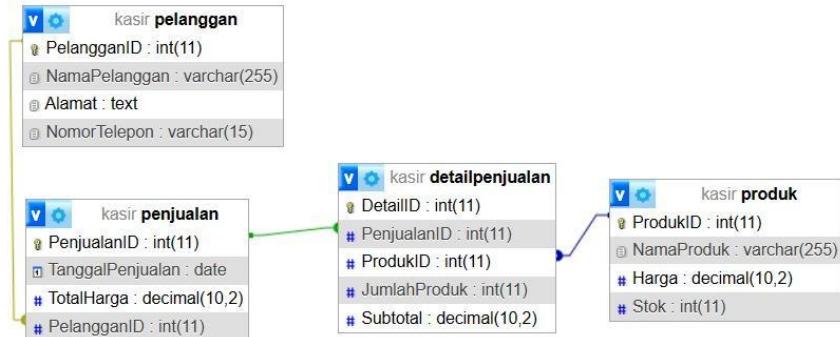
Klik kanan dibagian database ukk\_kasir, lalu kebagian buat baru , lalu pilih table, dan masukkan nama table pelanggan, dll

- Memberikan nama pada kolom table, setelah itu klik ‘simpan’



Mengisi nama kolom di dalam table, mengatur panjang karakter pada kolom, serta menetapkan kolom ‘PelangganID’ sebagai primary key.

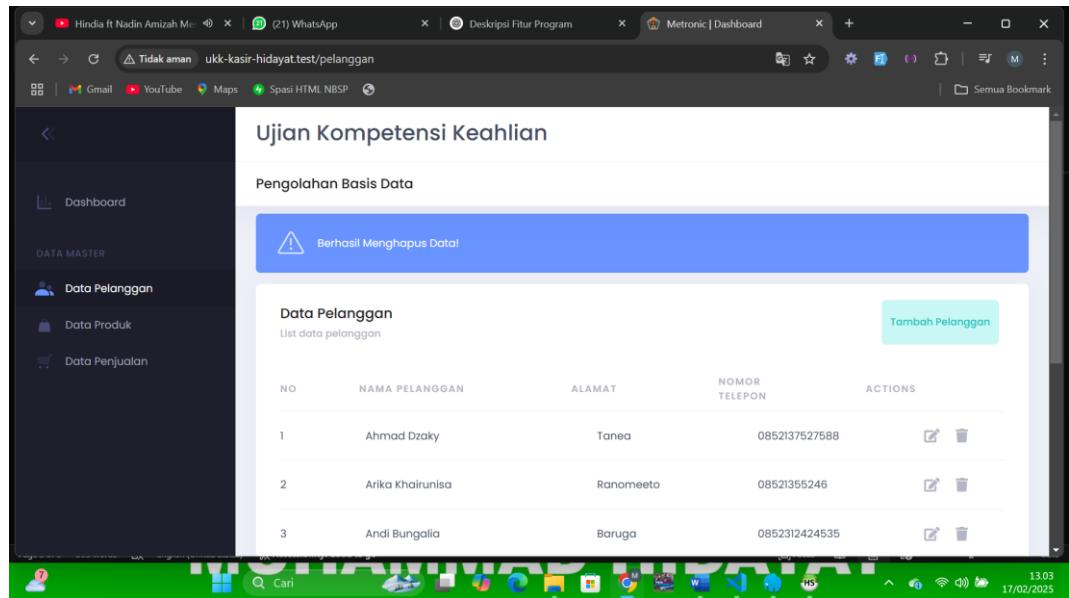
- Melanjutkan membuat table sesuai dengan lembar kerja



Lanjut membuat table sesuai dengan yang ada di lembar kerja, setelah itu menghubungkan antara *foreign key* dan *primary key* untuk membuat relasi antar table.

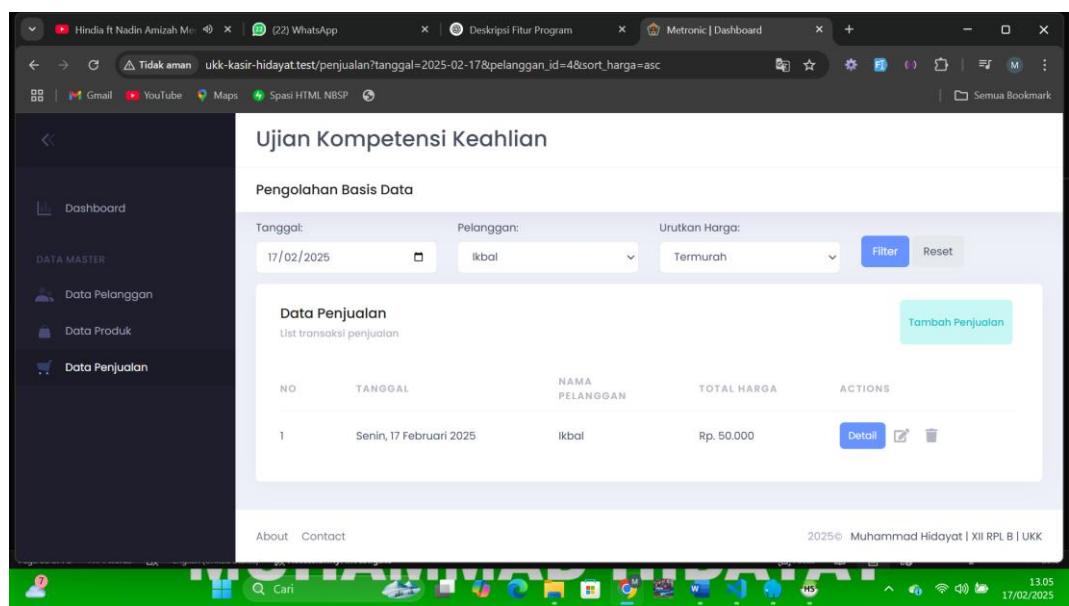
#### 4.1.9 Melakukan running dan testing

- Testing Fitur CRUD (Create, Read, Update, dan Delete) dari data Pelanggan, Produk, Penjualan, dan Detail Penjualan



Fitur CRUD berjalan dengan baik.

- Testing Fitur Filter data untuk halaman Transaksi Penjualan



Fitur Filter berjalan dengan baik.

- Testing Fitur untuk Perhitungan SubTotal Penjualan

| NO | TANGGAL                 | NAMA PELANGGAN | TOTAL HARGA | ACTIONS  |
|----|-------------------------|----------------|-------------|--|
| 1  | Senin, 17 Februari 2025 | Ikbal          | Rp. 50.000  | <button>Detail</button> <button>Print</button> <button>Delete</button> |

Fitur Perhitungan berjalan dengan baik.

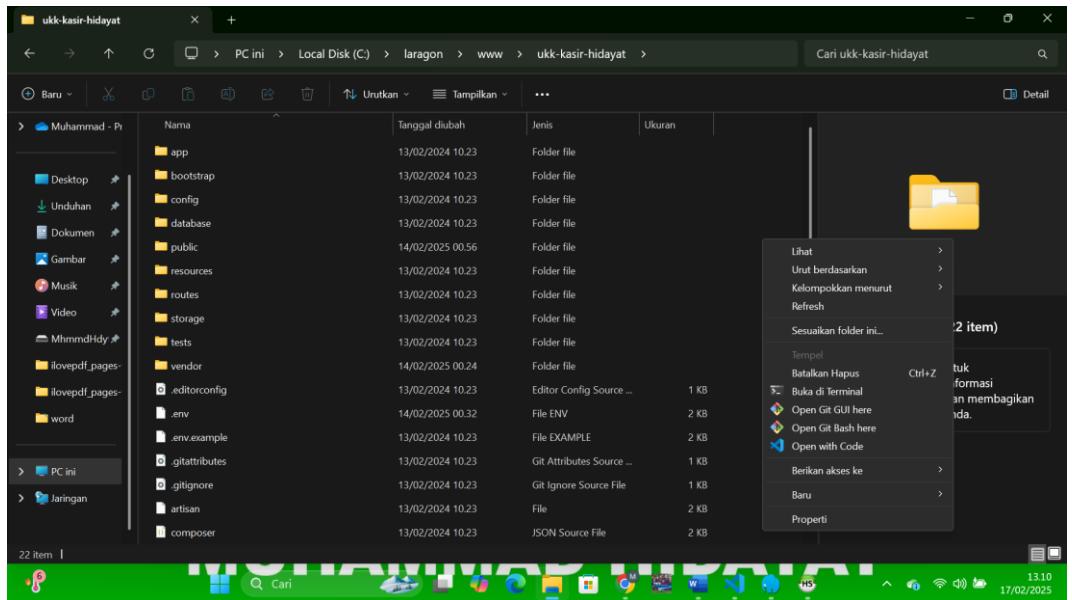
- Testing Fitur Pengurangan Stok setelah terjadi Transaksi Penjualan

| ID | Produk        | Harga        | Stok | ACTIONS                                       |
|----|---------------|--------------|------|---|
| 2  | Pulpen        | Rp. 3000.00  | 100  | <button>Edit</button> <button>Delete</button> |
| 3  | Topi          | Rp. 25000.00 | 59   | <button>Edit</button> <button>Delete</button> |
| 4  | Dasi          | Rp. 15000.00 | 79   | <button>Edit</button> <button>Delete</button> |
| 5  | Lambang       | Rp. 5000.00  | 98   | <button>Edit</button> <button>Delete</button> |
| 6  | Eskrim Jagung | Rp. 3000.00  | 15   | <button>Edit</button> <button>Delete</button> |
| 7  | Penggaris     | Rp. 5000.00  | 30   | <button>Edit</button> <button>Delete</button> |
| 8  | Penghapus     | Rp. 2000.00  | 100  | <button>Edit</button> <button>Delete</button> |

Fitur Pengurangan Stok berjalan dengan baik.

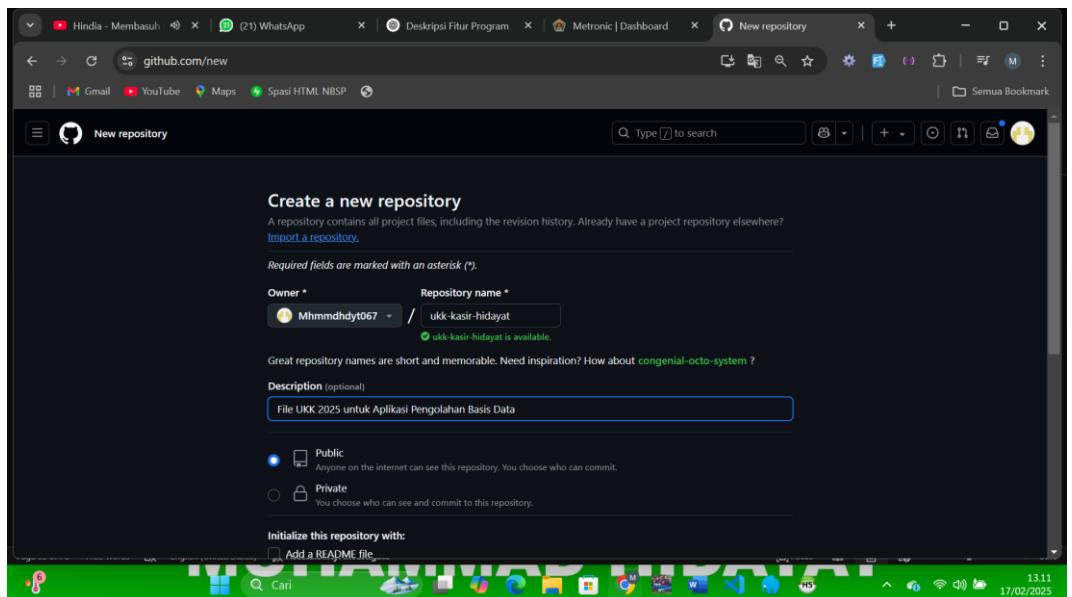
#### 4.1.10 Buatlah dokumentasi kode program dan upload ke github

- Memastikan bahwa aplikasi *Git* sudah terisntall di laptop/computer



Ketika git telah tersinstall di laptop/komputer, akan ada ‘Open Git Bash here’ di context menu.

- Repository



Membuat repository baru dengan nama “ukkan-kasir-hidayat”.

- Terminal

```
muhhi@MUHAMMAD-HIDAYAT MINGW64 /c/laragon/www/ukk-kasir-hidayat
$ git init
Initialized empty Git repository in C:/laragon/www/ukk-kasir-hidayat/.git/
$ |
```

Membuka terminal di root folder source code.

- Git init

```
muhhi@MUHAMMAD-HIDAYAT MINGW64 /c/laragon/www/ukk-kasir-hidayat
$ git init
Initialized empty Git repository in C:/laragon/www/ukk-kasir-hidayat/.git/
$ |
```

Membuat repository baru di directory lokal menggunakan perintah git init.

- Git add .

Memasukan semua source code kedalam staging are dengan menggunakan perintah ‘git add .’

- Git commit

The screenshot shows a Windows desktop environment. In the center, there is a file explorer window titled "ukk-kasir-hidayat" which displays the contents of a folder named "ukk-kasir-hidayat". The folder contains several files and subfolders, including "artisan", "composer", "env", "env.example", "gitattributes", "gitignore", "Kernel.php", "LICENSE", "README.md", "composer.json", "composer.lock", "KernelController.php", "PengulanganController.php", "PenjualanController.php", "ProdukController.php", "Authenticates.php", "EncryptCookies.php", "PreventRequestsDuringMaintenance.php", "RedirectIfAuthenticated.php", and "TrimStrings.php". A tooltip from a context menu indicates that the user has selected multiple items. On the left side of the screen, there is a sidebar with icons for "PC ini", "Jaringan", and "23 item". At the bottom, the taskbar shows various pinned icons and the system tray. To the right of the file explorer, a terminal window titled "MINGW64/c/laragon/www/ukk-kasir-hidayat" is open, showing the command "git commit -m 'first commit'" and its output. The output includes a commit message "first commit", a timestamp "2024-02-13T10:23:00+07:00", and a commit hash "cd0c77b1". The commit message also lists the changes made: "2067 files changed, 481901 insertions(+)".

Menyimpan perubahan pertama yang telah ditambahkan ke dalam staging area ke dalam repositori Git dengan pesan “first commit”.

- Git remote

```
muhhi@MUHAMMAD-HIDAYAT MINGW64 /c/laragon/www/ukk-kasir-hidayat (master)
$ ^[[200~
bash: $^[[200~: command not found
> muhhi@MUHAMMAD-HIDAYAT MINGW64 /c/laragon/www/ukk-kasir-hidayat (master)
$ git remote add origin https://github.com/Mhmdhdty067/ukk-kasir-hidayat.git
> muhhi@MUHAMMAD-HIDAYAT MINGW64 /c/laragon/www/ukk-kasir-hidayat (master)
```

ukk-kasir-hidayat (23 item)

Pilih satu file untuk mendapatkan informasi selengkapnya dan membagikan konten cloud Anda.

Menghubungkan repository lokal dengan repository jarak jauh (remote).

- Git push

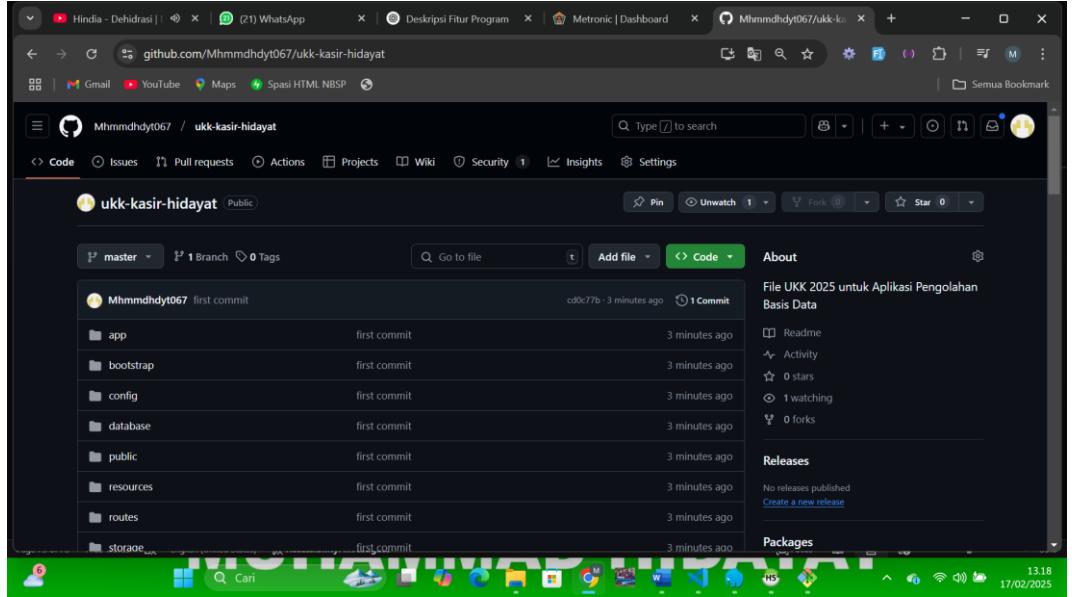
```
muhhi@MUHAMMAD-HIDAYAT MINGW64 /c/laragon/www/ukk-kasir-hidayat (master)
$ ^[[200~
bash: $^[[200~: command not found
> muhhi@MUHAMMAD-HIDAYAT MINGW64 /c/laragon/www/ukk-kasir-hidayat (master)
$ git remote add origin https://github.com/Mhmdhdty067/ukk-kasir-hidayat.git
> muhhi@MUHAMMAD-HIDAYAT MINGW64 /c/laragon/www/ukk-kasir-hidayat (master)
$ git push -u origin master
 Enumerating objects: 2331, done.
  Compressing objects: 100% (2260/2260), done.
  Delta compression using up to 8 threads
  (Compressing objects: 100% (2231/2231)), done.
  Writing objects: 100% (2260/2260) 23.26 MiB | 3.06 MiB/s, done.
    Total 2260 (delta 1111), reused 0 (delta 0) from 0
  remote: Resolving deltas: 100% (509/509), done.
  To https://github.com/Mhmdhdty067/ukk-kasir-hidayat.git
    * [new branch]      master    -> master
  branch 'master' set up to track 'origin/master'.
> muhhi@MUHAMMAD-HIDAYAT MINGW64 /c/laragon/www/ukk-kasir-hidayat (master)
```

ukk-kasir-hidayat (23 item)

Pilih satu file untuk mendapatkan informasi selengkapnya dan membagikan konten cloud Anda.

Mengirimkan perubahan yang telah dicommit dari repositori lokal ke repositori jarak jauh.

- Upload selesai



Upload source code ke github berhasil, Kembali ke halaman repository dan refresh halaman untuk melihat hasil.

## **4.2 Pembahasan**

### **4.2.1 Manajemen Data Pelanggan**

Fitur ini bertujuan untuk mempermudah pengelolaan data pelanggan yang ada dalam system, kita dapat menambah, mengedit, menghapus, dan melihat data pelanggan yang ada, yang mencakup informasi penting seperti nama pelanggan, alamat, dan nomor telepon.

### **4.2.2 Daftar Data Produk**

Fitur ini digunakan untuk menampilkan daftar seluruh produk yang tersedia untuk dijual. Setiap produk dilengkapi dengan informasi seperti nama produk, harga jual, dan stok yang tersedia,. Fitur ini juga memungkinkan untuk menambahkan produk baru, mengedit data produk yang ada, serta menghapus produk yang sudah tidak tersedia atau tidak terjual lagi. Dengan daftar data produk, kita dapat memantau ketersediaan produk dan melakukan pengelolaan produk dengan lebih mudah.

### **4.2.3 Pencatatan Transaksi Penjualan**

Fitur pencatatan transaksi penjualan berguna untuk mencatat setiap transaksi yang terjadi. Fitur ini menyimpan informasi tentang tanggal transaksi, dan total harga yang dibayar. Fitur ini mempermudah administrasi dalam mencatat penjualan secara otomatis dan efisien, serta menjadi dasar untuk laporan transaksi keuangan yang lebih akurat.

### **4.2.4 Riwayat Detail Penjualan**

Fitur ini menyajikan data lengkap mengenai riwayat penjualan, memungkinkan pengguna untuk melihat rincian transaksi yang telah dilakukan. detail penjualan mencakup informasi seperti tanggal transaksi, nama pelanggan, produk yang dibeli, jumlah dan harga produk, serta total harga yang dibayarkan. Fitur ini berguna untuk melacak semua transaksi yang sudah dilakukan dan memberikan informasi lebih mendalam tentang setiap penjualan yang terjadi.

#### **4.2.5 Pencarian Data Penjualan Berdasarkan Tanggal**

Fitur pencarian berdasarkan tanggal memungkinkan pengguna untuk mencari dan menampilkan data transaksi penjualan yang terjadi dalam tanggal tertentu. Pengguna bisa menentukan tanggal untuk memfilter transaksi yang terjadi di hari itu. Fitur ini memudahkan pencarian data penjualan berdasarkan waktu, sehingga pengguna dapat dengan cepat mendapatkan informasi yang relevan untuk analisis atau laporan tertentu.

#### **4.2.6 Pencarian Data Penjualan Berdasarkan Pelanggan**

Fitur pencarian berdasarkan pelanggan memudahkan pengguna untuk mencari riwayat transaksi yang dilakukan oleh pelanggan tertentu. Dengan memilih nama pelanggan, pengguna bisa melihat semua transaksi yang dilakukan oleh pelanggan tersebut. Fitur ini sangat berguna untuk melacak riwayat pembelian pelanggan, memberikan pelayanan yang lebih baik, atau untuk analisis lebih lanjut mengenai pola pembelian pelanggan tertentu.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

1. Memastikan siswa memiliki akses untuk mendapatkan sertifikat kompetensi yang diperlukan sebagai bukti kemampuan mereka dalam bidang keahlian tertentu.
2. Memberikan penilaian yang komprehensif terhadap praktik kompetensi kejuruan siswa yang telah mengikuti pembelajaran produktif selama periode waktu tertentu, sehingga dapat menilai kemampuan mereka secara praktis.
3. Memberikan pemahaman menyeluruh kepada siswa tentang proses pengembangan aplikasi pengolahan basis data yang dirangkaikan kedalam aplikasi kasir, yang merupakan aplikasi konkret dari pengetahuan dan keterampilan yang mereka pelajari dalam bidang teknologi informasi.
4. Memberikan panduan praktis dan langkah-langkah yang jelas dalam pembuatan aplikasi kasir (Pengolahan Basis Data) sesuai dengan Standar Operasional Prosedur (SOP) yang baik dan benar, sehingga siswa dapat menerapkan pengetahuan dan keterampilan mereka dalam situasi praktis.

Dengan demikian, penyelenggaraan UKK diharapkan dapat memberikan manfaat yang signifikan bagi siswa dalam mencapai sertifikasi kompetensi, penilaian praktik kompetensi kejuruan, pemahaman proses pengembangan aplikasi, dan panduan praktis dalam penerapan pengetahuan dan keterampilan dalam pembuatan aplikasi.

#### **5.2 Saran**

1. Untuk adik-adik kelas yang akan mengikuti Ujian Kompetensi Kejuruan (UKK) di tahun depan, disarankan untuk meningkatkan upaya belajar mengenai pemrograman. Selain belajar di sekolah, sangat dianjurkan untuk mulai belajar secara mandiri melalui sumber-sumber di internet.

2. Untuk melihat dokumentasi kode program yang lebih jelas dan lengkap lagi, silahkan kunjungi halaman repository github saya di :

<https://github.com/Mhmmdhdyt067/ukk-kasir-hidayat>

## DAFTAR PUSTAKA

Wikipedia. (n.d.). *PHP*. Diakses pada 17 Februari 2025, dari <https://id.wikipedia.org/wiki/PHP>

W3Schools. (n.d.). *PHP Tutorial*. Diakses pada 17 Februari 2025, dari <https://www.w3schools.com/php/>

Wikipedia. (n.d.). *JavaScript*. Diakses pada 17 Februari 2025, dari <https://id.wikipedia.org/wiki/JavaScript>

Mozilla Developer Network (MDN). (n.d.). *JavaScript*. Diakses pada 17 Februari 2025, dari <https://developer.mozilla.org/id/docs/Web/JavaScript>

W3Schools. (n.d.). *JavaScript Tutorial*. Diakses pada 17 Februari 2025, dari <https://www.w3schools.com/js/>

Laragon. (n.d.). *Laragon Website*. Diakses pada 17 Februari 2025, dari <https://laragon.org/>

Wikipedia. (n.d.). *MySQL*. Diakses pada 17 Februari 2025, dari <https://id.wikipedia.org/wiki/MySQL>

MySQL Documentation. (n.d.). *MySQL Documentation*. Diakses pada 17 Februari 2025, dari <https://dev.mysql.com/doc/>

W3Schools. (n.d.). *MySQL Tutorial*. Diakses pada 17 Februari 2025, dari <https://www.w3schools.com/sql/>

Nginx. (n.d.). *Nginx Website*. Diakses pada 17 Februari 2025, dari <https://nginx.org/>

Nginx. (n.d.). *Nginx Documentation*. Diakses pada 17 Februari 2025, dari <https://nginx.org/en/docs/>

Sterling Team. (2022). *Apa itu Aplikasi Kasir (Point of Sale)?* Diakses pada 17 Februari 2025, dari [https://www.sterling-team.com/news/apa-itu-aplikasi-kasir-point-of-sale-ireap\\_pos/#:~:text=Aplikasi%20kasir%20atau%20Point%20of%20Sale%20\(POS\)%20adalah%20perangkat%20lunak,yang%20lebih%20efisien%20dan%20akurat.](https://www.sterling-team.com/news/apa-itu-aplikasi-kasir-point-of-sale-ireap_pos/#:~:text=Aplikasi%20kasir%20atau%20Point%20of%20Sale%20(POS)%20adalah%20perangkat%20lunak,yang%20lebih%20efisien%20dan%20akurat.)

Laravel. (n.d.). Laravel Documentation (Versi 11.x). Diakses pada 17 Februari 2025, dari <https://laravel.com/docs/11.x>

Keenthemes. (n.d.). Metronic Tailwind Documentation. Diakses pada 17 Februari 2025, dari <https://keenthemes.com/metronic/tailwind/docs/>

## TENTANG PENULIS



**Muhammad Hidayat**, lahir di Kendari pada 6 Mei 2007, adalah seorang pelajar semester akhir di SMK Negeri 6 Kendari, mengambil program keahlian Rekayasa Perangkat Lunak (RPL). Sejak duduk di bangku Sekolah Dasar, ia telah memiliki ketertarikan yang besar terhadap dunia komputer dan teknologi, yang terus ia tekuni hingga saat ini. Selain menempuh pendidikan formal, ia juga bekerja part-time sebagai Pelatih organisasi Praja Muda Karana (Pramuka) di salah satu Sekolah Dasar di sekitar tempat tinggalnya. Selain itu, ia juga menawarkan jasa Scan dan Pengetikan, yang telah menjadi salah satu keahliannya. Pada tahun 2022, ia mendapatkan kesempatan untuk mengikuti berbagai perlombaan, seperti Lomba Kompetensi Siswa Bidang Lomba Komputasi Awan hingga ke tingkat nasional serta Lomba Karya Jurnalistik tingkat kota. Pengalaman ini semakin memperluas wawasannya dalam dunia teknologi dan media. Selain itu, ia juga telah menjalani Praktik Kerja Lapangan (PKL) selama tiga bulan di sebuah perusahaan yang bergerak di bidang pembuatan website dan IT consulting. Selama menjalani PKL, ia mempelajari pengembangan website berbasis framework Laravel dan berhasil menyelesaikan proyek "Database Nilai Peternakan", sebuah website yang digunakan oleh Fakultas Peternakan Universitas Haluoleo. Saat ini, Muhammad Hidayat masih terus mendalami dunia pemrograman, khususnya dalam pengembangan website, dan sedang menjalani Ujian Kompetensi Keahlian (UKK) untuk menguji serta mengasah kemampuannya di bidang teknologi informasi. Dengan semangat belajar yang tinggi, ia terus berupaya mengembangkan keterampilannya agar dapat berkontribusi lebih dalam dunia teknologi.