

RANCANG BANGUN APLIKASI PENJUALAN DAN INVENTORI APOTEK BERBASIS WEBSITE MENGGUNAKAN FRAMEWORK LARAVEL

Moh. Nabalah Fikri Azky

D3 Manajemen Informatika, Fakultas Teknik, Universitas Negeri Surabaya

Email : mohazky16050623024@mhs.unesa.ac.id

Asmunin

Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

Email : asmunin@unesa.ac.id

Abstrak

Dari sekian banyak apotek yang ada di Indonesia beberapa diantaranya masih melakukan semuanya secara manual, baik dalam hal pengecekan akan penyimpanan barang maupun pembukuan mengenai pengeluaran dan juga pendapatan apotek. Informasi mengenai jumlah barang masuk dan keluar, pengeluaran dan pendapatan apotek merupakan hal yang sangat penting, apabila terjadi kesalahan sekecil apapun terhadap informasi mengenai jumlah barang dan pembukuan mampu membuat pihak apotek mengalami kerugian dan mampu mengakibatkan apotek tersebut bangkrut. Dibuatnya aplikasi ini dengan menggunakan *framework laravel* agar aplikasi terlihat rapi dan memudahkan programmer dalam pengembangannya. Maka dari itu dibuatlah aplikasi Apotek-Ku berbasis website menggunakan *framework laravel* untuk membantu pihak apotek agar dapat meminimalisir kesalahan yang terjadi dalam hal penyimpanan barang dan juga pembukuan agar dapat bersaing dengan apotek lainnya supaya mendapatkan keuntungan dan tidak mengalami kerugian.

Kata Kunci : *Apotek, Penyimpanan, Informasi.*

Abstract

Of the many pharmacies in Indonesia, some of them still do everything manually, both in terms of checking goods inventory and bookkeeping regarding pharmacy expenses and also income. Information regarding the number of goods entering and leaving, pharmacy expenditure and income is very important, if there is the slightest error in information regarding the number of goods and bookkeeping, it can cause the pharmacy to suffer losses and can result in the pharmacy going bankrupt. This application is made using *the laravel framework* so that the application looks neat and makes it easier for programmers to develop. There fore a website-based Apotek-Ku application was created using *the laravel framework* to help pharmacies minimize errors that occur in terms of storing goods and also bookkeeping so that they can compete with other pharmacies in order to get profit and not suffer losses.

Keywords: *Pharmacy, Inventory, Information.*

PENDAHULUAN

Perkembangan teknologi di Indonesia pada era saat ini melalui proses yang relatif singkat dan pesat, dengan berkembangnya teknologi memudahkan manusia dalam mencari informasi yang dapat membantu menyelesaikan suatu masalah dengan keputusan yang tepat. Suatu informasi harus disertai bukti kuat agar menjadi sebuah informasi yang akurat dan dapat dipercaya. Apotek adalah tempat yang digunakan untuk menjual obat berdasarkan resep dokter dan juga peralatan medis lainnya. Pengecekan stok barang dan perhitungan laporan merupakan sebuah informasi penting yang harus dilakukan dengan benar dan akurat agar tidak terjadi kesalahan.

Pandemi virus corona atau yang lebih dikenal dengan sebutan *Covid – 19* merupakan wabah menular yang berbahaya sampai mampu membuat seseorang yang terjangkit virus tersebut meninggal dunia, sehingga terjadi kepanikan dimana mana. Penularan virus corona di Indonesia menyebar dengan sangat cepat, karena penularan dapat terjadi ketika seseorang batuk, bersin, berbicara, dan bersalaman. Para dokter dan medis mewajibkan kepada setiap masyarakat untuk selalu menjaga jarak satu sama lain dan wajib menggunakan masker dan membawa hand sanitizer.

Pada awal masa pandemi virus corona, karena banyak permintaan akan barang yang diwajibkan seperti masker dan hand sanitizer sering terjadi kesalahan yang dilakukan oleh pegawai apotek pada saat pengecekan stok barang dan juga perhitungan untuk laporan bulanan. Kesalahan kecil seperti itu mampu membuat apotek mengalami kerugian yang

sangat besar dan bisa membuat apotek tersebut bangkrut.

Dengan latar belakang itulah, pembuatan aplikasi “Apotek-Ku” sangat diperlukan, untuk membantu pegawai apotek dalam mencatat stok barang dan laporan agar tidak terjadi kesalahan. Dibuatnya aplikasi ini dengan menggunakan *framework laravel* agar aplikasi terlihat rapi dan memudahkan programmer dalam pengembangannya.

Rancang Bangun

Rancang bangun merupakan suatu proses perancangan hasil analisis yang di dapatkan dari data yang ada sehingga mendapatkan suatu sistem yang baru atau memperbaiki sistem yang telah ada. Kata rancang adalah langkah yang dilakukan dengan memasukkan data dari hasil analisis kedalam bahasa pemrograman untuk menjelaskan detail komponen yang akan diimplementasikan. Kata bangun adalah suatu langkah untuk membuat sistem baru atau memperbaiki system yang ada secara keseluruhan.

Persediaan (*Inventory*)

Persediaan adalah barang yang disimpan dan akan digunakan atau disediakan kepada konsumen untuk memenuhi kebutuhan sehari – hari. Persediaan dibagi menjadi 3 macam yaitu, barang baku, setengah jadi dan barang jadi. Persediaan barang setengah jadi dan barang jadi akan disimpan sebelum digunakan atau dimasukkan proses produksi, sedangkan persediaan barang setengah jadi disimpan sebelum dijual kepada konsumen.

Apotek

Apotek adalah tempat dimana apoteker dan juga tenaga kefarmasian melakukan praktek

peracikan obat sesuai resep dokter. Apotek juga tempat untuk menyimpan dan menjual obat serta alat medis lainnya yang diperlukan oleh masyarakat.

Website

Website adalah kumpulan dari beberapa halaman yang berisi informasi mengenai tulisan, gambar, suara dan video dari dokumen local ataupun jarak jauh yang dapat diakses melalui browser dan terhubung ke internet. Website terbagi menjadi 2 jenis dibedakan menurut kebutuhan dan kegunaannya, website statis dan juga website dinamis. Berikut pengertian mengenai perbedaannya:

1. Website statis merupakan web yang mudah dalam proses pengembangannya, apabila ingin merubah atau menambah tampilan web tidak bisa dilakukan secara langsung pada web namun harus membuka source code terlebih dahulu dan mencari tampilan mana yang akan diganti.
2. Website dinamis merupakan web yang proses pengembangannya lebih rumit dari web statis. Tingkat kerumitan yang dimaksud ada karena pada web dinamis programmer dapat menambah, mengubah dan menghapus data secara langsung pada web tanpa harus membuka source code terlebih dahulu.

Framework Laravel

Framework adalah kerangka kerja yang terdiri dari beberapa script pada suatu program yang membantu *programmer* untuk mengatasi masalah yang ada pada program. *Framework* merupakan gabungan dari beberapa fungsi yang dimana mempermudah *programmer* agar tidak

membuat dari awal, biasa dikenal dengan *library*. Dengan menggunakan *framework*, suatu aplikasi akan terlihat tersusun rapi dan terstruktur.

Laravel merupakan salah satu dari *framework* yang banyak digunakan oleh programmer, laravel juga dikenal sebagai *framework* dinamis yang terus berubah dan berkembang secara aktif. Laravel mengusung konsep yang membuatnya populer dikalangan programmer yaitu MCV (*Model, Controller, View*) dan memiliki fitur *command line tool* yang sangat berguna dalam pengemasan *bundle* dan instalasi *bundle* melalui *command prompt*.

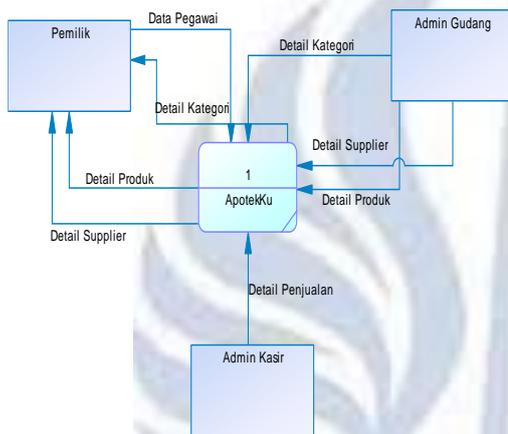
Beberapa fitur dari *framework laravel* yaitu:

1. *Application logic*, merupakan bagian dari aplikasi yang menggunakan *controller* meskipun deklarasi dari *route*.
2. *Bundles*, merupakan fitur yg ada dalam pengemasan *modular* dan *bundle* yang digunakan dalam pembuatan aplikasi.
3. *Eloquent ORM*, merupakan bagian dari penerapan PHP yang berfungsi untuk mengatasi kendala pada saat menghubungkan *script* dan *database*.
4. *Reverse Routing*, definisi hubungan antara *Link* dan *Route*.
5. *Migrations*, merupakan system yang mengatur skema *database* dan memiliki fungsi menghubungkan *database* dan *framework laravel*.
6. *View Composers*, merupakan kode yang aktif pada saat view dijalankan.
7. *Restfull Controller*, berfungsi untuk memberikan logika secara terpisah saat menjalankan *GET* atau *POST, PUT/PATCH* dan *DELETE*.

METODE

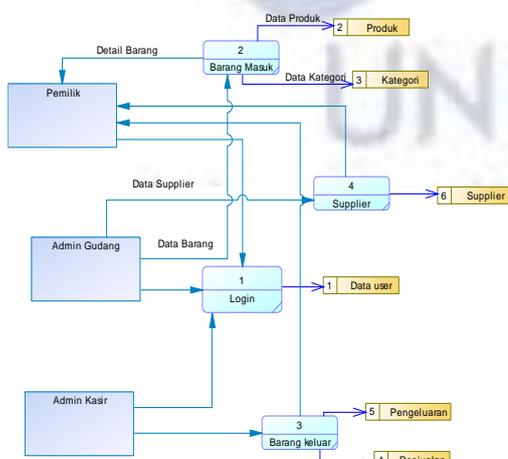
Desain DFD

Alur data yang terdapat pada aplikasi Apotek-Ku digambarkan dengan DFD (*Data Flow Diagram*), pengertian DFD yaitu suatu proses yang dibuat berdasarkan data - data yang ada kemudian di implementasikan ke dalam system yang berjalan sesuai logika, terstruktur dan jelas. Gambar dibawah ini membahas alur aplikasi pada DFD level 0:



Gambar 1. DFD Level 0

DFD level 0 yang ada pada gambar 1 diatas memiliki tiga entitas yaitu admin, admin kasir, dan admin gudang.

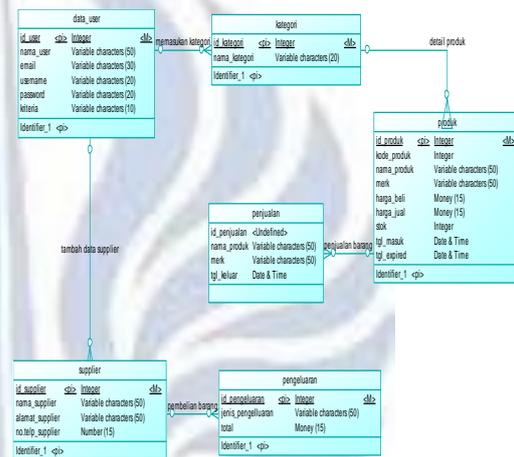


Gambar 2. DFD Level 1

DFD level 1 yang ada pada gambar 2 diatas terdapat empat aktivitas. Aktivitas pertama dari

pemilik karena memiliki hak akses untuk mengatur jalannya aplikasi. Aktivitas kedua adalah admin gudang dimana langkah pertama melakukan login agar dapat menambahkan data dari supplier, langkah kedua yang dilakukan yaitu menambahkan data barang masuk yang dimana terdiri dari data produk suatu barang dan juga data kategori suatu barang. Aktivitas terakhir dilakukan oleh admin kasir dengan menambahkan data barang keluar.

Desain CDM



Gambar 3. Desain CDM

Pada gambar 3 desain *Conceptual Data Model* (CDM) diperoleh beberapa relasi yang ada pada *database*. Pada gambar diatas terdapat enam tabel yaitu tabel *data_user*, *supplier*, *kategori*, *produk*, *penjualan*, dan *pengeluaran*. Pada tabel *data_user* memiliki enam atribut yaitu *id_user* sebagai *PK (primary key)*, *nama_user* digunakan untuk menyimpan nama dari user, *email* digunakan untuk menyimpan email dari user, *username* digunakan untuk menyimpan username dari user, *password* digunakan untuk menyimpan password dari user, kriteria digunakan untuk menyimpan kriteria yang dipilih oleh user.

Pada tabel *supplier* terdapat empat atribut yaitu *id_supplier* sebagai *primary key*,

nama_supplier digunakan untuk menyimpan nama dari supplier, alamat_supplier digunakan untuk menyimpan alamat dari supplier, no_telp_supplier digunakan untuk menyimpan no_telp dari supplier.

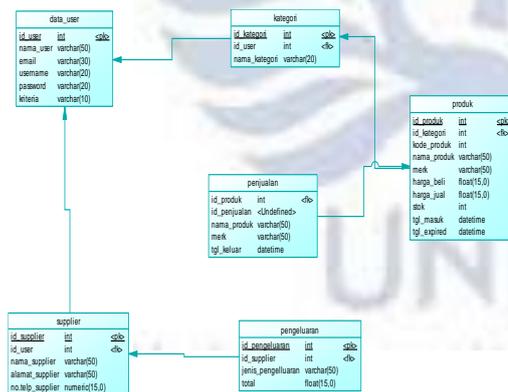
Pada tabel produk terdapat sembilan atribut yaitu id_produk sebagai *primary key*, kode_produk digunakan untuk menyimpan kode dari suatu produk, nama_produk digunakan untuk menyimpan nama dari suatu produk, merk digunakan untuk menyimpan merk dari suatu produk, harga_beli digunakan untuk menyimpan harga dari suatu barang yang dibeli dari supplier, harga_jual digunakan untuk menyimpan harga dari suatu barang yang dijual kepada customer, stok digunakan untuk menyimpan stok dari suatu barang, tgl_masuk digunakan untuk menyimpan tanggal barang yang masuk, tgl_expired digunakan untuk menyimpan tanggal expired dari suatu barang.

data_user. Dimana pada tabel supplier memiliki *primary key* “id_supplier” dan memiliki *foreign key* “id_user” sebagai penghubung antara tabel supplier dengan tabel data_user.

HASIL DAN PEMBAHASAN

Dari penjelasan diatas dapat disimpulkan bahwa sistem ini mempunyai maksud dan tujuan dalam pembuatan Rancang Bangun Aplikasi Penjualan dan Inventori Apotek Berbasis Website Menggunakan Framework Laravel. Maka keluaran dari tugas akhir ini adalah membuat aplikasi apotek berbasis website dengan design tampilan yang menarik dan dapat memudahkan pengguna dalam menyelesaikan pekerjaan. Pada tahap ini, penulis melakukan uji coba pada website. Berikut adalah contoh scenario pengujian yang dibuat:

Desain PDM



Gambar 4. Desain PDM

PDM pada aplikasi Apotek-Ku tidak berbeda jauh dengan CDM. Setelah CDM telah jadi langkah selanjutnya yang dilakukan adalah generate CDM agar menjadi *Physical Data Model* (PDM) sehingga membuat relasi yang terhubung antar tabel menjadi lebih jelas dari CDM seperti pada tabel supplier dan tabel

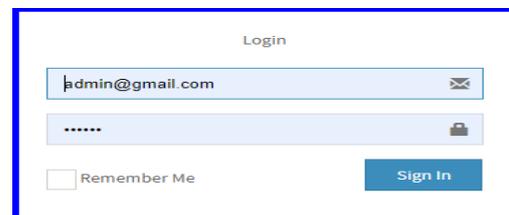
Tampilan awal Apotek-Ku



Gambar 5. Tampilan dashboard admin

Gambar 5 menjelaskan tampilan dashboard pada saat admin(pemilik) masuk ke aplikasi, admin disini memiliki hak akses tertinggi pada web dimana dapat membuat, melihat, merubah, dan menghapus semua data yang ada.

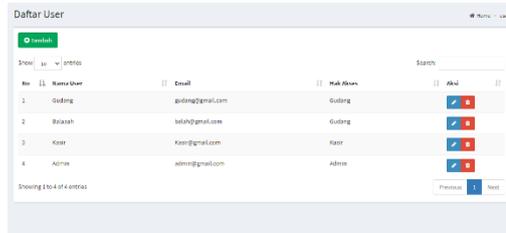
Tampilan login



Gambar 6. Tampilan Login

Gambar 6 merupakan tampilan login dari admin, admin gudang dan admin kasir pada aplikasi. Apabila user ingin melakukan login harus memasukkan email dan juga password dengan minimal 6 karakter.

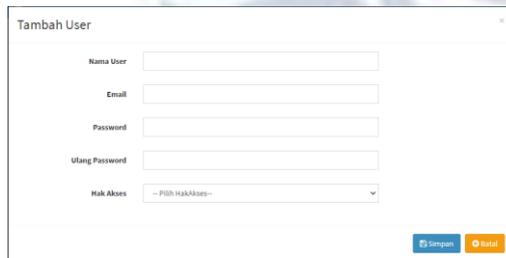
Tampilan data user



Gambar 7. Tampilan data user

Gambar 7 merupakan tampilan dari setiap user yang dapat login ke dalam aplikasi dan dapat menggunakan aplikasi. Pada tampilan ini hanya admin yang bisa mengakses nya, disini admin mampu menambahkan, merubah dan menghapus setiap user yang ada.

Tampilan form tambah data user



Gambar 8. Tampilan form tambah data user

Gambar 8 merupakan tampilan dimana admin menambahkan data user, disini data yang harus diisi oleh admin yaitu nama user, email user, password user, password ulang untuk memastikan password user, dan yang terakhir hak akses yang terdiri dari admin gudang dan admin kasir.

Tampilan data kategori



Gambar 9. Tampilan data kategori

Gambar 9 menjelaskan tentang data kategori suatu barang yang telah ditambahkan oleh admin dan admin gudang. Admin dan admin gudang dapat membuat, melihat, merubah dan menghapus data yang ada pada halaman ini.

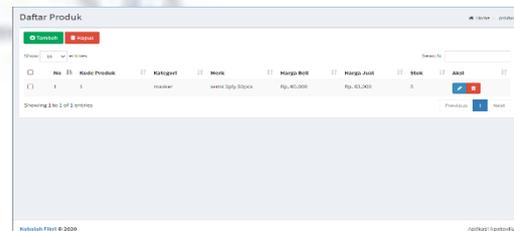
Tampilan form tambah data kategori



Gambar 10. Tampilan form tambah data kategori

Gambar 10 merupakan tampilan dimana admin dan admin gudang menambahkan data kategori barang. Disini data yang harus diisi hanya nama kategori dari suatu barang yang masuk ke dalam gudang apotek.

Tampilan data produk

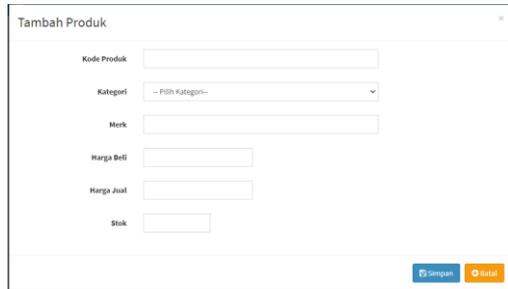


Gambar 11. Tampilan data produk

Gambar 11 menampilkan tentang data produk suatu barang yang telah ditambahkan oleh admin dan admin Gudang. Admin dan admin gudang dapat membuat, melihat,

merubah, dan menghapus data yang ada pada halaman data produk.

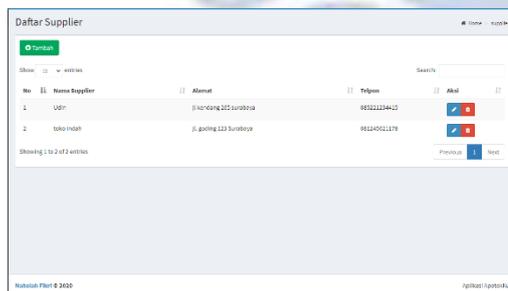
Tampilan form tambah data produk



Gambar 12. Tampilan form tambah data produk

Gambar 12 merupakan tampilan dimana admin dan admin gudang menambahkan data produk barang. Disini data yang harus diisi meliputi kode produk, kategori mengambil dari data kategori, merk dari produk, harga beli merupakan harga produk pada saat membeli dari supplier, harga jual merupakan harga produk pada saat dijual dipasaran dan dibeli oleh customer, stok produk.

Tampilan data supplier



Gambar 13. Tampilan data supplier

Gambar 13 menjelaskan tentang data supplier yang dimana hanya admin yang dapat mengaksesnya. Disini admin dapat membuat, melihat, merubah dan menghapus data yang ada pada halaman ini.

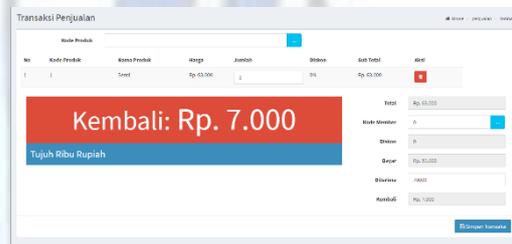
Tampilan form tambah data supplier



Gambar 14. Tampilan form tambah data supplier

Gambar 14 merupakan tampilan dimana hanya admin yang dapat mengisi data supplier. Data supplier yang harus diisi terdiri dari nama supplier, alamat supplier dan juga nomor telpon supplier.

Tampilan laporan pembayaran



Gambar 15. Tampilan laporan pembayaran

Gambar 15 merupakan tampilan dari pembayaran obat yang hanya dapat diakses oleh admin kasir. Untuk melakukan pembayaran admin kasir memasukan kode produk dan sistem akan otomatis menemukan produk tersebut, kemudian memasukan jumlah barang yang dibeli dan otomatis total pembayaran akan keluar. Terakhir admin kasir memasukan total uang pembayaran dari customer.

Tampilan laporan bulanan



Gambar 16. Tampilan laporan bulanan

Gambar 16 menampilkan pendapatan dari penjualan barang dan juga pembelian barang dari supplier. Tampilan laporan dapat berupa harian maupun bulanan.

PENUTUP

Simpulan

Berdasarkan dari analisis dan uji coba yang telah dilakukan pada aplikasi Apotek-Ku, dapat disimpulkan bahwa:

1. Pembuatan aplikasi Apotek-Ku ini dibangun dengan menggunakan framework Laravel.
2. Aplikasi Apotek-Ku ini akan mempermudah pihak apotek dalam melakukan semuanya dengan cepat dan tepat tanpa adanya kesalahan.
3. Peran admin disini untuk memantau semua kegiatan keluar masuknya barang dan juga pendapatan serta pengeluaran agar tidak ada kesalahan yang terjadi.

Saran

Berdasarkan dari kesimpulan dan juga pembahasan yang ada, aplikasi yang telah dibangun ini masih jauh dari sempurna. Dibutuhkan kejujuran dan ketelitian dari admin gudang dan juga admin kasir agar aplikasi ini dapat berjalan dengan sangat baik dan membantu.

DAFTAR PUSTAKA

Agusvianto, Hendra. 2017. *Sistem Informasi Inventory Gudang Untuk Mengontrol Persediaan Barang Pada Gudang Studi Kasus: PT. Alaisys Sidoarjo*. Surabaya: Skripsi.

Arizal. 2014. *Sistem Informasi Inventory Control Pada Perusahaan Industri Gas di PT. XYZ*. Surabaya: Skripsi.

David, Naista. 2017. *CODEIGNITER VS LARAVEL*. Jakarta: Loko Media.

Nugroho, Bunafit. 2004. *Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL*. Yogyakarta: Gava Media.

Supardi, Yuniar. dan Sulaeman, 2019. *Semua Bisa Menjadi Programmer Laravel Basic*. Jakarta: Elex Media Komputindo.