

DOKUMENTASI

APLIKASI TES PROGRAMMER FAST PRINT

1. Tentang Aplikasi

Aplikasi ini dibuat untuk Seleksi Tahap Pertama Fast Print Indonesia cabang Surabaya untuk posisi Junior Programmer. Aplikasi ini dibangun menggunakan framework CodeIgniter4 dengan bahasa pemrograman PHP.

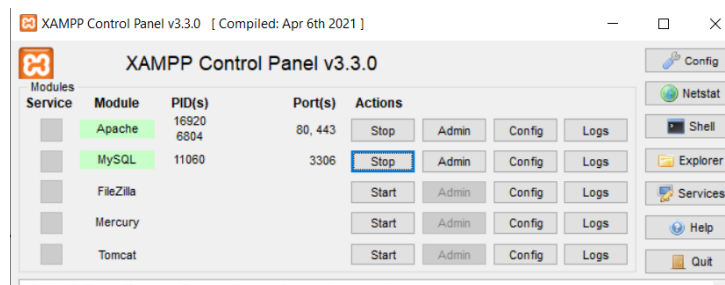
2. Instalasi Aplikasi

Untuk melakukan penginstalan aplikasi dibutuhkan beberapa alat berikut:

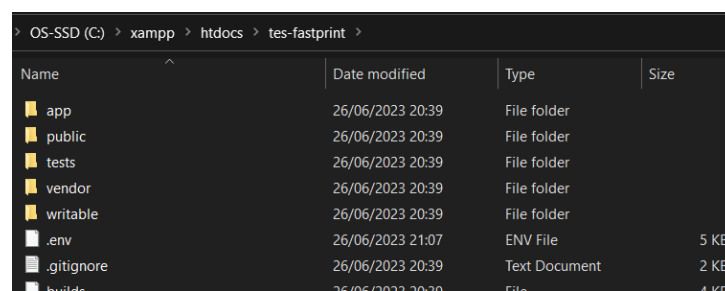
- 1) XAMPP Server atau yang sejenisnya
- 2) PHP 7.4+
- 3) MySQL 5.6 atau MariaDB 10
- 4) Web Browser

Proses Instalasi:

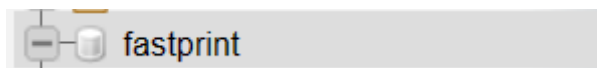
- 1) Jalankan XAMPP Server dan aktifkan module apache dan mysql



- 2) Pindahkan folder project ke dalam folder C:\xampp\htdocs



- 3) Buat database baru pada MySQL dengan nama fastprint melalui phpMyAdmin



- 4) Import file sql yang ada didalam folder database ke mysql server melalui phpMyAdmin

Importing into the database "fastprint"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.

A compressed file's name must end in **[format].[compression]**. Example: **.sql.zip**

Browse your computer: fastprint.sql (Max: 40MiB)

You may also drag and drop a file on any page.

Character set of the file:

- 5) Ubah nama file env didalam folder project menjadi .env dan ubah identitas database didalam file .env

```
database.default.hostname = localhost
database.default.database = fastprint
database.default.username = root
database.default.password =
database.default.DBDriver = MySQLi
database.default.DBPrefix =
database.default.port = 3306
```

- 6) Cmd ke folder project lalu lakukan composer install

```
C:\Windows\System32\cmd.exe
^C
C:\xampp\htdocs\tes-fastprint>composer install
```

- 7) Setelah itu jalankan aplikasi melalui cmd dengan melakukan cd kedalam direktori project, lalu tulis command “php spark serve”, jika berhasil maka aplikasi sudah berjalan

```
C:\Windows\System32\cmd.exe - php spark serve
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

C:\xampp\htdocs\tes-fastprint>php spark serve

CodeIgniter v4.3.6 Command Line Tool - Server Time: 2023-06-28 09:57:03 UTC+00:00

CodeIgniter development server started on http://localhost:8080
Press Control-C to stop.
[Wed Jun 28 16:57:03 2023] PHP 7.4.29 Development Server (http://localhost:8080) started
```

3. Halaman Utama

Untuk mengakses halaman utama, buka web browser lalu ketik alamat <http://localhost:8080/> maka akan muncul tampilan awal aplikasi

Data Produk

[Tambah Produk](#)

Id	Nama Produk	Harga	Kategori	Status	Aksi
6	ALCOHOL GEL POLISH CLEANSER GP-CLN01	12500	L QUEENLY	bisa dijual	Ubah Hapus
9	ALUMINIUM FOIL ALL IN ONE BULAT 23mm IM	1000	L MTH AKSESORIS (IM)	bisa dijual	Ubah Hapus
11	ALUMINIUM FOIL ALL IN ONE BULAT 30mm IM	1000	L MTH AKSESORIS (IM)	bisa dijual	Ubah Hapus
15	ALUMINIUM FOIL HDPE/PE BULAT 23mm IM	12500	L MTH AKSESORIS (IM)	bisa dijual	Ubah Hapus
17	ALUMINIUM FOIL HDPE/PE BULAT 30mm IM	1000	L MTH AKSESORIS (IM)	bisa dijual	Ubah Hapus
22	ARM PENDEK MODEL U	13000	L MTH AKSESORIS (IM)	bisa dijual	Ubah Hapus
26	ARM SUPPORT PENDEK POLOS	13000	L MTH TABUNG (LK)	bisa dijual	Ubah Hapus
28	ARM SUPPORT T (IMPORT)	13000	L MTH AKSESORIS (IM)	bisa dijual	Ubah Hapus
29	ARM SUPPORT T - MODEL 1 (LOKAL)	10000	L MTH TABUNG (LK)	bisa dijual	Ubah Hapus

Kode Pemrograman untuk menampilkan data produk sebagai berikut:

```
public function index()
{
    $produk = new Produk();

    $dataProduk = $produk->where('status', 'bisa dijual')->findAll();

    $data = [
        'produk' => $dataProduk
    ];

    return view('index', $data);
}
```

Kode diatas berada pada bagian controller, pada kode diatas hal pertama yang dilakukan yaitu memanggil model Produk(), lalu mengambil data produk dimana status = bisa dijual lalu disimpan pada variabel \$dataproduk yang kemudian dikirimkan ke view untuk ditampilkan.

4. Tambah Produk

Proses tambah produk dilakukan dengan cara berikut:

- 1) Menekan tombol tambah produk

Data Pro

[Tambah Produk](#)

Id	Nama Produk	Harga	
6	ALCOHOL GEL POLISH CLEANSER GP-CLN01	12500	
9	ALUMINIUM FOIL ALL IN ONE BULAT 23mm IM	1000	L M

- 2) Mengisi form dengan benar dan menekan tombol tambah produk

Tambah Produk

Id Produk
73

Nama Produk
Tes Produk

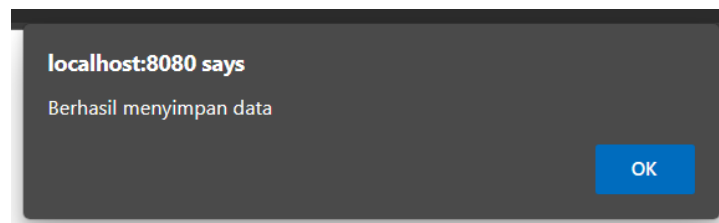
Harga Produk
1000

Kategori Produk
Kategori Tes

Status Produk
Bisa dijual

Tambah Produk

- 3) Jika input berhasil maka akan muncul alert berhasil dan data akan muncul pada halaman utama



70	BOTOL KOTAK 100ML LK	1000	L MTH AKSESORIS (LK)	bisa dijual	Ubah	Hapus
73	Tes Produk	1000	Kategori Tes	bisa dijual	Ubah	Hapus

- 4) Jika terdapat kesalahan data atau ketidaksesuaian data maka akan dikembalikan kehalaman form lagi dan terdapat peringatan kesalahan pada form.

Tambah Produk

Id Produk
73
The id_produk field must contain a unique value.

Nama Produk
The nama_produk field is required.

Harga Produk
The harga field must contain only numbers.

Kategori Produk
Tes Kategori

Status Produk
-- Pilih Status --

Tambah Produk

Kode yang digunakan untuk melakukan proses tambah data sebagai berikut

```
public function tambahProduk()
{
    $data = [
```

```

        'validation' => validation_errors()
    ];

    return view('tambah_produk', $data);
}
public function aksiTambahProduk()
{
    $produk = new Produk();

    if (!$this->validate([
        'id_produk' => [
            'rules' => 'required|is_unique[produk.id_produk]'
        ],
        'nama_produk' => [
            'rules' => 'required'
        ],
        'harga' => [
            'rules' => 'required|numeric'
        ],
        'kategori' => [
            'rules' => 'required'
        ],
        'status' => [
            'rules' => 'required'
        ],
    ])) {
        return redirect()->back()->withInput();
    }

    $data = [
        'id_produk' => $this->request->getPost('id_produk'),
        'nama_produk' => $this->request->getPost('nama_produk'),
        'harga' => $this->request->getPost('harga'),
        'kategori' => $this->request->getPost('kategori'),
        'status' => $this->request->getPost('status')
    ];

    $produk->save($data);

    return redirect()->to(base_url())->with('sukses', 'Berhasil
    menyimpan data');
}

```

Kode diatas berada pada bagian controller, pada kode diatas fungsi tambahProduk() berfungsi untuk menampilkan halaman form, pada fungsi tambahProduk terdapat validation_errors() untuk menangkap pesan error validasi. Fungsi aksiTambahProduk() berfungsi untuk melakukan proses tambah data, pada fungsi tersebut dilakukan proses validasi inputan dengan memanfaatkan fitur validasi bawaan CodeIgniter4 jika data

tidak sesuai maka akan dikembalikan dengan pesan error, jika sesuai data selanjutnya akan dimasukkan kedalam database untuk disimpan dan pengguna diarahkan kembali ke halaman utama.

5. Ubah Produk

Proses ubah produk dilakukan dengan cara berikut

- 1) Menekan tombol ubah pada salah satu data yang akan diubah

70	BOTOL KOTAK 100ML LK	1000	L MTH AKSESORIS (LK)	bisa dijual	Ubah	Hapus
73	Tes Produk	1000	Kategori Tes	bisa dijual	Ubah	Hapus

- 2) Mengisi form dengan data baru yang ingin diubah dengan benar, lalu menekan tombol ubah produk

Ubah Produk

Nama Produk

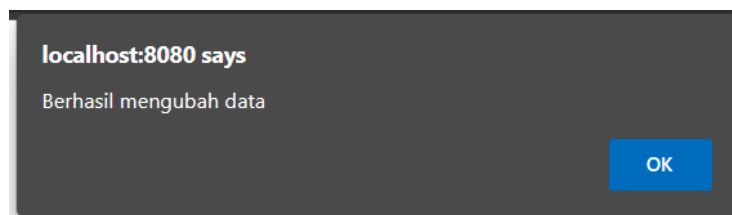
Harga Produk

Kategori Produk

Status Produk

Ubah Produk

- 3) Jika berhasil maka akan muncul alert data berhasil diubah dan data pada halaman utama akan berubah.



73	Tes Produk Diubah	10000	Kategori Tes	bisa dijual	Ubah	Hapus
----	-------------------	-------	--------------	-------------	------	-------

- 4) Jika terdapat kesalahan data atau ketidaksesuaian data maka akan dikembalikan ke halaman form lagi dan terdapat peringatan kesalahan pada form

Ubah Produk

Nama Produk

The nama_produk field is required.

Harga Produk

10000

The harga field is required.

Kategori Produk

Kategori Tes

The kategori field is required.

Kode yang digunakan untuk proses ubah produk sebagai berikut:

```
public function ubahProduk($id_produk)
{
    $produk = new Produk();

    $dataProduk = $produk->find($id_produk);

    $data = [
        'validation' => validation_errors(),
        'produk' => $dataProduk
    ];

    return view('ubah_produk', $data);
}

public function aksiUbahProduk()
{
    $produk = new Produk();

    if (!$this->validate([
        'id_produk' => [
            'rules' => 'required'
        ],
        'nama_produk' => [
            'rules' => 'required'
        ],
        'harga' => [
            'rules' => 'required|numeric'
        ],
        'kategori' => [
            'rules' => 'required'
        ],
        'status' => [
            'rules' => 'required'
        ],
    ])) {
```

```

        return redirect()->back()->withInput();
    }

    $data = [
        'id_produk' => $this->request->getPost('id_produk'),
        'nama_produk' => $this->request->getPost('nama_produk'),
        'harga' => $this->request->getPost('harga'),
        'kategori' => $this->request->getPost('kategori'),
        'status' => $this->request->getPost('status')
    ];

    $produk->save($data);

    return redirect()->to(base_url())->with('sukses', 'Berhasil
mengubah data');
}

```

Kode diatas berada pada bagian controller, pada kode diatas fungsi ubahProduk() berfungsi untuk menampilkan halaman form edit, pada fungsi tersebut terdapat parameter \$id_produk yang digunakan untuk mengidentifikasi data yang ingin diubah oleh pengguna, yang selanjutnya data akan ditampilkan ke pengguna. Fungsi aksiUbahProduk() berfungsi untuk melakukan proses ubah data produk, hampir sama seperti prose tambah produk, data akan divalidasi terlebih dahulu jika terdapat kesalahan maka akan dikembalikan kehalaman form dengan pesan error, lalu jika data sudah sesuai maka dilanjutkan untuk menyimpan perubahan ke database.

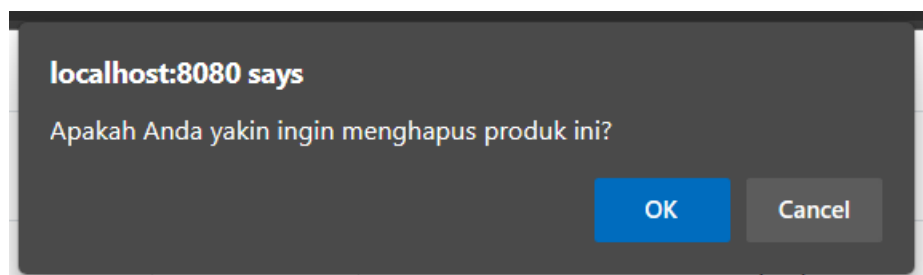
6. Hapus Produk

Proses hapus dilakukan dengan cara berikut:

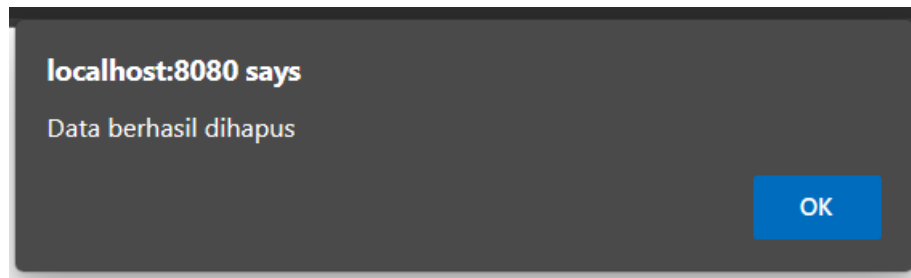
- 1) Menekan tombol hapus pada salah satu data yang ingin dihapus

73	Tes Produk Diubah	10000	Kategori Tes	bisa dijual	Ubah	Hapus
					Ubah	Hapus

- 2) Akan muncul alert konfirmasi untuk melanjutkan aksi atau tidak



- 3) Jika pengguna memilih melanjutkan aksi maka akan muncul alert berhasil dihapus dan data akan dihapus permanen.



70	BOTOL KOTAK 100ML LK	1000	L MTH AKSESORIS (LK)	bisa dijual	Ubah	Hapus
----	----------------------	------	----------------------	-------------	------	-------

Kode yang digunakan untuk proses hapus data sebagai berikut:

```
public function aksiHapusProduk()
{
    $id_produk = $this->request->getPost('id_produk');

    $produk = new Produk();
    $produk->delete($id_produk);

    return redirect()->to(base_url())->with('sukses', 'Data berhasil
dihapus');
}
```

Kode diatas berada pada bagian controller, pada kode diatas terdapat fungsi aksiHapusProduk() untuk melakukan proses hapus data produk. pada fungsi tersebut mengambil parameter id_produk yang digunakan sebagai identifikasi data.