

## **BAB II**

### **PELAKSANAAN UJIAN KOMPETENSI**

#### **2.1 Topik yang Dipelajari**

Materi yang dipelajari saat melaksanakan Uji Kompetensi dan Sertifikasi Web Developer meliputi :

- a. Menganalisis tools
- b. Menganalisis skalabilitas perangkat lunak
- c. Melakukan identifikasi library, komponen atau framework yang diperlukan
- d. Menggunakan struktur data
- e. Menerapkan perintah eksekusi bahasa pemrograman berbasis teks, grafik dan multimedia
- f. Melakukan instalasi software tools pemrograman
- g. Melakukan pengaturan software tools pemrograman
- h. Menerapkan metode dan praktik penggunaan kembali (Reusable) Subrutin-Subrutin
- i. Menulis kode dengan prinsip sesuai guidelines dan best practice
- j. Mengimplementasikan pemrograman berorientasi objek
- k. Menggunakan library atau komponen pre-existing
- l. Mengimplementasikan algoritma pemrograman
- m. Membuat dokumen kode program
- n. Melakukan debugging
- o. Melaksanakan pengujian unit program
- p. Melaksanakan pengujian oleh pengguna (UAT)
- q. Membuat paket instalasi perangkat lunak
- r. Melakukan logging aplikasi

#### **2.2 Tujuan**

Tujuan yang diharapkan ketika mengikuti Uji Kompetensi & Sertifikasi Web Developer ini meliputi :

- a. Mengukur kemampuan secara objektif:
  - Mengukur dan mengetahui sejauh mana penguasaan skill Web Developer yang dimiliki
  - Mendapat penilaian independent dari asesor bersertifikat
- b. Standarisasi kemampuan:
  - Memastikan kemampuan sesuai dengan standar industri nasional
  - Mengacu kepada Standar Kompetensi Kerja Nasional Indonesia (SKKNI)
- c. Akselerasi karier:
  - Membuka peluang posisi yang lebih tinggi
  - Mempercepat proses promosi dan kenaikan jabatan
- d. Pembelajaran dan pengembangan
  - Penguasaan teknologi design thinking yang tepat
  - Aktualisasi dengan trend dan penerapan praktek terkini

### 2.3 Manfaat

Manfaat yang diperoleh setelah mengikuti Uji Sertifikasi Kompetensi Web Developer BNSP yang diselenggarakan oleh Jobhun.id:

- a. Menganalisis tools, skalabilitas perangkat lunak dan identifikasi library atau framework
- b. Menggunakan struktur data dan menerapkan perintah eksekusi
- c. Melakukan instalasi dan pengaturan software tools pemograman
- d. Menerapkan metode dan praktik penggunaan kembali (Reusable)
- e. Menulis kode dengan prinsip sesuai guidelines dan best practice
- f. Mengimplementasikan pemograman berorientasi objek dan menggunakan library
- g. Mengimplementasikan algoritma pemograman
- h. Melakukan debugging

- i. Melaksanakan pengujian unit program dan pengujian oleh pengguna (UAT)
- j. Membuat paket instalasi perangkat lunak
- k. Melakukan logging aplikasi

## 2.4 Silabus

Silabus Uji Kompetensi dapat dilihat pada Tabel 1.1 Tabel Silabus

**Table 1.1. Tabel Silabus**

No	Kode Unit	Unit Kompetensi
1	J.620100.001.01	Menganalisis tools
2	J.620100.002.01	Menganalisis skalabilitas perangkat lunak
3	J.620100.003.01	Melakukan identifikasi library, komponen atau framework yang diperlukan
4	J.620100.020.02	Menggunakan SQL
5	J.620100.022.02	Mengimplementasikan algoritma pemrograman
6	J.620100.024.02	Melakukan migrasi ke teknologi baru
7	J.620100.025.02	Melakukan Debugging
8	J.620100.028.02	Menerapkan pemrograman real time
9	J.620100.029.02	Menerapkan pemrograman parallel
10	J.620100.030.02	Menerapkan pemrograman multimedia
11	J.620100.038.01	Melaksanakan pengujian oleh pengguna (UAT)
12	J.620100.039.02	Memberikan petunjuk teknis kepada pelanggan
13	J.620100.041.01	Melaksanakan cutover aplikasi
14	J.620100.043.01	Menganalisis dampak perubahan terhadap aplikasi
15	J.620100.044.01	Menerapkan alert notification jika aplikasi bermasalah
16	J.620100.045.01	Melakukan pemantauan resource yang digunakan aplikasi
17	J.620100.047.01	Melakukan pembaharuan perangkat lunak
18	J.62090.018.01	Mengelola Risiko Keamanan Informasi
19	M.702090.001.01	Mengelola Proyek Secara Terintegrasi (Project Integration Management)
20	M.702090.002.01	Mengelola Ruang Lingkup Proyek (Project Scope Management)
21	M.702090.005.01	Mengelola kualitas proyek (project quality management)
22	TIK.SM03.001.01	Menentukan arsitektur perangkat keras

## 2.5 Bimbingan Teknis

Bimbingan teknis Uji Kompetensi Sertifikasi BNSP Batch 4 dilaksanakan pada 26 April 2025 pukul 09.00 WIB-11.00 WIB melalui zoom meeting.

### 1. Pembukaan Bimbingan Teknis

Pembukaan bimbingan teknis Uji Kompetensi Sertifikasi BNSP Batch 4 terdapat pada gambar 2.1 Pembukaan Bimbingan Teknis.



**Gambar 2. 2 Pembukaan Bimbingan Teknis**

### 2. Penjelasan Uji Kompetensi

Penjelasan uji kompetensi sertifikasi BNSP oleh Jobhun terdapat pada gambar 2.2 Penjelasan Uji Kompetensi.



**Gambar 2. 3 Penjelasan Uji Kompetensi**

3. Tahapan Uji Kompetensi

Tahapan uji kompetensi meliputi briefing teknis dan persiapan ujikom, persiapan kelengkapan berkas, serta interview bersama asesor. Untuk lebih jelasnya terdapat pada gambar 2.3 Tahapan Uji Kompetensi.



**Gambar 2. 4 Tahapan Uji Kompetensi**

4. Pendukung Uji Kompetensi

Dalam bimbingan teknis juga disebutkan hal-hal yang harus

diperhatikan dalam melaksanakan Uji Kompetensi yang dapat dilihat pada Gambar 2.4 Hal-hal Pendukung Uji Kompetensi.

### Hal-hal yang Perlu Diperhatikan



1. Briefing Teknis persiapan Uji Kompetensi dilaksanakan secara daring pada hari **Jumat, 26 April 2025 pukul 09.00–11.00 WIB**
2. Peserta akan dibagi ke dalam **4 breakout room** untuk dijelaskan mengenai Unit Kompetensi, pengerjaan tugas, penyusunan portofolio, dan persiapan *interview*.
3. Dalam masing-masing *breakout room*, peserta ditemani oleh **1 orang mentor** yang akan menjelaskan mengenai teknis pengerjaan tugas dan teknis pra asesmen
4. Peserta diwajibkan untuk mengunggah seluruh berkas dan menyelesaikan tugas pada link yang sudah disediakan maksimal pada hari **Selasa, 29 April 2025 pukul 23.59 WIB**
5. Uji Kompetensi dan *interview* akan dilaksanakan pada hari **Rabu, 30 April 2025 pukul 09.00–selesai WIB**
6. Seluruh peserta diwajibkan untuk hadir tepat waktu dan bergabung pada Zoom **maksimal 15 menit sebelum *interview* berlangsung**



**Gambar 2. 5 Hal-hal Pendukung Uji Kompetensi**

7. Durasi *interview* masing-masing peserta estimasi **selama 5–15 menit**.
8. Peserta wajib menyiapkan **Laptop, jaringan Wifi yang stabil, dan area yang kondusif** saat ujikom berlangsung
9. Peserta wajib menggunakan **pakaian rapi, sopan, dan formal (Berkerah)** saat *interview* bersama Asesor berlangsung
10. Saat proses mengerjakan tugas wajib menyertakan **bukti dokumentasi** (Contohnya: Sedang mengerjakan tugas skema Digital Marketing di depan laptop bisa minta tolong pada orang lain untuk dokumentasi dari depan atau belakang.
11. Dipastikan bahwa peserta **minimal lulusan SMA/K sederajat**
12. Hasil ujikom ditentukan oleh Asesor dari LSP yang menguji. Sertifikat akan diterbitkan kurang lebih **40–60 hari kerja** setelah berkas diterima oleh BNSP dalam bentuk *hard file*. Sertifikat akan dikirimkan ke alamat masing-masing peserta, dan **ongkos kirim + packing menjadi tanggungan peserta**.



**Gambar 2. 6 Lanjutan Hal-hal Pendukung Uji Kompetensi**

#### 5. Estimasi Perilisan Sertifikat

Setelah semua peserta dapat menyelesaikan semua tugas proyek dan ujian, maka peserta akan mendapatkan sertifikat berbasis BNSP. Setelah Uji Kompetensi terlaksana dan data sudah disetorkan ke BNSP, maka estimasi perilisan sertifikat dapat dilihat pada Gambar 2.6 Estimasi Pembuatan Sertifikat BNSP.

**Estimasi Perilisan Sertifikat**

Setelah Uji Kompetensi Sertifikasi terlaksana dan data sudah disetorkan ke BNSP selama 14 hari kerja, peserta dapat menunggu sesuai *timeline* berikut:

Durasi	Proses	Estimasi tanggal
40 - 60 hari kerja	Proses sertifikat fisik dan digital dari BNSP	Estimasi tanggal (Minimal 40 hari kerja): <b>Senin, 7 Juli 2025</b> Estimasi tanggal (Maksimal 60 hari kerja): <b>Senin, 4 Agustus 2025</b>
14 hari kerja	Proses Surat Keterangan Kompeten Sementara	Estimasi tanggal (Maksimal 14 hari kerja): <b>Jumat, 23 Mei 2025</b>

**Surat Keterangan Kompeten Sementara**

**Sertifikat BNSP**

**Gambar 2. 7 Estimasi Pembuatan Sertifikasi BNSP**

## 2.6 Hand on Lab

Dalam Uji Kompetensi ini menekankan pembelajaran berbasis praktik langsung:

- a. Menulis kode program dari nol
- b. Mengkonfigurasi dan mengelola tools pengembangan
- c. Menerapkan struktur data dan algoritma
- d. Melakukan proses debugging
- e. Melaksanakan pengujian
- f. Menggunakan dan mengintegrasikan library atau framework

## 2.7 Bukti Pendukung Ujian Kompetensi

Dalam pengembangan sebuah web admin toko elektronik, setiap tahap memiliki peran yang krusial untuk menghasilkan aplikasi yang fungsional, efisien dan bebas bug. Berikut adalah uraian lebih lanjut:

1. Penginstalan Composer dapat dilihat pada Gambar 2.7 Hasil

## Instalasi Composer.

```
Terminal Local x Local (2) x Local (3) x + v
==> Installing composer
==> Pouring composer--2.8.10.arm64_sequoia.bottle.tar.gz
📦 /opt/homebrew/Cellar/composer/2.8.10: 4 files, 3MB
==> Running `brew cleanup composer`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
==> No outdated dependents to upgrade!
arraya@MacBook-Pro-Muhammad ~ % composer --version
Composer version 2.8.10 2025-07-10 19:08:33
PHP version 8.4.10 (/opt/homebrew/Cellar/php/8.4.10/bin/php)
Run the "diagnose" command to get more detailed diagnostics output.
arraya@MacBook-Pro-Muhammad ~ %
```

**Gambar 2. 8 Hasil Instalasi Composer**

2. Penginstalan Laravel dan Akses ke server lokal dapat dilihat pada Gambar 2.8 Hasil Instalasi Laravel dan Gambar 2.9 Akses Server Lokal.

```
Terminal Local x Local (2) x Local (3) x + v
[INFO] Discovering packages.

laravel/pail ..... DONE
laravel/sail ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE

79 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

[INFO] No publishable resources for tag [laravel-assets].

No security vulnerability advisories found.
> @php artisan key:generate --ansi

[INFO] Application key set successfully.

> @php -r "file_exists('database/database.sqlite') || touch('database/database.sqlite');"
> @php artisan migrate --graceful --ansi

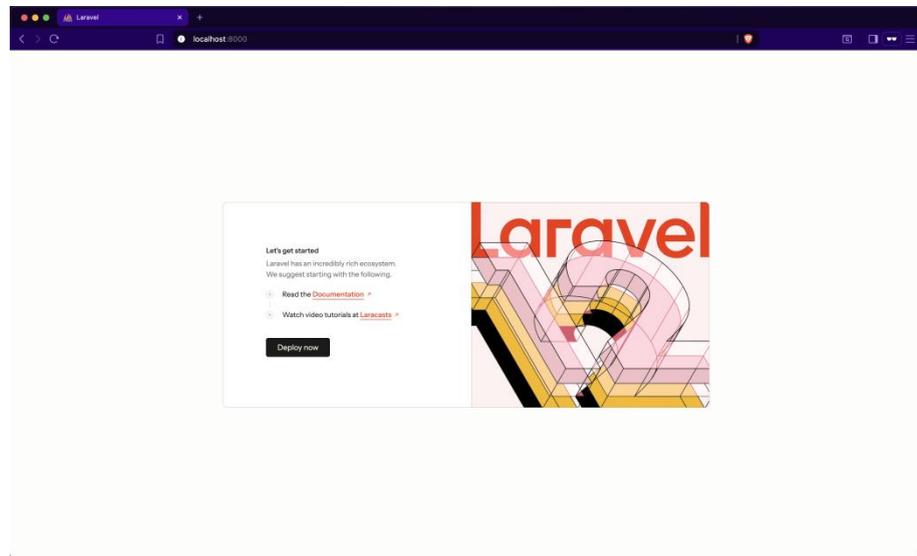
[INFO] Preparing database.

Creating migration table ..... 1.79ms DONE

[INFO] Running migrations.

0001_01_01_000000_create_users_table ..... 2.83ms DONE
0001_01_01_000001_create_cache_table ..... 1.06ms DONE
0001_01_01_000002_create_jobs_table ..... 2.24ms DONE
```

**Gambar 2. 9 Hasil Instalasi Laravel**



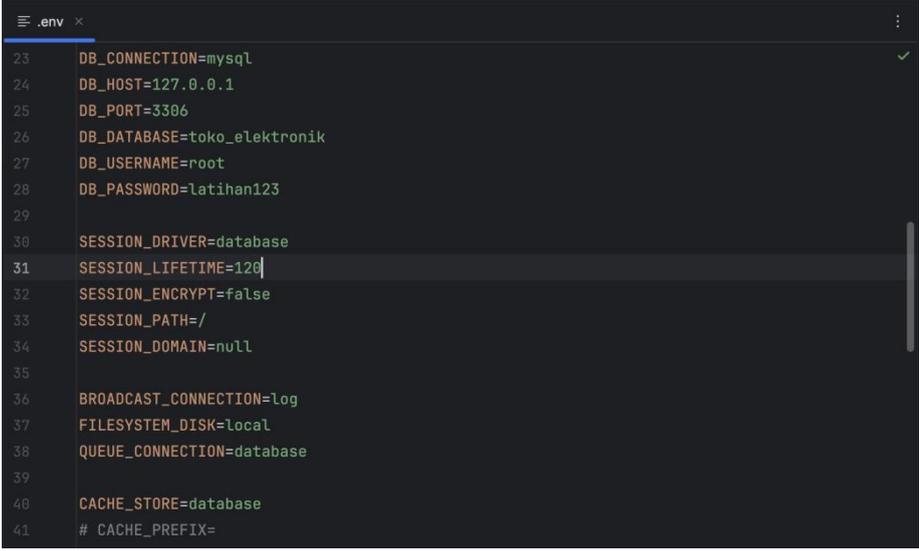
**Gambar 2. 10 Akses Server Lokal**

3. Pembuatan Database dan Konfigurasi untuk file .env dapat dilihat pada Gambar 2.10 Pembuatan Database dan Gambar 2.11 Konfigurasi File .env.

```

data.sql x
1  -- Membuat database
2  CREATE DATABASE IF NOT EXISTS toko_elektronik;
3  USE toko_elektronik;
4
5  -- Tabel untuk menyimpan data produk
6  CREATE TABLE produk (
7      id INT AUTO_INCREMENT PRIMARY KEY,
8      thumbnail VARCHAR(255) NOT NULL,
9      nama_produk VARCHAR(255) NOT NULL,
10     kategori_id INT NOT NULL,
11     harga DECIMAL(15,2) NOT NULL,
12     stok INT NOT NULL DEFAULT 0
13 );
14
15 -- Tabel untuk menyimpan kategori produk
16 CREATE TABLE kategori (
17     id INT AUTO_INCREMENT PRIMARY KEY,
18     nama VARCHAR(100) NOT NULL
19 );
20
21 -- Tabel untuk menyimpan penjualan
22 CREATE TABLE penjualan (
23     id INT AUTO_INCREMENT PRIMARY KEY,
24     tanggal DATE NOT NULL,
  
```

**Gambar 2. 11 Pembuatan Database**



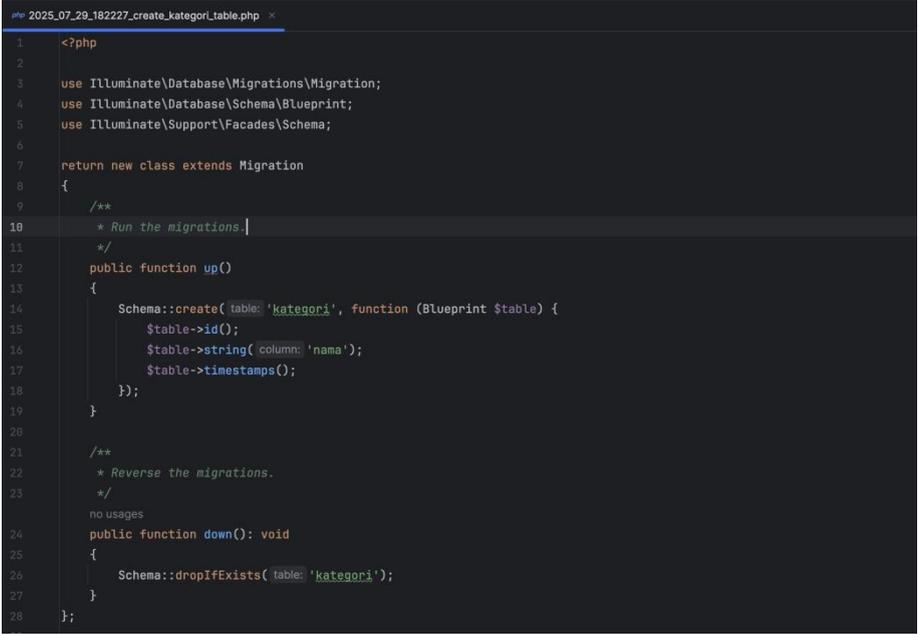
```

23 DB_CONNECTION=mysql
24 DB_HOST=127.0.0.1
25 DB_PORT=3306
26 DB_DATABASE=toko_elektronik
27 DB_USERNAME=root
28 DB_PASSWORD=latihan123
29
30 SESSION_DRIVER=database
31 SESSION_LIFETIME=120
32 SESSION_ENCRYPT=false
33 SESSION_PATH=/
34 SESSION_DOMAIN=null
35
36 BROADCAST_CONNECTION=log
37 FILESYSTEM_DISK=local
38 QUEUE_CONNECTION=database
39
40 CACHE_STORE=database
41 # CACHE_PREFIX=

```

**Gambar 2. 12 Konfigurasi File .env**

4. Pembuatan Migrasi Tabel Kategori dapat dilihat pada Gambar 2.12 Hasil Membuat Tabel Kategori.



```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up()
13     {
14         Schema::create('kategori', function (Blueprint $table) {
15             $table->id();
16             $table->string('nama');
17             $table->timestamps();
18         });
19     }
20
21     /**
22      * Reverse the migrations.
23      */
24     no usages
25     public function down(): void
26     {
27         Schema::dropIfExists('kategori');
28     }
29 };

```

**Gambar 2. 13 Hasil Membuat Tabel Kategori**

5. Pembuatan Migrasi Tabel Produk dapat dilihat pada Gambar 2.13 Hasil Membuat Tabel Produk.

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up()
13     {
14         Schema::create(table: 'produk', function (Blueprint $table) {
15             $table->id();
16             $table->string(column: 'thumbnail');
17             $table->string(column: 'nama_produk');
18             $table->foreignId(column: 'kategori_id')->constrained(table: 'kategori');
19             $table->decimal(column: 'harga', total: 15, places: 2);
20             $table->integer(column: 'stok')->default(value: 0);
21             $table->timestamps();
22         });
23     }
24
25 };

```

**Gambar 2. 14 Hasil Membuat Tabel Produk**

6. Pembuatan Migrasi Tabel Penjualan dapat dilihat pada Gambar 2.14 Hasil Membuat Tabel Penjualan.

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up()
13     {
14         Schema::create(table: 'penjualan', function (Blueprint $table) {
15             $table->id();
16             $table->date(column: 'tanggal');
17             $table->decimal(column: 'total_penjualan', total: 15, places: 2);
18             $table->integer(column: 'jumlah_item');
19             $table->timestamps();
20         });
21     }
22
23     /**
24      * Reverse the migrations.
25      */
26     no usages
27     public function down(): void
28     {
29         Schema::dropIfExists(table: 'penjualan');
30     }
31 };

```

**Gambar 2. 15 Hasil Membuat Tabel Penjualan**

7. Pembuatan Migrasi Tabel Penjualan Kategori dapat dilihat pada Gambar 2.15 Hasil Membuat Tabel Penjualan Kategori.

```

1 <?php
2
3 > use ...
4
5
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up()
13     {
14         Schema::create( table: 'penjualan_kategori', function (Blueprint $table) {
15             $table->id();
16             $table->foreignId( column: 'kategori_id' )->constrained( table: 'kategori' );
17             $table->integer( column: 'bulan' );
18             $table->integer( column: 'tahun' );
19             $table->decimal( column: 'total_penjualan', total: 15, places: 2 );
20             $table->integer( column: 'jumlah_item' );
21             $table->timestamps();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      */
28     no usages
29     public function down(): void
30     {
31         Schema::dropIfExists( table: 'penjualan_kategori' );
32     }
33 };

```

**Gambar 2. 16 Hasil Membuat Tabel Penjualan Kategori**

## 8. Migrasi

Setelah pembuatan Tabel berhasil, Migrasi adalah cara yang efisien untuk membuat dan memodifikasi tabel database menggunakan kode PHP. Hasil Migrasi dapat dilihat pada Gambar 2.16 Hasil Migrasi Tabel.

```

Terminal Local +
arraya@MacBook-Pro-Muhammad toko-elektronik % php artisan migrate

[INFO] Preparing database.

Creating migration table ..... 42.51ms DONE

[INFO] Running migrations.

0001_01_01_000000_create_users_table ..... 37.63ms DONE
0001_01_01_000001_create_cache_table ..... 8.01ms DONE
0001_01_01_000002_create_jobs_table ..... 17.09ms DONE
2025_07_29_180626_create_produk_table ..... 4.18ms DONE
2025_07_29_180644_create_kategoris_table ..... 3.30ms DONE
2025_07_29_180655_create_penjualans_table ..... 5.56ms DONE
2025_07_29_180705_create_penjualan_kategoris_table ..... 1.91ms DONE

arraya@MacBook-Pro-Muhammad toko-elektronik %

```

**Gambar 2. 17 Hasil Migrasi Tabel**

## 9. Pembuatan model kategori dapat dilihat pada Gambar 2.17 Model Kategori.

```
Kategori.php x
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Kategori extends Model
8 {
9     protected $table = 'kategori';
10     no usages
11     protected $fillable = ['nama'];
12
13     public function produk()
14     {
15         return $this->hasMany(related: Produk::class);
16     }
17 }
```

**Gambar 2. 18 Model Kategori**

10. Pembuatan model penjualan dapat dilihat pada Gambar 2.18 Model Penjualan.

```
Penjualan.php x
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Penjualan extends Model
8 {
9     protected $table = 'penjualan';
10     no usages
11     protected $fillable = ['tanggal', 'total_penjualan', 'jumlah_item'];
12 }
```

**Gambar 2. 19 Model Penjualan**

11. Pembuatan model penjualan kategori dapat dilihat pada Gambar 2.19 Model Penjualan Kategori.

```
PenjualanKategori.php x
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class PenjualanKategori extends Model
8 {
9     protected $table = 'penjualan_kategori';
10    protected $fillable = ['kategori_id', 'bulan', 'tahun', 'total_penjualan', 'jumlah_item'];
11
12    public function kategori()
13    {
14        return $this->belongsTo(related: Kategori::class);
15    }
16 }
17
```

**Gambar 2. 20 Model Penjualan Kategori**

12. Pembuatan model produk dapat dilihat pada Gambar 2.20 Model Produk.

```
Produk.php x
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Produk extends Model
8 {
9     protected $table = 'produk';
10    public $timestamps = false;
11    protected $fillable = ['thumbnail', 'nama_produk', 'kategori_id', 'harga', 'stok'];
12
13    public function kategori()
14    {
15        return $this->belongsTo(related: Kategori::class);
16    }
17 }
18
```

**Gambar 2. 21 Model Produk**

13. Pembuatan model User dapat dilihat pada Gambar 2.21 Model User.

```
19 class User extends Authenticatable
20     protected $fillable = [
21         'name',
22         'email',
23         'password',
24     ];
25
26     /**
27      * The attributes that should be hidden for serialization.
28      *
29      * @var list<string>
30      */
31     no usages
32     protected $hidden = [
33         'password',
34         'remember_token',
35     ];
36
37     /**
38      * Get the attributes that should be cast.
39      *
40      * @return array<string, string>
41      */
42     no usages
43     protected function casts(): array
44     {
45         return [
46             'email_verified_at' => 'datetime',
47             'password' => 'hashed',
48         ];
49     }
50 }
```

**Gambar 2. 22 Model User**

14. Pembuatan Dashboard Controller dapat dilihat pada Gambar 2.22 Dashboard Controller.

```

10 class DashboardController extends Controller
11 {
12     public function index()
13     {
14         $total_produk = Produk::count();
15         $total_harga = Produk::sum('harga');
16
17         // Data untuk line chart
18         $penjualan = Penjualan::orderBy('tanggal', 'ASC')
19             ->limit(6)
20             ->get();
21
22         $labels_line = $penjualan->map(function($item) {
23             return date(format: 'M Y', strtotime($item->tanggal));
24         });
25         $data_line = $penjualan->pluck('total_penjualan');
26
27         // Data untuk pie chart
28         $penjualan_kategori = PenjualanKategori::with('kategori')
29             ->where('bulan', date(format: 'n'))
30             ->where('tahun', date(format: 'Y'))
31             ->get();
32
33         $labels_pie = $penjualan_kategori->map(function($item) {
34             return $item->kategori->nama;
35         });
36         $data_pie = $penjualan_kategori->pluck('value: total_penjualan');
37         $colors_pie = ['#FF6384', '#36A2EB', '#FFCE56', '#4BC0C0', '#9966FF', '#FF9F40'];
38
39         return view('dashboard.index', compact(
40             'total_produk',
41             'total_harga',
42             'labels_line',
43             'data_line',

```

**Gambar 2. 23 Dashboard Controller**

15. Pembuatan Produk Controller dapat dilihat pada Gambar 2.23 Produk Controller.

```

9 class ProdukController extends Controller
10 {
11     public function index()
12     {
13         $produk = Produk::with('kategori')->orderBy('id', 'DESC')->get();
14         return view('produk.index', compact('produk'));
15     }
16
17     public function create()
18     {
19         $kategori = Kategori::all();
20         return view('produk.create', compact('kategori'));
21     }
22
23     no usages
24     public function store(Request $request)
25     {
26         $validated = $request->validate([
27             'thumbnail' => 'required|url',
28             'kategori_id' => 'required|exists:kategori,id',
29             'nama_produk' => 'required',
30             'harga' => 'required|numeric',
31             'stok' => 'required|numeric'
32         ]);
33         Produk::create($validated);
34         return redirect()->route('produk.index')->with('success', 'Produk berhasil ditambahkan');
35     }
36
37     no usages
38     public function edit(Produk $produk)
39     {
40         $kategori = Kategori::all();
41         return view('produk.edit', compact('produk', 'kategori'));
42     }
43 }

```

**Gambar 2. 24 Produk Controller**

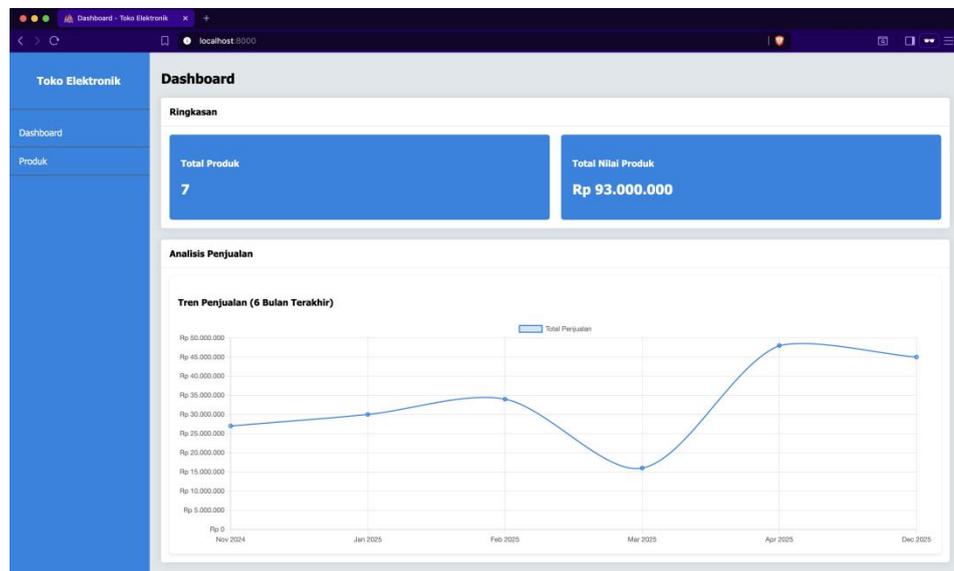
16. Pembuatan halaman dashboard dapat dilihat pada Gambar 2.24 Pembuatan Halaman Dashboard dan tampilan halaman dashboard ketika diakses dapat dilihat pada Gambar 2.25 Tampilan Halaman Dashboard.

```

1 @extends('layouts.app')
2
3 @section('title', 'Dashboard')
4
5 @section('content')
6 <h2>Dashboard</h2>
7
8 <div class="card">
9     <div class="card-header">
10         Ringkasan
11     </div>
12     <div class="card-body">
13         <div style="display: flex; gap: 20px;">
14             <div style="flex: 1; background-color: #998463; color: white; padding: 20px; border-radius: 5px;">
15                 <h4>Total Produk</h4>
16                 <h2>{{ $total_produk }}</h2>
17             </div>
18             <div style="flex: 1; background-color: #998463; color: white; padding: 20px; border-radius: 5px;">
19                 <h4>Total Nilai Produk</h4>
20                 <h2>Rp {{ number_format($total_harga, 0, ',', '.') }}</h2>
21             </div>
22         </div>
23     </div>
24 </div>
25
26 <div class="card">
27     <div class="card-header">
28         Analisis Penjualan
29     </div>
30     <div class="card-body">
31         <div class="chart-card">
32             <h4>Tren Penjualan (6 Bulan Terakhir)</h4>
33             <div class="chart-container" style="height: 400px; width: 100%;">
34
35         </div>
36     </div>
37 </div>

```

**Gambar 2. 25 Pembuatan Halaman Dashboard**



**Gambar 2. 26 Tampilan Halaman Dashboard**

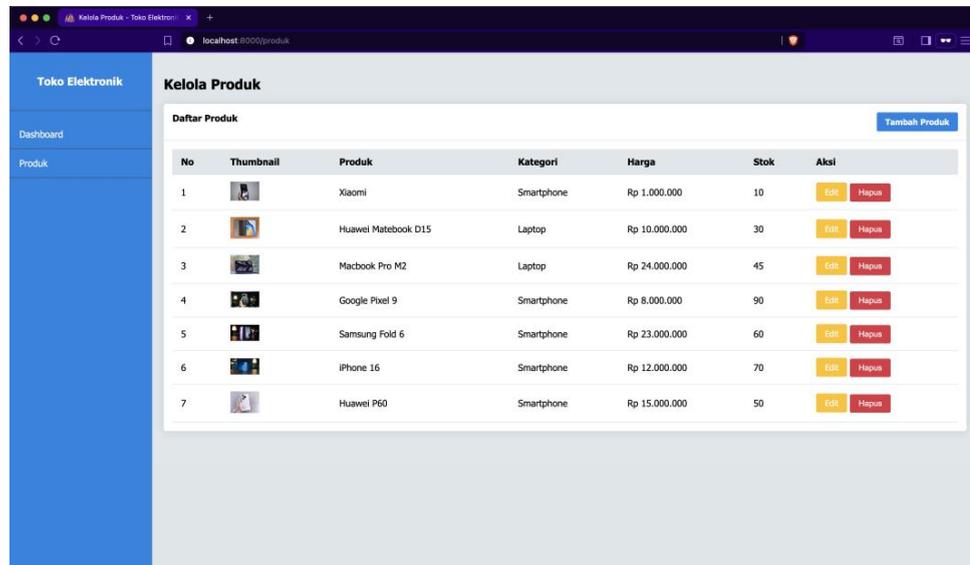
17. Pembuatan halaman produk dapat dilihat pada Gambar 2.26  
 Pembuatan Halaman Produk dan tampilan halaman produk dapat dilihat  
 pada Gambar 2.27 Tampilan Halaman produk.

```

index.blade.php
5 @section('content')
6 <h2>Kelola Produk</h2>
7
8 <div class="card">
9 <div class="card-header">
10 <span>Daftar Produk</span>
11 <a href="{{ route('produk.create') }}" class="btn btn-primary">Tambah Produk</a>
12 </div>
13 <div class="card-body">
14 @if(session('success'))
15 <div style="background-color: #28a745; color: white; padding: 5px;>
16 {{ session('success') }}
17 </div>
18 @endif
19
20 <table>
21 <thead>
22 <tr>
23 <th>No</th>
24 <th>Thumbnail</th>
25 <th>Produk</th>
26 <th>Kategori</th>
27 <th>Harga</th>
28 <th>Stok</th>
29 <th>Aksi</th>
30 </tr>
31 </thead>
32 <tbody>
33 @foreach($produk as $index => $item)
34 <tr>
35 <td>{{ $index + 1 }}</td>
36 <td></td>
37 <td>{{ $item->nama_produk }}</td>
38 <td>{{ $item->kategori->nama }}</td>

```

**Gambar 2. 27 Pembuatan Halaman Produk**



No	Thumbnail	Produk	Kategori	Harga	Stok	Aksi
1		Xiaomi	Smartphone	Rp 1.000.000	10	<a href="#">Edit</a> <a href="#">Hapus</a>
2		Huawei Matebook D15	Laptop	Rp 10.000.000	30	<a href="#">Edit</a> <a href="#">Hapus</a>
3		Macbook Pro M2	Laptop	Rp 24.000.000	45	<a href="#">Edit</a> <a href="#">Hapus</a>
4		Google Pixel 9	Smartphone	Rp 8.000.000	90	<a href="#">Edit</a> <a href="#">Hapus</a>
5		Samsung Fold 6	Smartphone	Rp 23.000.000	60	<a href="#">Edit</a> <a href="#">Hapus</a>
6		iPhone 16	Smartphone	Rp 12.000.000	70	<a href="#">Edit</a> <a href="#">Hapus</a>
7		Huawei P60	Smartphone	Rp 15.000.000	50	<a href="#">Edit</a> <a href="#">Hapus</a>

**Gambar 2. 28 Tampilan Halaman Produk**

18. Pembuatan fitur tambah produk dapat dilihat pada Gambar 2.28. Pembuatan Fitur Tambah Produk dan tampilan halaman tambah produk dapat dilihat pada Gambar 2.29 Tampilan Halaman Tambah Produk.

```

1  @section('title', 'Tambah Produk')
2
3  @section('content')
4
5  <h2>Tambah Produk</h2>
6
7
8  <div class="card">
9    <div class="card-header">
10     Form Tambah Produk
11    </div>
12    <div class="card-body">
13      @if($errors->any())
14        <div style="background-color: #f8d7da; padding: 5px;>
15          <ul style="list-style-type: none; padding: 0; margin: 0;>
16            @foreach($errors->all() as $error)
17              <li>{{ $error }}</li>
18            @endforeach
19          </ul>
20        </div>
21      @endif
22
23      <form action="{{ route('produk.store') }}" method="POST">
24        @csrf
25        <div class="form-group">
26          <label for="thumbnail">Link Thumbnail</label>
27          <input type="text" class="form-control" id="thumbnail" name="thumbnail" value="{{ old('thumbnail') }}" required>
28        </div>
29        <div class="form-group">
30          <label for="kategori_id">Kategori</label>
31          <select class="form-control" id="kategori_id" name="kategori_id" required>
32            <option value="">Pilih Kategori</option>
33            @foreach($kategori as $kat)
34              <option value="{{ $kat->id }}" {{ old('kategori_id') == $kat->id ? 'selected' : '' }}>
35                {{ $kat->nama }}
36            </option>

```

**Gambar 2. 29 Pembuatan Fitur Tambah Produk**

**Gambar 2. 30 Tampilan Halaman Tambah Produk**

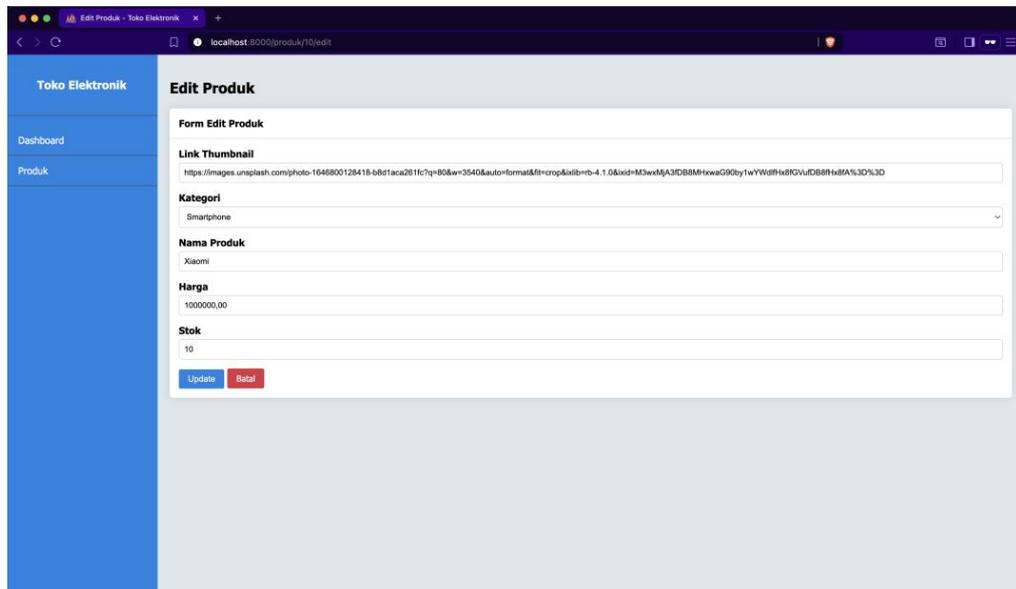
19. Pembuatan fitur edit pada produk dapat dilihat pada Gambar 2.30 Pembuatan Fitur Edit Produk dan tampilan halaman edit dapat dilihat pada Gambar 2.31 Tampilan Halaman Edit Produk.

```

edit.blade.php
3  @section('title', 'Edit Produk')
4
5  @section('content')
6  <h2>Edit Produk</h2>
7
8  <div class="card">
9    <div class="card-header">
10     Form Edit Produk
11    </div>
12    <div class="card-body">
13      @if($errors->any())
14        <div style="background-color: #f8d7da; padding: 5px;>
15          <ul style="list-style-type: none; padding-left: 0;>
16            @foreach($errors->all() as $error)
17              <li>{{ $error }}</li>
18            @endforeach
19          </ul>
20        </div>
21      @endif
22
23      <form action="{{ route('produk.update', $produk) }}" method="POST">
24        @csrf
25        @method('PUT')
26        <div class="form-group">
27          <label for="thumbnail">Link Thumbnail</label>
28          <input type="text" class="form-control" id="thumbnail" name="thumbnail"
29            value="{{ old('thumbnail', $produk->thumbnail) }}" required>
30        </div>
31        <div class="form-group">
32          <label for="kategori_id">Kategori</label>
33          <select class="form-control" id="kategori_id" name="kategori_id" required>
34            <option value="">Pilih Kategori</option>
35            @foreach($kategori as $kat)
36              <option value="{{ $kat->id }}">

```

**Gambar 2. 31 Pembuatan Fitur Edit Produk**



**Gambar 2. 32 Tampilan Halaman Edit Produk**

## 2.8 Sesi Interview

Pada tahap ini, dilaksanakan sesi akhir dari uji kompetensi bersama jobhun pada tanggal 30 April 2025, yang dipimpin langsung oleh Bapak Oki Irawan selaku asesor skema Web Development.

Adapun beberapa pertanyaan yang diajukan dalam sesi wawancara beserta jawabannya sebagai berikut:

1. bahasa pemrograman apa yang dipakai untuk membuat aplikasi ini?  
Jawaban: Bahasa yang digunakan adalah PHP dan CSS.
2. berapa lama proses pembuatannya?  
Jawaban: Proses pengerjaan memerlukan waktu selama 3 hari.
3. kendala apa saja yang ditemui saat mengerjakan tugas?  
Jawaban: Kendala yang dihadapi adalah pembuatan grafik penjualan yang menggunakan Javascript.
4. apakah kamu mengetahui mengenai web server?  
Jawaban: web server yang saya tau adalah apache.

5. apakah kamu bisa mengembangkan tugas ini menggunakan laravel?

Jawaban: Ke depannya, saya berencana mengembangkan aplikasi ini dengan menggunakan framework Laravel.

6. apa tujuan kamu mengikuti uji kompetensi dari Jobhun.id?

Jawaban: Tujuan saya adalah untuk meningkatkan keahlian dalam bidang Web Development serta memperoleh sertifikat uji kompetensi berlisensi BNSP yang dapat digunakan untuk memenuhi syarat tugas akhir.

Proses wawancara ini terdokumentasi dalam gambar 2.33 Sesi Interview sebagai bukti pelaksanaan tahap akhir uji kompetensi.



**Gambar 2. 33 Sesi Interview**