

SISTEM MANAJEMEN PRODUK MENGGUNAKAN FRAMEWORK CODEIGNITER 3

Disusun untuk Memenuhi Tes Junior Programmer

Fast Print Indonesia

Posisi: Programmer



Disusun Oleh:

Nama : Imam Maskuri
Email : imamkuri5@gmail.com
Nomor HP : 0895395170139

Tanggal Pengumpulan:

[13 Januari 2025]

1.1 Deskripsi Website

Aplikasi ini merupakan sistem manajemen produk berbasis web menggunakan framework CodeIgniter 3. Sistem ini memungkinkan pengguna untuk melakukan operasi CRUD (*Create, Read, Update, Delete*) pada data produk, serta dilengkapi fitur filter berdasarkan status produk (contoh: "Bisa Dijual"). Aplikasi ini dirancang untuk memudahkan pengelolaan data produk dengan antarmuka yang sederhana dan *user-friendly*. Adapun tujuan dibuat program ini adalah sebagai berikut.

- Membangun sistem yang dapat mengelola data produk secara efisien.
- mempraktikkan konsep MVC (*Model-View-Controller*) dalam pengembangan web.
- Memberikan pengalaman belajar membangun fitur CRUD pada framework PHP.
- Sebagai tes untuk junior web *programmer*

1.2 Fitur Website

Berikut adalah fitur utama yang ada dalam aplikasi:

1.1 Tambah Produk

- Pengguna dapat menambahkan produk baru dengan mengisi *form* yang mencakup: nama produk, harga, kategori, dan status.
- Validasi data dilakukan sebelum data disimpan ke database.

1.2 Lihat Produk

- Menampilkan daftar produk yang tersedia di database.
- Fitur filter berdasarkan status produk, seperti "Bisa Dijual" atau "Semua Produk".

1.3 Ubah Produk

- Pengguna dapat mengedit informasi produk, termasuk nama, harga, kategori, dan status.

1.4 Hapus Produk

- Pengguna dapat menghapus produk tertentu dengan konfirmasi penghapusan.

1.5 Sinkronisasi Data

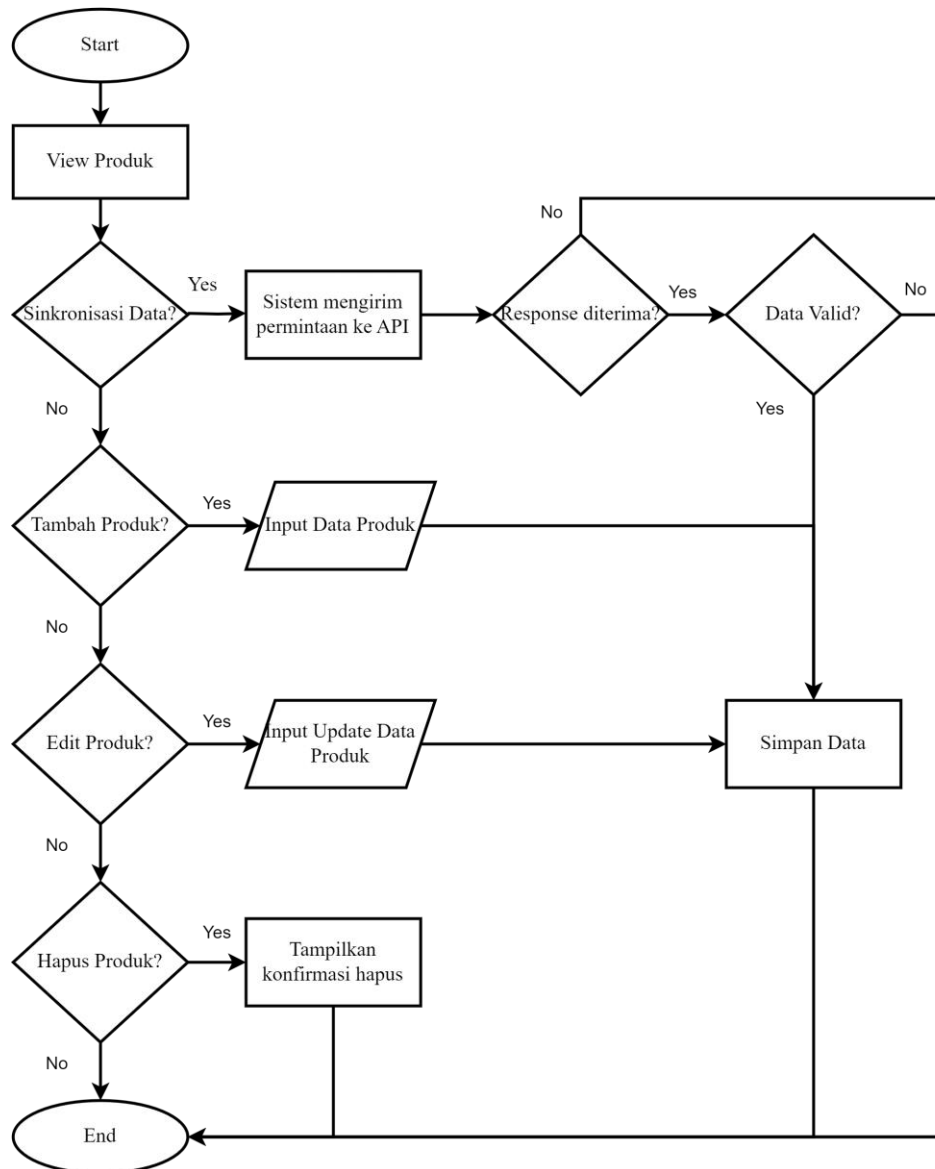
- Fitur untuk menyinkronkan data dari sumber eksternal yaitu mengambil data dari API yang telah disediakan yaitu :

https://recruitment.fastprint.co.id/tes/api_tes_programmer

1.6 Notifikasi

- Pesan notifikasi untuk operasi sukses (seperti "Produk berhasil ditambahkan") atau kesalahan (seperti "Produk tidak ditemukan").

1.3 Arsitektur Sistem



Gambar 1 Arsitektur Sistem

A. Database

Langkah pertama adalah memahami struktur database di buat sebagai berikut.

1. Tabel produk

Ini adalah tabel utama yang akan menyimpan informasi produk, termasuk hubungan ke tabel kategori dan status.

Struktur Tabel:

- `id_produk` (INT, *Primary Key*): ID unik untuk setiap produk.

- nama_produk (VARCHAR, 255): Nama dari produk.
- harga (INT): Harga produk.
- kategori_id (INT, *Foreign Key*): Menghubungkan produk dengan kategori di tabel kategori.
- status_id (INT, *Foreign Key*): Menghubungkan produk dengan status di tabel status.

Relasi:

- kategori_id merujuk ke kolom id_kategori di tabel kategori.
- status_id merujuk ke kolom id_status di tabel status.

```
CREATE TABLE `produk` (
  `id_produk` INT PRIMARY KEY,
  `nama_produk` VARCHAR(255),
  `harga` INT,
  `kategori_id` INT,
  `status_id` INT,
  FOREIGN KEY (`kategori_id`) REFERENCES `kategori`(`id_kategori`),
  FOREIGN KEY (`status_id`) REFERENCES `status`(`id_status`)
);
```

Kode Program 1 SQL Produk

2. Tabel kategori

Tabel ini menyimpan kategori dari produk.

Struktur Tabel:

- id_kategori (INT, *AUTO_INCREMENT*, *Primary Key*): ID unik untuk setiap kategori.
- nama_kategori (VARCHAR, 255, *UNIQUE*): Nama kategori. Harus unik, agar tidak ada duplikasi nama kategori.

```
CREATE TABLE `kategori` (
  `id_kategori` INT AUTO_INCREMENT PRIMARY KEY,
  `nama_kategori` VARCHAR(255) UNIQUE
);
```

Kode Program 2 SQL kategori

3. Tabel status

Tabel ini menyimpan status penjualan produk.

Struktur Tabel:

- id_status (INT, *AUTO_INCREMENT*, *Primary Key*): ID unik untuk setiap status.
- nama_status (VARCHAR, 255, *UNIQUE*): Nama status. Harus unik, agar tidak ada duplikasi nama status.

```
CREATE TABLE `status` (  
  `id_status` INT AUTO_INCREMENT PRIMARY KEY,  
  `nama_status` VARCHAR(255) UNIQUE  
);
```

Kode Program 3 SQL Status

B. Ambil Data API

1. *ControllerProducts : FetchData*

Fungsi:

- Menghapus data yang ada di tabel produk sebelum sinkronisasi.
- Mengambil data dari API menggunakan cURL dengan *username* dan *password* dinamis berdasarkan waktu.
- Menyimpan data yang berhasil diambil ke database menggunakan fungsi di model.

Penjelasan Detail:

a. Waktu Dinamis

- *DateTime* digunakan untuk menangani waktu saat ini.
- Ditambahkan 1 jam ke waktu saat ini untuk menghitung *username* dan *password*.

b. *Username* dan *Password*:

- *Username* diformat berdasarkan tanggal, bulan, tahun, dan jam.
- *Password* diformat sesuai pola dan di-hash menggunakan MD5.

c. Permintaan API dengan cURL:

- Data dikirim dengan *POST*.
- Respons API di-decode menjadi *array* dengan *json_decode*.

d. Penyimpanan Data:

- Jika respons API berhasil (*error == 0*), data disimpan ke database menggunakan fungsi *saveData* di model.
- Jika gagal, pesan *error* ditampilkan.

2. *ModelProducts : SaveData*

Berfungsi untuk memasukkan data API ke dalam tabel produk, kategori, dan status.

Penjelasan Detail:

- a. *Loop* Data API : Iterasi setiap item dari data API.
- b. Kategori dan Status:

- Menggunakan fungsi *insertGetId* untuk mengecek apakah data kategori atau status sudah ada.
 - Jika sudah ada, kembalikan ID.
 - Jika belum ada, masukkan data baru dan dapatkan ID-nya.
- c. Produk : Data produk disimpan ke tabel produk dengan ID kategori dan status yang diperoleh.
- d. *Feedback* : Menyimpan pesan sukses ke *session flashdata* untuk ditampilkan di halaman.

3. **ModelProducts : InsertGetId**

Fungsi:

- Mengecek apakah data tertentu sudah ada di tabel.
- Jika data ada, kembalikan ID.
- Jika data tidak ada, masukkan data baru dan kembalikan ID.

Penjelasan Detail:

- a. Pengecekan Data:
- Menggunakan *where* untuk mencari data yang cocok di tabel.
 - Jika data ada (*num_rows* > 0), ID data yang sudah ada dikembalikan.
- b. Penambahan Data:
- Jika data tidak ditemukan, data baru ditambahkan dengan *insert*.
 - ID dari data yang baru dimasukkan dikembalikan menggunakan *insert_id*.

C. Lihat Produk

1. **ControllerProducts : viewAllProducts**

Fungsi:

- Mengambil data produk dari database, baik semua produk maupun berdasarkan status tertentu.
- Mengirim data ke *view* untuk ditampilkan.

Penjelasan Detail:

- a. Mengambil Status dari *Input*:
- Status diperoleh dari POST (*dropdown* atau filter status).
 - Jika tidak ada *input* status, secara default akan diisi 'semua'.
- b. Pengaturan Data:

- Jika status adalah 'bisa dijual', data yang diambil hanya yang memiliki status tersebut menggunakan model *getProductsByStatus*.
- Jika status bukan 'bisa dijual', semua data diambil menggunakan model *getAllProducts*.

c. Mengirim Data ke View:

- Data produk, status yang dipilih, dan judul halaman dikirim ke view.

2. **ModelProducts** : *getAllProducts*

Berfungsi mengambil semua data produk dari database dengan *join* ke tabel kategori dan status.

Penjelasan Detail:

a. *Query Builder*:

- Menggunakan *select* untuk memilih kolom dari tabel produk, kategori, dan status.
- *Join* digunakan untuk menggabungkan tabel berdasarkan relasi kategori_id dan status_id.
- Data diurutkan berdasarkan nama_produk secara *ascending*.

b. Hasil *Query*:

- *result()* digunakan untuk mengembalikan hasil *query* dalam bentuk *array* objek.

3. **ModelProducts** : *getProductsByStatus*

Berfungsi mengambil data produk dengan status tertentu dari database.

Penjelasan Detail:

a. *Query Builder*:

- Mirip dengan *getAllProducts*, namun ditambahkan kondisi *where* berdasarkan status.nama_status.
- Hanya data produk dengan status tertentu yang diambil.

b. Hasil *Query*:

- Data diurutkan berdasarkan nama_produk secara *ascending*.
- Mengembalikan hasil dalam bentuk *array* objek.

D. Tambah Produk

1. **ControllerProducts** : *addProduct*

Fungsi:

- Menampilkan halaman *form* untuk menambahkan produk baru.
- Mengambil data kategori dan status dari database untuk digunakan dalam *dropdown*.

Detail:

- Judul Halaman:
 - Mengatur judul halaman (*\$title*) untuk digunakan di *view*.
- Mengambil Data Kategori dan Status:
 - Menggunakan model *getAllCategories* dan *getAllStatuses* untuk mendapatkan daftar kategori dan status dari tabel terkait.
- Mengirim Data ke View:
 - Data kategori dan status dikirimkan ke *view* untuk *dropdown*.

2. **ModelProducts : getAllCategories dan ModelProducts : getAllStatuses**

Berfungsi mengambil semua data dari tabel kategori dan status.

Detail:

- Data diambil dengan *query* sederhana menggunakan *get*.
- Hasil dikembalikan dalam bentuk *array*.

3. **ModelProducts : getMaxIdProduct**

Fungsi:

- Mengambil nilai maksimum dari kolom *id_produk* di tabel produk.
- Digunakan untuk menentukan ID produk baru secara otomatis.

Detail:

- *select_max* digunakan untuk mengambil nilai terbesar dari kolom *id_produk*.
- Jika belum ada data, nilai *default* yang dikembalikan akan NULL.

4. **ControllerProducts : addProductData**

Berfungsi memproses data *form* yang dikirimkan untuk menambahkan produk baru ke database.

Detail:

- Validasi Form:
 - Aturan validasi untuk setiap *field* (wajib diisi, harus angka, dll.).
 - Jika validasi gagal, kembali ke halaman *form* dengan pesan *error*.
- Persiapan Data:
 - ID produk baru dihitung berdasarkan nilai maksimum yang sudah ada di database.

- Data *form* dikumpulkan ke dalam *array*.
- c. Simpan Data ke Database:
- Data produk baru dimasukkan ke tabel produk menggunakan model *insertProduct*.
- d. Pesan dan *Redirect*:
- Jika berhasil, menampilkan pesan sukses dan kembali ke halaman daftar produk.

5. **ModelProducts** : *insertProduct*

Berfungsi menambahkan data baru ke tabel produk menggunakan SQL *insert* pada tabel produk dengan data yang diberikan.

E. Ubah Produk

1. **ControllerProducts** : *updateProduct*

Fungsi:

- Menampilkan *form* untuk mengedit data produk berdasarkan ID.
- Mengambil data produk dari database, beserta data kategori dan status untuk *dropdown*.

Langkah-langkah:

- a. Judul Halaman:
- Menentukan *\$title* untuk digunakan di *view*.
- b. Ambil Data Produk:
- Menggunakan *getProductById* untuk mendapatkan data produk berdasarkan ID.
- c. Validasi Keberadaan Produk:
- Jika produk ditemukan, data dikirim ke *view*.
 - Jika produk tidak ditemukan, pesan *error* dikirim menggunakan *flashdata*, lalu *redirect* ke halaman daftar produk.

2. **ModelProducts** : *getProductById*

Fungsi:

- Mengambil data produk tertentu berdasarkan *id_produk* dengan *join* ke tabel kategori dan status.

Detail:

- *Query* menggunakan *select* dengan *join* ke tabel kategori dan status.

- Mengembalikan data sebagai *array* (karena data produk spesifik hanya satu).

3. **ControllerProducts : updateProductData**

Fungsi:

- Memproses data yang dikirimkan dari *form* untuk memperbarui data produk di database.

Langkah-Langkah:

a. Validasi *Form*:

- Aturan validasi untuk memastikan data lengkap dan valid.

b. Jika Validasi Gagal:

- Kembali ke *form* dengan pesan *error* menggunakan metode *updateProduct*.

c. Jika Validasi Berhasil:

- Data dari *form* dikumpulkan dalam *array*.
- Data diperbarui di database menggunakan model *updateProduct*.

d. Pesan dan *Redirect*:

- Jika berhasil, pesan sukses dikirim menggunakan *flashdata*, lalu *redirect* ke halaman daftar produk.

4. **ModelProducts : updateProduct**

Berfungsi memperbarui data produk di tabel produk berdasarkan id_produk.

Detail:

- Menggunakan *where* untuk menentukan ID produk.
- *update* digunakan untuk memperbarui data dengan *array* yang diberikan.

F. Hapus Produk

1. **ControllerProducts : deleteProduct**

Fungsi:

- Menghapus data produk dari database berdasarkan ID.
- Menampilkan pesan keberhasilan atau kegagalan penghapusan produk.

Langkah-Langkah:

a. Ambil Data Produk:

- Menggunakan model *getProductById* untuk memeriksa apakah produk dengan ID yang diberikan ada di database.

b. Validasi Keberadaan Produk:

- Jika produk ditemukan, maka data dihapus menggunakan model *deleteProduct*.

- Jika produk tidak ditemukan, menampilkan pesan *error* menggunakan *flashdata*.

c. *Redirect*:

- Setelah proses selesai, pengguna diarahkan kembali ke halaman daftar produk.

2. *ModelProducts : deleteProduct*

Berfungsi menghapus data produk berdasarkan *id_produk*.

Langkah-Langkah:

a. Gunakan Kondisi *where*:

- Menentukan produk yang akan dihapus berdasarkan ID.

b. Hapus Data:

- Menggunakan *delete* pada tabel produk.

G. Views

1. *Template Header (Templates/header)*

Fungsi:

- Berisi elemen-elemen standar yang ada di semua halaman (misalnya, tag HTML dasar, Bootstrap CSS, dan navbar).

Poin Penting:

- Navbar memiliki tautan utama menuju halaman produk.
- Bootstrap CDN digunakan untuk *styling* responsif.

2. *Template Footer (Templates/footer)*

Fungsi:

- Berisi elemen penutup HTML dan Bootstrap JavaScript *bundle* untuk interaktivitas.

Poin Penting:

- Ditutup dengan elemen *div container* utama dari *header*.
- Mengimpor *script* untuk Bootstrap.

3. *Daftar Produk (Products/viewProducts)*

Fungsi:

- Menampilkan seluruh daftar produk dari database dengan opsi filter, sinkronisasi, dan tombol aksi (Tambah, Edit, Hapus).

Poin Penting:

- a. Filter Produk:
 - *Dropdown* untuk memilih status produk (semua atau bisa dijual).
 - Menggunakan *onchange* untuk mengirimkan filter secara otomatis.
- b. Tabel Produk:
 - Ditampilkan dengan kolom: No, Nama Produk, Harga, Kategori, Status, dan Aksi.
 - Jika data kosong, menampilkan pesan "Tidak ada data produk."
- c. Tombol Aksi:
 - Edit: Mengarahkan ke halaman ubah produk.
 - Hapus: Konfirmasi sebelum penghapusan produk.

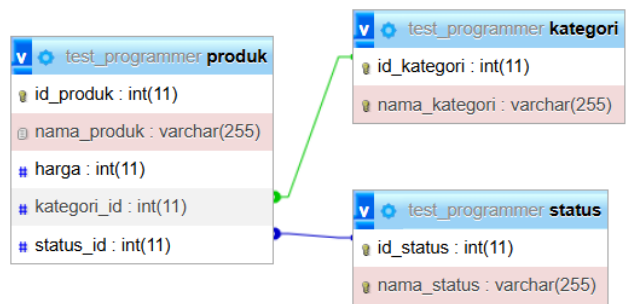
4. Formulir Produk (*Products/formProduct*)

Berfungsi untuk menambah atau mengubah data produk.

Poin Penting:

- a. *Form* Dinamis:
 - Formulir berfungsi ganda, baik untuk menambah atau mengedit produk.
 - URL *action* disesuaikan berdasarkan apakah produk tersedia di data.
- b. Validasi:
 - Menampilkan pesan *error* untuk *input* yang tidak valid.
 - *Dropdown* Kategori dan Status:
 - Menggunakan data dari *controller* yang diambil melalui model.
- c. Tombol Aksi:
 - Simpan Perubahan/Tambah Produk: Mengirim data ke *controller*.
 - Kembali: Mengarahkan kembali ke daftar produk.

1.4 Screenshot Website



Gambar 2 Database

Produk					
Daftar Produk					
Filter Produk: Semua Produk					
#	Nama Produk	Harga	Kategori	Status	Aksi
1	ALCOHOL GEL POLISH CLEANSER GP-CLN01	Rp12.500	L QUEENLY	bisa dijual	Edit Hapus
2	ALUMUNIU FOIL ALL IN ONE BULAT 23mm IM	Rp1.000	L MTH AKSESORIS (IM)	bisa dijual	Edit Hapus
3	ALUMUNIU FOIL ALL IN ONE BULAT 30mm IM	Rp1.000	L MTH AKSESORIS (IM)	bisa dijual	Edit Hapus
4	ALUMUNIU FOIL ALL IN ONE SHEET 250mm IM	Rp12.500	L MTH AKSESORIS (IM)	tidak bisa dijual	Edit Hapus
5	ALUMUNIU FOIL HDPE/PE BULAT 23mm IM	Rp12.500	L MTH AKSESORIS (IM)	bisa dijual	Edit Hapus
6	ALUMUNIU FOIL HDPE/PE BULAT 30mm IM	Rp1.000	L MTH AKSESORIS (IM)	bisa dijual	Edit Hapus
7	ALUMUNIU FOIL HDPE/PE SHEET 250mm IM	Rp13.000	L MTH AKSESORIS (IM)	tidak bisa dijual	Edit Hapus
8	ALUMUNIU FOIL PET SHEET 250mm IM	Rp1.000	L MTH AKSESORIS (IM)	tidak bisa dijual	Edit Hapus
9	ARM PENDEK MODEL U	Rp13.000	L MTH AKSESORIS	bisa dijual	Edit

Gambar 3 View Produk

Daftar Produk					
Filter Produk: Bisa Dijual					
#	Nama Produk	Harga	Kategori	Status	Aksi
1	ALUMUNIU FOIL ALL IN ONE BULAT 23mm IM	Rp1.000	L MTH AKSESORIS (IM)	bisa dijual	Edit Hapus
2	ALUMUNIU FOIL ALL IN ONE BULAT 30mm IM	Rp1.000	L MTH AKSESORIS (IM)	bisa dijual	Edit Hapus
3	ALUMUNIU FOIL HDPE/PE BULAT 23mm IM	Rp12.500	L MTH AKSESORIS (IM)	bisa dijual	Edit Hapus
4	ALUMUNIU FOIL HDPE/PE BULAT 30mm IM	Rp1.000	L MTH AKSESORIS (IM)	bisa dijual	Edit Hapus
5	ARM PENDEK MODEL U	Rp13.000	L MTH AKSESORIS (IM)	bisa dijual	Edit Hapus
6	ARM SUPPORT PENDEK POLOS	Rp13.000	L MTH TABUNG (LK)	bisa dijual	Edit Hapus
7	ARM SUPPORT T (IMPORT)	Rp13.000	L MTH AKSESORIS (IM)	bisa dijual	Edit Hapus
8	ARM SUPPORT T - MODEL 1 (LOKAL)	Rp10.000	L MTH TABUNG (LK)	bisa dijual	Edit Hapus

Gambar 4 Filter Produk Bisa Dijual

Produk

Tambah Produk

Nama Produk

Harga

Kategori

Pilih Kategori

Status

Pilih Status

Tambah Produk

Kembali

Gambar 5 *Form* Tambah Produk

Produk

Edit Produk

Nama Produk

Harga

Kategori

L QUEENLY

Status

bisa dijual

Simpan Perubahan

Kembali

Gambar 6 *Form* Edit Produk

localhost says

Apakah Anda yakin ingin menghapus produk?

OK

Cancel

Sinkronisasi Data

Tambah Produk

	Harga	Kategori	Status	Aksi
1	Rp12.500	L QUEENLY	bisa dijual	<div>Edit</div> <div>Hapus</div>
m IM	Rp1.000	L MTH AKSESORIS (IM)	bisa dijual	<div>Edit</div> <div>Hapus</div>
m IM	Rp1.000	L MTH AKSESORIS	bisa dijual	<div>Edit</div>

Gambar 7 Konfirmasi Hapus Produk

Produk

Daftar Produk

Sinkronisasi Data

Tambah Produk

Produk berhasil dihapus.

Filter Produk:

Semua Produk

#	Nama Produk	Harga	Kategori	Status	Aksi
1	ALUMUNIUM FOIL ALL IN ONE BULAT 23mm IM	Rp1.000	L MTH AKSESORIS (IM)	bisa dijual	<div>Edit</div> <div>Hapus</div>
2	ALUMUNIUM FOIL ALL IN ONE BULAT 30mm IM	Rp1.000	L MTH AKSESORIS (IM)	bisa dijual	<div>Edit</div> <div>Hapus</div>
3	ALUMUNIUM FOIL ALL IN ONE SHEET 250mm IM	Rp12.500	L MTH AKSESORIS (IM)	tidak bisa dijual	<div>Edit</div> <div>Hapus</div>
4	ALUMUNIUM FOIL HDPE/PE BULAT 23mm IM	Rp12.500	L MTH AKSESORIS -- --	bisa dijual	<div>Edit</div>

Gambar 8 Notifikasi