

Perancangan Aplikasi “Plement” untuk Proses Penjualan (Studi Kasus: Nott Sleepwear)

Vincent Gunaeri^{#1}, Meliana Christianti Johan^{*2}

[#]Program Studi SI Teknik Informatika, Universitas Kristen Maranatha
Jl. Surya Sumantri No.65 Bandung

¹1772001@maranatha.ac.id

²meliana.christianti@it.maranatha.edu

Abstract — Simple Management (Plement) is a web-based application that supports Enterprise Resource Planning (ERP) System with minified, tiny, and small scale. Plement is a simple management application which run in website platform. Modules implemented inside Plement are two of primary modules implemented in Enterprise Resource Planning system. There are two modules implemented which are Sales and Distribution and Purchase and Master Data. Plement provides many features such as creating and manipulating transaction document or master data. Plement also provides Report Features which is usable and useful for future planning. The application itself is developed using Laravel framework which is PHP supported.

Keywords— ERP, Laravel, Management, MiniERP, Sales.

I. PENDAHULUAN

Plement adalah sebuah aplikasi berbasis *website* yang merupakan singkatan dari *Simple Management*. Sesuai dengan namanya, *Plement* berarti sebuah aplikasi yang mendukung manajemen praktis untuk sebuah pengelolaan perusahaan yang *simple*. *Plement* dibuat dengan skala yang kecil sehingga mudah digunakan dan dipelajari, namun fitur yang diimplementasi dalam aplikasi *Plement* tidak melenceng dari *Enterprise Resource Planning* sendiri.

Aplikasi *Plement* dibuat dan dikembangkan berdasarkan sebuah studi kasus usaha kecil penjualan baju tidur bernama *Nott Sleewpear*. Proses penjualan yang dijalankan dalam *Nott Sleepwear* menggunakan media *online Instagram* dan *Shopee*, dengan menggunakan *Plement* diharapkan proses transaksi dan pengelolaan keuangan atau perencanaan bisnis dapat terkelola dengan lebih efisien dan efektif serta akurat. Dalam sebuah transaksi penjualan *Nott Sleepwear*, perusahaan akan menerima *inquiry* atau permintaan pembelian dari *customer*, kemudian setelah itu dapat dibuat sebuah dokumen sebagai bukti permintaan tersebut yang berupa *sales document*. Setelah terbuat *sales document* produk yang dipesan oleh *customer* dapat dikirimkan dan *document delivery* serta *document billing* pun dapat dibuat sebagai bukti pengiriman produk ke *customer* dan/atau sebagai bukti tagihan pembayaran yang diterima *customer*.

II. KAJIAN TEORI

A. Enterprise Resource Planning (ERP)

Enterprise Resource Planning atau kepanjangan dari ERP adalah sebuah terminologi yang diberikan kepada sistem informasi yang mendukung transaksi atau operasi sehari-hari dalam pengelolaan sumber daya perusahaan atau instansi [1]. Sumber daya tersebut dapat meliputi dana, manusia, mesin, suku cadang, produk, waktu, material dan kapasitas. Umumnya ERP sendiri dikemas dalam sebuah software atau perangkat lunak untuk membantu pengelolaan sistemnya. ERP digunakan oleh banyak perusahaan untuk melakukan integrasi proses bisnis yang berhubungan dengan operasi management perusahaan seperti production, distribution, marketing, finance, inventory, purchasing, logistic, dan lain-lain.

B. Modul Sales Distribution MiniERP

Modul *Sales and Distribution* ditekankan pada penggunaan strategi penjualan yang mampu mengantisipasi perubahan pasar. Prioritas utama dari penggunaan modul ini adalah untuk mampu merekam, menganalisis dan merencanakan serta mengontrol aktivitas untuk memberikan prediksi yang akurat, signifikan dan lebih efisien [1]. Sehingga keputusan yang dihasilkan dapat menghasilkan keuntungan yang layak dan logis. Modul *Sales and*

Distribution mengandung berbagai proses yang mendukung proses penjualan dalam sebuah sistem *ERP* yang dimiliki oleh perusahaan atau lembaga.

C. Odoo

Odoo adalah sebuah aplikasi *web* yang dibangun menggunakan bahasa pemrograman *Python*, *XML* dan *Javascript* [2]. Saat pertama kali dirilis, *Odoo* memiliki nama *TinyERP* yang kemudian diganti menjadi *OpenERP* pada tahun 2009. Setelah itu kembali dikembangkan dan berganti nama lagi menjadi *Odoo*. Aplikasi *Odoo* memiliki tingkat fleksibilitas yang tinggi sehingga dapat menjangkau hampir seluruh jenis perusahaan yang ada.

D. Web Framework

Web Framework atau merupakan kerangka kerja pengembangan aplikasi *website*. *Web Framework* digunakan dengan tujuan untuk memudahkan pengembangan atau pembuatan sebuah aplikasi *website* dengan lebih cepat dan efisien dan *framework* tersebut sendiri dapat digunakan kembali pada aplikasi lainnya yang sejenis [3]. Banyak bahasa pemrograman yang sudah memiliki *framework* seperti *Javascript* dengan *framework NodeJS*, *AngularJS*, *ReactJS*; *PHP* seperti *Code Igniter*, *Laravel*; *CSS* seperti *Bootstrap*, *Materialize*; dan lain-lain.

E. Laravel Framework

Laravel Framework merupakan salah satu *framework* yang didukung oleh bahasa pemrograman *server PHP* untuk melakukan pengembangan atau pembuatan aplikasi *website* yang diciptakan oleh Taylor Otwell pada tahun 2011 [4]. *Laravel* menggunakan struktur pola *Model View Controller* (*MVC*). Pola *MVC* pada *Laravel* memiliki sebuah fitur mirip seperti jembatan yang disebut *routing* dimana fitur ini menghubungkan antara *view* dengan *controller*. Di dalam *Laravel* sendiri pun terdapat sebuah bahasa unik yang dapat menggantikan bahasa *query* yaitu *Eloquent*.

F. Model View Controller (MVC)

Pola desain pengembangan perangkat lunak (*design pattern*) merupakan deskripsi dari gabungan *class* dan/atau *object* yang saling berkomunikasi dan tersusun secara terstruktur dan sistematis untuk memecahkan masalah secara umum pada sebuah konteks masalah tertentu [5].

Salah satu pola desain pengembangan perangkat lunak yang digunakan untuk melakukan pengembangan aplikasi adalah *Model View Controller* atau *MVC*. Pola desain ini terdiri dari tiga buah instansi pendukung yakni *Model*, *View*, dan *Controller*. *Model* adalah objek aplikasi yang umumnya merupakan representasi dari entitas pada *database*. *View* adalah sebuah instansi yang menyediakan tampilan antarmuka. *Controller* adalah penghubung antara pengguna dan tampilan yang akan mengendalikan tampilan yang diterima oleh pengguna.

G. PHP

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman yang bersifat *open-sourced* yang sangat kompatibel dan dikhususkan untuk pengembangan aplikasi *web* [6] terutama pada bagian *server-side*. Bahasa pemrograman *PHP* merupakan bahasa *scripting server-side* dimana pemrosesan terjadi pada *server*. *PHP* dapat ditanamkan pada sebuah skrip *HTML*.

H. Bootstrap

Bootstrap adalah sebuah *framework* yang digunakan khusus untuk mengembangkan *front-end* dari aplikasi *website*. *Bootstrap* merupakan *framework* yang dibentuk dalam dua bahasa yaitu *CSS* dan *Javascript*. Penggunaan *framework Bootstrap* menggunakan daftar kelas yang telah tersedia kemudian menggunakannya dengan melakukan implementasi penulisan kelas tersebut pada salah satu *object* atau *tag* pada *HTML*. *Grid system* pada *bootstrap* merupakan pengaturan ukuran yang ditampilkan pada monitor [7].

I. My Structured Query Language (MySQL)

MySQL atau kependekan dari *My Structured Query Language* adalah salah satu jenis *database server* [8]. *MySQL* merupakan salah satu jenis *RDBMS* (*Relational Database Management System*). Pada *MySQL*, sebuah *database* mengandung satu atau lebih tabel. Tabel terdiri atas kolom yang berfungsi menjadi indeks atau nama *field* dan baris yang akan menjadi *value* dari masing-masing *cell* kolom.

J. Cascading Style Sheet (CSS)

CSS adalah bahasa pemrograman yang merepresentasikan halaman *web* [9]. Dengan menggunakan CSS, seorang pengembang aplikasi dapat membuat tampilan *UI* aplikasi yang dikembangkan menjadi lebih beradaptasi dengan berbagai macam jenis ukuran layar. Pembuatan *script* CSS dapat dilakukan dengan memisahkan dan juga menggabungkannya dengan *source-code* aplikasi *web* yang diimplementasikan.

K. Javascript

JavaScript adalah sebuah bahasa pemrograman web yang bersifat *client-side Programming Language* yang artinya pemrosesan *script* dari *javascript* berlangsung dan dilakukan pada *client-side* [9]. Aplikasi *client* yang dimaksud merujuk pada *web browser* seperti *Google Chrome*, *Mozilla Firefox*, *Microsoft Edge*, *Internet Explorer*, *Opera*, dan lain-lain.

JavaScript digunakan pada pemrograman aplikasi *web* dengan mengimplementasikan fitur-fitur yang dirancang untuk mengendalikan *website* tersebut. Terdapat banyak fitur *built-in* yang bermanfaat untuk mengendalikan komponen-komponen yang ada di dalam kode *HTML*.

L. Javascript Object Notation (JSON)

JSON atau yang merupakan singkatan dari *JavaScript Object Notation* adalah sebuah *format* pesan balikan yang berukuran relatif kecil, mudah dibaca dan ditulis oleh manusia, serta juga mudah ditulis dan diurai oleh mesin [10]. *JSON* merupakan bagian dari *JavaScript* namun dapat dibaca juga oleh bahasa pemrograman lainnya seperti *C#*, *Java*, *C++*, *Dart*, dan lain-lain.

M. JQuery

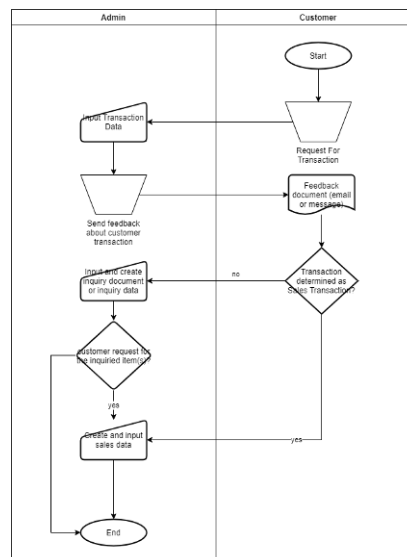
JQuery adalah *open-sourced add-on library* dari *JavaScript* yang menekankan pada interaksi antara *JavaScript* dengan *HTML* [9]. *JQuery* sendiri merupakan sebuah kode *javascript* yang telah ditulis dan disatukan membentuk sebuah *framework* dan umumnya digunakan untuk mengendalikan *event (event listener)* yang mengendalikan *website*.

III. ANALISIS DAN RANCANGAN SISTEM

A. Flowchart

Di dalam aplikasi *Plement* khususnya dalam *Sales And Distribution Module* terdapat dua buah menu utama atau proses utama yang terjadi di dalamnya. Berikut ini akan dibahas kedua proses tersebut dan direpresentasikan dalam bentuk *Flowchart*.

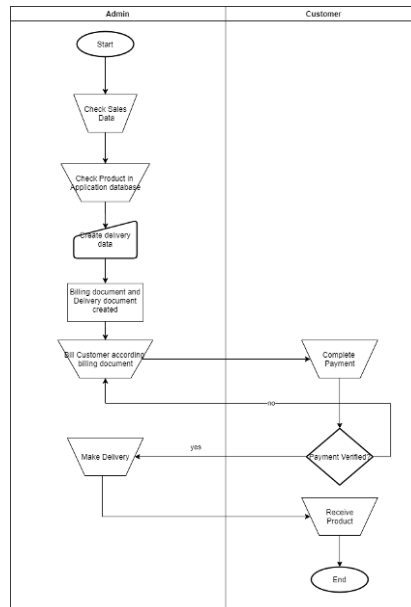
1) Flowchart Sales And Inquiry



Gambar 1 Flowchart Sales And Inquiry

Pada Gambar 1 *Flowchart Sales And Inquiry* ditunjukkan *flow* atau aliran proses yang berjalan pada saat pembuatan dokumen *sales* dan/atau *inquiry*. Dimulai dari *customer* dari luar sistem aplikasi yang melakukan *request* secara langsung ke perusahaan melalui *contact* perusahaan. Kemudian *admin* dapat melakukan pencatatan data, yang kemudian dapat menciptakan sebuah dokumen *sales* resmi perusahaan. Dokumen *sales* tersebut adalah bukti sekaligus referensi yang digunakan dalam pembuatan dokumen resmi selanjutnya.

2) *Flowchart Delivery And Billing*

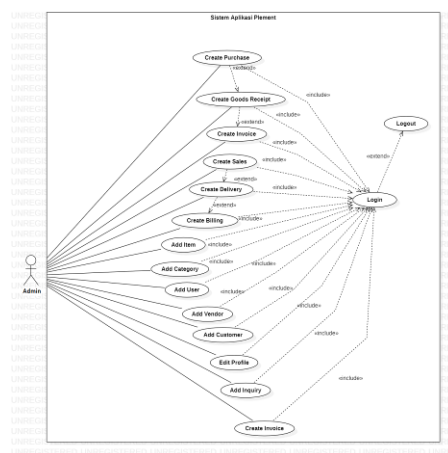


Gambar 2 Flowchart Delivery And Billing

Pada Gambar 2 ditunjukkan gambar *flowchart delivery and billing*. Merupakan sebuah *flowchart* yang merepresentasikan dan menjabarkan alur proses pembuatan dokumen *billing*. Apabila dokumen *sales* telah dibuat, maka pembuatan dokumen *delivery* dan *billing* dapat dilakukan. Setelah melakukan referensi ke dokumen *sales* maka sistem akan melakukan pembuatan dokumen *delivery* sebagai bukti pengiriman produk dan *billing* sebagai bukti tagihan pembayaran *customer*. Dengan terciptanya dua dokumen tersebut, maka sebuah transaksi dianggap selesai.

B. Use Case Diagram

Dalam aplikasi *Plement* terdapat *role* utama yaitu *admin*. *Admin* merupakan satu-satunya *user* yang menggunakan sistem aplikasi ini. *User* dapat melakukan pengelolaan data-data yang terkait dengan sistem pengelolaan *management* perusahaan. Berikut ini adalah gambar *use case diagram* yang menjabarkan fitur atau fungsi yang dapat diakses oleh *user Admin*.

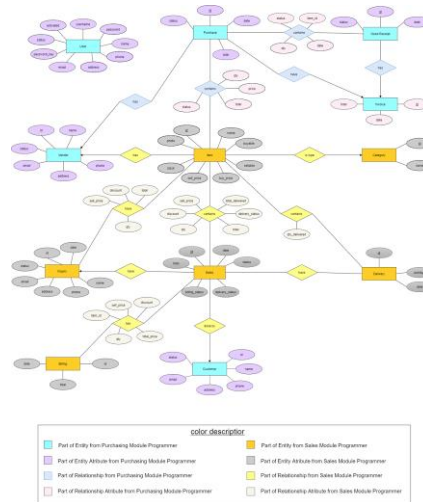


Gambar 3 Use Case Diagram Sistem Plement

Gambar 3 menunjukkan gambaran dari *use case diagram* yang merepresentasikan kasus pada setiap proses yang terimplementasi pada aplikasi *Plement*. Actor pada *use case* tersebut adalah *user Admin*. *User admin* dapat melakukan berbagai aktivitas seperti *login* dan *logout*, pembuatan dokumen transaksi (*sales, purchase, inquiry, good receipt, billing and delivery*). Selain itu, *user Admin* juga dapat melihat *report* penjualan dan/atau pembelian pada kurun waktu tertentu.

C. Rancangan Database

Aplikasi *Plement* memiliki relasi *database* yang lebih *simple* dibandingkan dengan aplikasi *Enterprise Resource Planning* yang lainnya. Beberapa dari *relasi* yang dimiliki *Plement* bertipe *one-to-many* dan sedikit ada yang bertipe *many-to-many*. Berikut ini akan ditunjukkan sketsa rancangan *database* yang digunakan untuk membangun aplikasi *Plement*.



Gambar 4 Rancangan Database Sistem Plement

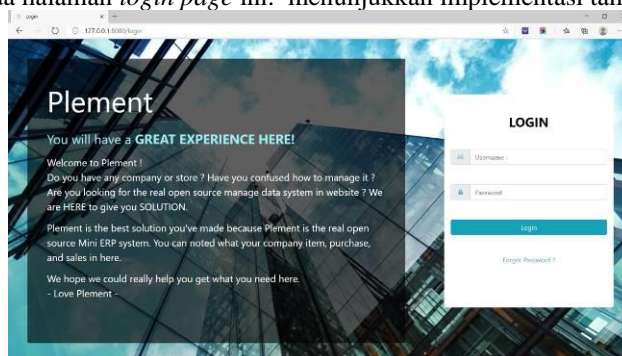
Pada Gambar 4 ditunjukkan melalui gambar sebuah *entity relational diagram* dari *database* sistem *Plement*. Terdapat sekitar 11 (sebelas) entitas berbeda yang masing-masing memiliki relasi dan terdapat sebuah entitas yang tidak terhubung oleh relasi yang entitas *user*. Terdapat kurang lebih 13 relasi yang tercantum dalam diagram tersebut.

IV. IMPLEMENTASI

Berdasarkan rancangan tampilan yang telah dibuat sebelum melakukan pengerjaan aplikasi. Implementasi pengerjaan aplikasi menyesuaikan rancangan tampilan tersebut sehingga aplikasi yang telah dikerjakan dan diimplementasi memiliki tampilan yang akan dijabarkan melalui setiap sub-bab berikut ini.

A. Tampilan Login Page

Halaman *login page* merupakan tampilan awal yang akan dilihat oleh *user* yang belum melalui proses *login* (*user* masih dalam status *guest*). Apabila *user* ingin masuk ke dalam sistem aplikasi, maka *user* perlu melakukan *login* dengan *input* yang benar dan tepat pada halaman *login page* ini. menunjukkan implementasi tampilan *login page*.

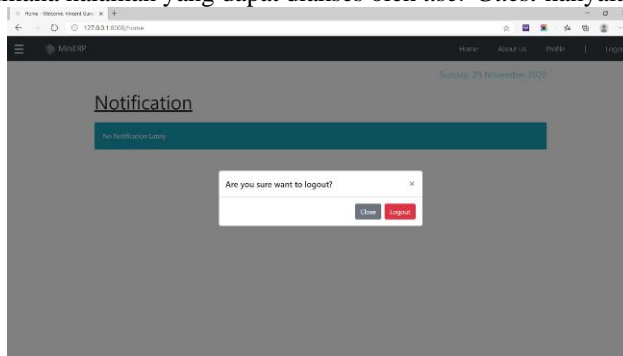


Gambar 5 Implementasi Tampilan Login Page

Pada tampilan *login page* yang ditunjukkan pada Gambar 5 dapat dilihat bahwa tampilan *login* sangatlah sederhana dimana hanya terdapat dua buah *input text* dan sebuah tombol serta sebuah *hyperlink*. Dua buah *input text* masing-masing digunakan untuk memasukkan *username* dan *password*. Tombol yang disediakan digunakan untuk melakukan proses *login*. Dan *hyperlink* bertuliskan “*Forgot Password*” digunakan untuk permintaan lupa *password* oleh *user Admin*.

B. Tampilan Logout

Halaman *logout* adalah halaman yang digunakan untuk mengakhiri sesi *login*. *User* akan segera dikeluarkan dari sesi sistem aplikasi yang kemudian akan dialihkan kembali ke halaman *login page*. Setelah melakukan *logout* status *user* akan kembali menjadi *guest* dimana halaman yang dapat diakses oleh *user Guest* hanyalah *login page*.



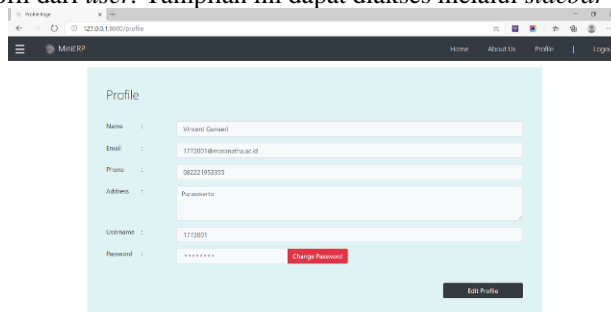
Gambar 6 Implementasi Tampilan Logout

Pada Gambar 6 dapat dilihat tampilan dari implementasi halaman *logout* apabila *user* melakukan *request logout* maka akan ditampilkan sebuah *dialog* yang menanyakan “Apakah *user* yakin ingin keluar?”. Apabila *user* menekan tombol *Close* maka permintaan dibatalkan, sebaliknya apabila *user* menekan tombol *logout*, maka *user* akan segera dikeluarkan dari sesi sistem aplikasi.

C. Tampilan Profile Page

1) Tampilan Profile Page Utama

Tampilan *profile page* adalah sebuah tampilan yang menyediakan informasi yang berkaitan dengan *user* yang sedang *login*. Di dalam tampilan *profile* terdapat fitur untuk mengubah *password* atau kata sandi dan mengubah data informasi profil dari *user*. Tampilan ini dapat diakses melalui *sidebar* atau *navigation bar*.

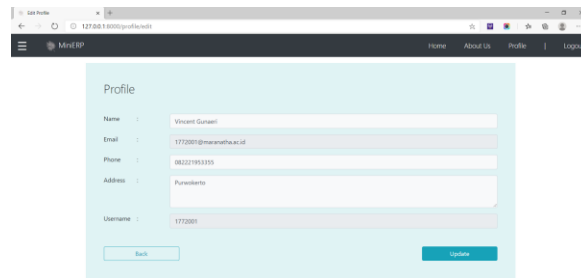


Gambar 7 Tampilan Utama Profile Page

Implementasi tampilan utama *profile page* dapat dilihat pada Gambar 7. Pada tampilan tersebut terdapat beberapa informasi mengenai akun yang dimiliki oleh *user* yang sedang *login*. Dengan menggunakan tombol *edit profile* maka *user* akan dialihkan ke halaman *edit profile page*. Terdapat pula tombol berwarna merah yang digunakan untuk memunculkan *dialog* pengubahan kata sandi/*change password*.

2) Tampilan Edit Profile Page

Tampilan *edit profile page* adalah tampilan yang dapat diakses melalui halaman *profile page*. Dengan menggunakan tampilan ini, *user* dapat melakukan perubahan data terhadap informasi akun yang dimilikinya.

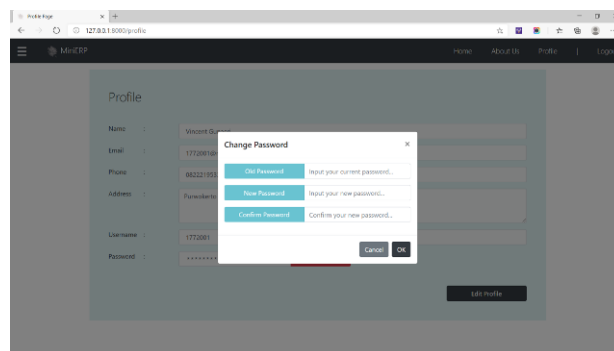


Gambar 8 Tampilan Edit Profile Page

Tampilan *edit profile page* dapat dilihat melalui Gambar 8. Pada tampilan tersebut terdapat beberapa *field input* yang bersifat *optional* (boleh diisi boleh tidak diisi untuk melakukan *submit form*) apabila seluruh data *input* sudah benar (*format, validitas, dan tidak kosong*) maka tombol *Update* untuk melakukan penyimpanan perubahan dapat ditekan dan akan melakukan proses validasi pada *server back-end*. Setelah validasi selesai maka respon akan ditampilkan pada halaman utama *profile page*.

3) Tampilan Change Password

Tampilan *change password* dapat digunakan oleh *user* untuk mengubah *password* lama dengan *password* baru. Pada *form* pengubahan *password*, *user* perlu memasukkan data *password* lama dan *password* baru. Apabila *input password* benar maka *password* akan segera diubah dan *user* dapat menggunakan *password* barunya untuk *login*.

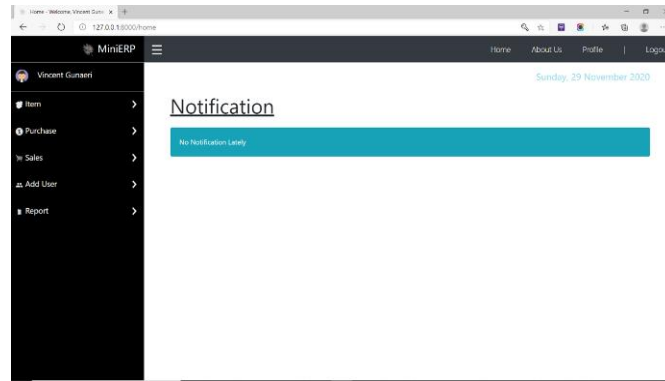


Gambar 9 Tampilan Change Password

Tampilan *change password* yang ditunjukkan pada Gambar 9 memperlihatkan tampilan sederhana yang digunakan untuk pengubahan data *password*. Tampilan ini hanya dapat diakses oleh pengguna yang sudah *login*. Untuk mengubah data *password* diperlukan 3 buah *input* diantara lain adalah *password* lama, *password* baru dan konfirmasi *password* baru. Apabila ketiga *input* sudah tepat dan valid, maka data *password* dapat diubah di *database*.

D. Tampilan Home Page

Tampilan *home page* menampilkan notifikasi pembelian item/material yang merupakan produk jual/beli dari perusahaan. Notifikasi berfungsi untuk memperlihatkan harga terbaru yang dimiliki oleh item atau produk tertentu.



Gambar 10 Implementasi Tampilan Home Page

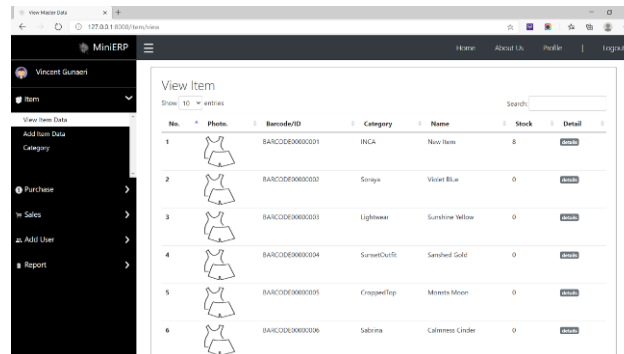
Pada Gambar 10 dapat dilihat implementasi dari tampilan *home page*. Tampilan *homepage* merupakan tampilan utama yang didesain secara *simple* agar ramah digunakan oleh pengguna. Dengan menyediakan tanggal sebagai penunjuk waktu dan notifikasi sebagai menu utama *home page* membuat halaman ini lebih efektif dan dapat lebih berguna sebagai *welcoming page* karena memiliki informasi yang cukup berguna.

E. Tampilan Menu Item

Di dalam menu item terdapat tiga *sub menu* yang dapat diakses oleh *user* yaitu *sub-menu View Item*, *sub-menu Add Item*, dan *sub-menu Category*. Setiap *sub-menu* memiliki fungsi dan tampilannya sendiri. Berikut ini adalah detail dari setiap *sub-menu* yang ada pada *menu item* tersebut.

1) Tampilan Sub-Menu View Item

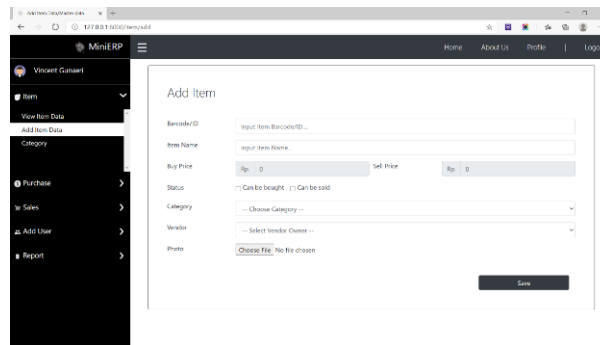
View Item Page merupakan tampilan yang menampilkan semua data item atau *master material data* yang ada atau telah diinput oleh *user* ke dalam system. Di setiap baris tampilan data yang dimunculkan terdapat sebuah tombol *details* yang berfungsi untuk mengalihkan *user* ke halaman *detail item* yang berkaitan (detail item yang dipilih sesuai dengan baris item tersebut). Terdapat sebuah fitur search yang dapat digunakan pada *search bar* pada bagian atas tampilan tabel data. Search bar tersebut dapat digunakan untuk mencari data item yang tercantum di dalam tabel data.



Gambar 11 Tampilan Sub-Menu View Item

2) Tampilan Sub-Menu Add Item

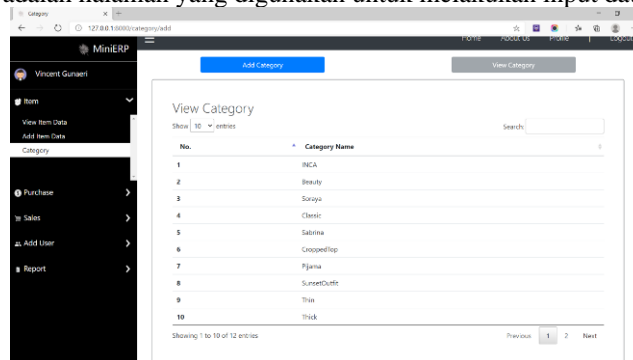
Sub-menu *add item* adalah tampilan yang digunakan untuk melakukan input *master* atau *material* atau *item* data. Dengan menggunakan tampilan ini *user* dapat memasukkan *master* data seperti nama produk, id atau *barcode* produk, status dari produk tersebut, vendor dan category dari produk tersebut serta dapat meng-*upload* foto dari produk tersebut. Setelah melakukan input system akan melakukan validasi terhadap input-input yang wajib dimasukkan, apabila tervalidasi maka system akan segera melakukan *insert* ke dalam *database*. Namun apabila terjadi kesalahan input atau kesalahan server, maka *user* akan mendapatkan sebuah pemberitahuan pada saat layar dialihkan.



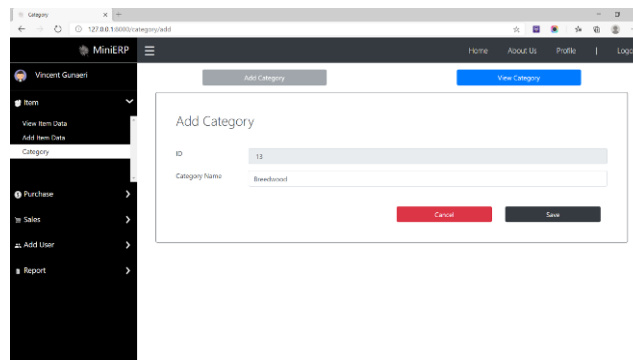
Gambar 12 Tampilan Sub-Menu Add Item

3) Tampilan Sub-Menu Category

Di dalam sub-menu *category* terdapat dua buah halaman utama yang dapat dinavigasikan secara statis di dalam halaman sub-menu tersebut. Kedua halaman tersebut adalah halaman *view category* dan *add category*. Halaman *view category* merupakan halaman yang menampilkan seluruh data *category* yang tercantum dalam *database* dan *add category* adalah halaman yang digunakan untuk melakukan input data *category*.



Gambar 13 Tampilan Sub-Menu Category



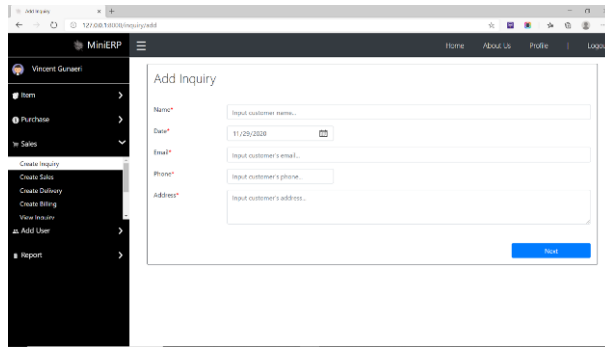
Gambar 14 Tampilan Sub-Menu Category (2)

F. Tampilan Menu Sales

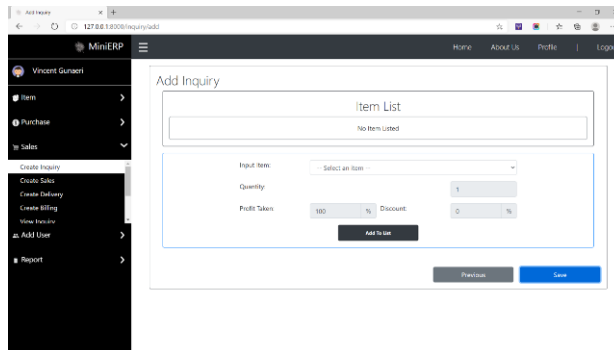
Menu *sales* merupakan menu yang mendukung proses fitur utama yang ada di dalam sistem aplikasi *Plement* ini. Terdapat 8 (delapan) buah sub-menu di dalam menu *sales* yakni *Create Inquiry*, *Create Sales*, *Create Delivery*, *Create Billing*, *View Inquiry*, *View Sales*, *View Billing*, dan *View Delivery*.

1) Tampilan Sub-Menu Create Inquiry

Sub-Menu pertama dari Menu *Sales* adalah *create inquiry*. Di dalam *sub-menu* ini terdapat fitur untuk membuat data *inquiry* dari sebuah pencarian produk oleh *customer*. Fitur ini diinput secara manual melalui aplikasi oleh *user*. Fitur *inquiry* sendiri berfungsi untuk mencatat pencarian atau permintaan produk dari *user* sehingga dapat diketahui data yang banyak diminati oleh *customer*.



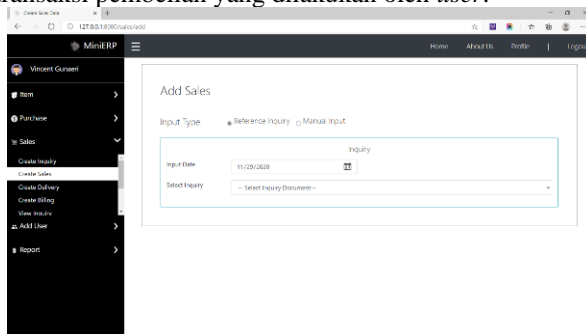
Gambar 15 Tampilan Sub-Menu Create Inquiry



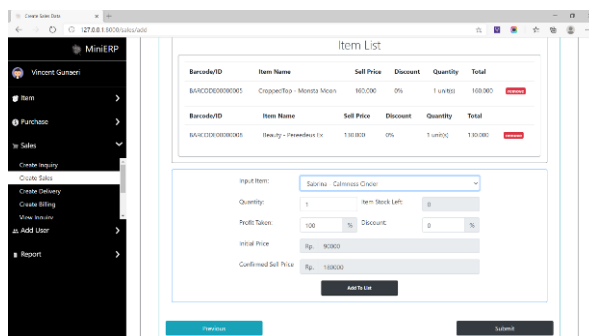
Gambar 16 Tampilan Sub-Menu Create Inquiry (2)

2) Tampilan Sub-Menu Create Sales

Sub-menu kedua setelah *create inquiry* adalah *create sales* dimana dalam sub-menu ini *user* atau *admin* dapat melakukan pembuatan dokumen atau data *sales*. Data *sales* dalam aplikasi *Plement* ini merupakan sebuah data yang merepresentasikan transaksi pembelian yang dilakukan oleh *user*.



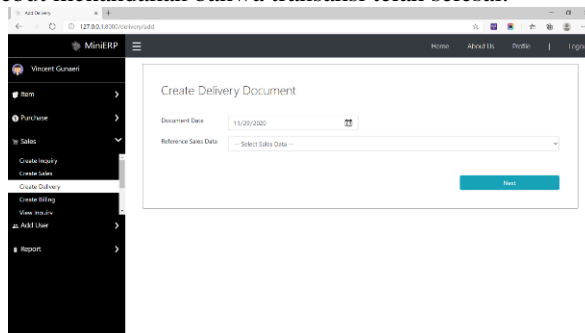
Gambar 17 Tampilan Sub-Menu Create Sales



Gambar 18 Tampilan Sub-Menu Create Sales (2)

3) Tampilan Sub-Menu Create Delivery

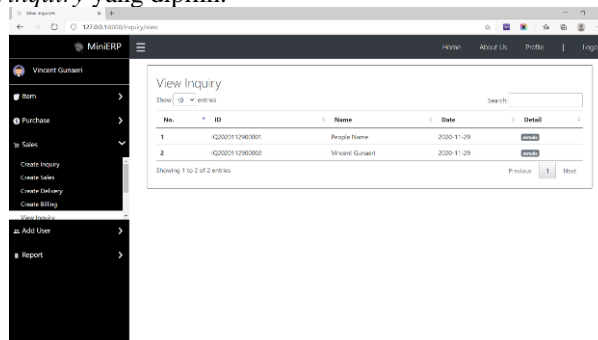
Sub-menu ketiga adalah pembuatan dokumen pengantaran atau *delivery document*. Sub-menu *Create Delivery* digunakan untuk membuat dokumen pengantaran atau *delivery document* sesuai dengan data *sales* yang telah diinput sebelumnya. Kemudian dokumen *billing* pun otomatis terbuat bersamaan dengan terbuatnya dokumen *delivery*. Kedua dokumen tersebut menandakan bahwa transaksi telah selesai.



Gambar 19 Tampilan Sub-Menu Create Delivery

4) Tampilan Sub-Menu View Inquiry

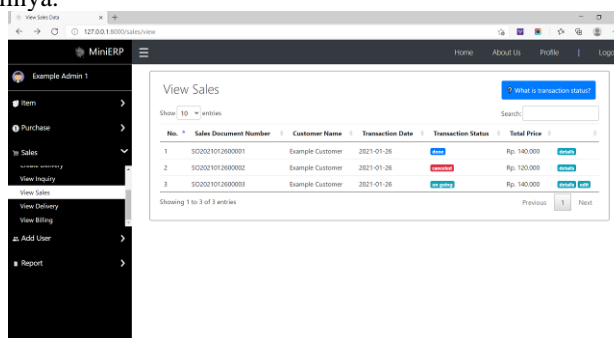
Sub-menu *view inquiry* adalah sub-menu keempat setelah sub-menu *create billing*. Pada sub-menu ini *user* dapat melihat semua daftar *inquiry data* yang telah diproses dan diinput sebelumnya. Terdapat juga halaman untuk melihat detail dari data *inquiry* yang dipilih.



Gambar 20 Tampilan Sub-Menu View Inquiry

5) Tampilan Sub-Menu View Sales

Sub-menu *view sales* digunakan untuk melihat semua data *sales* yang ada dan tercantum dalam *database* sistem. Data ditampilkan menggunakan tabel, di dalam tabel terdapat fitur pencarian dan dapat memilih salah satu *entry data* untuk dilihat detailnya.

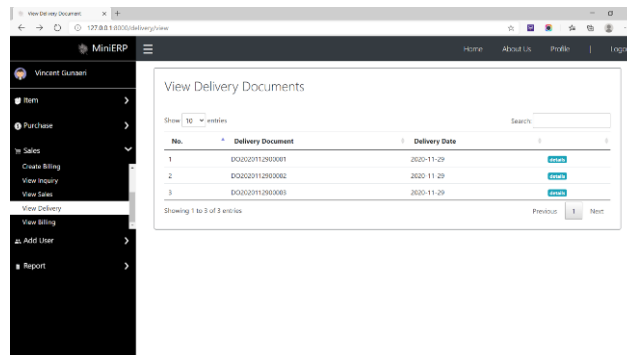


Gambar 21 Tampilan Sub-Menu View Sales

6) Tampilan Sub-Menu View Delivery

Sub-menu ini sama seperti sub-menu *sales* dan *inquiry* dimana digunakan untuk melihat semua data yang ada pada *database* sistem. Perbedaannya adalah pada sub-menu ini menampilkan data *delivery*. Data *delivery* ditampilkan dalam bentuk tabel, setiap baris merepresentasikan satu data unik. Di setiap baris terdapat sebuah

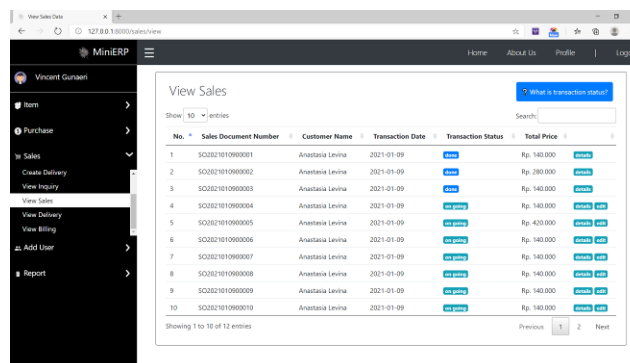
tombol *details* untuk mengakses halaman yang menampilkan detail data dari *delivery* yang dipilih. Dalam sub-menu *delivery* sendiri terdapat sebuah tampilan yang memperlihatkan dokumen pengiriman (*delivery document*) yang dapat dicetak.



Gambar 22 Tampilan Sub-Menu View Delivery

7) Tampilan Sub-Menu View Billing

Sub-menu ini pun sama seperti ketiga sub-menu sebelumnya. Sub menu digunakan untuk melihat seluruh data *billing* yang ada atau tercantum di dalam *database*. Setiap *billing* memiliki sebuah fungsi untuk menentukan status dari *billing* tersebut sudah selesai atau belum selesai. Fitur tersebut dapat diakses melalui halaman detail data *billing*.



Gambar 23 Tampilan Sub-Menu View Billing

G. Tampilan Menu Report

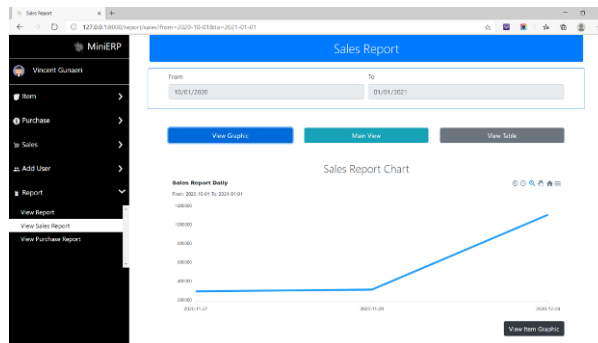
Sub-menu *report* berfungsi untuk menampilkan laporan data penjualan maupun data pembelian yang dilakukan oleh perusahaan dalam kurun waktu tertentu. Pemilihan kurun waktu atau rentang tanggal dapat dilakukan dengan dua buah *datepicker* yang telah disediakan di tampilan halaman, yakni dari tanggal *from* sampai tanggal *to*. Kemudian untuk memunculkan data yang dipilih sesuai dengan rentang tanggal dapat menekan tombol *View Data*.

1) Report Sales

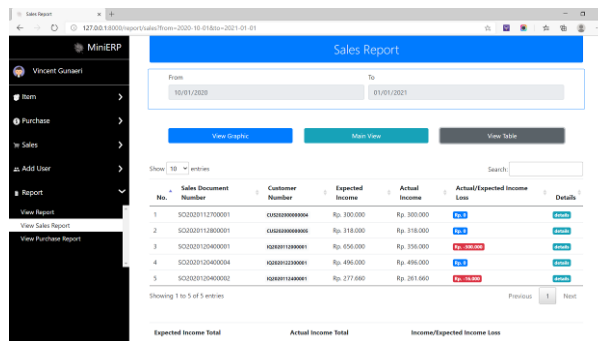
Report Sales adalah tampilan untuk menampilkan laporan penjualan yang dilakukan dalam kurun waktu tertentu (sesuai dengan *input* yang diberika oleh *user*). Terdapat *statistic* yang memperjelas perhitungan dan kalkulasi data penjualan.

2) Report Purchase

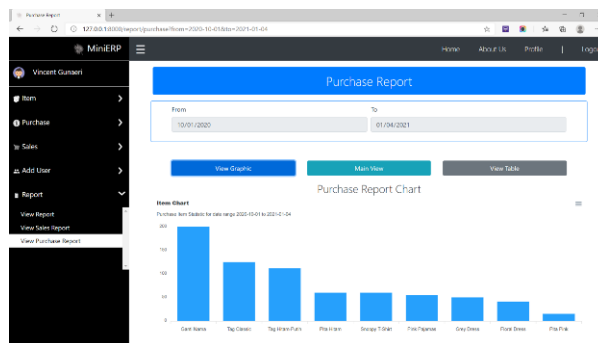
Report Purchase adalah tampilan untuk menampilkan laporan pembelian. Sama seperti *report sales* hanya saja pada tampilan ini data pembelian yang ditampilkan. Fitur grafik dan *statistic* pada *report purchase* sama seperti *report sales*.



Gambar 24 Tampilan Sub-Menu View Report (1)



Gambar 25 Tampilan Sub-Menu View Report (2)



Gambar 26 Tampilan Sub-Menu View Report (3)

V. KESIMPULAN

A. Simpulan

Berdasarkan hasil dari rumusan masalah, tujuan pembahasan, dan hasil pengerjaan aplikasi serta pengamatan serta analisis yang telah dilakukan, kesimpulan yang dapat ditarik adalah sebagai berikut:

- 1) Penggunaan *framework laravel* untuk mengimplementasikan modul yang tercantum di dalam aplikasi *miniERP* telah dimanfaatkan dengan baik. Fitur *Object Relational Mapping* atau *ORM* yang tersedia di dalam *framework laravel* sangat membantu untuk implementasi modul seperti *sales* dan *master data*.
- 2) Penggunaan *framework bootstrap* untuk implementasi tampilan/front-end aplikasi *Plement* sudah cukup dimanfaatkan. Terutama pemanfaatan *framework* pada *filtering* yang diimplementasikan pada tampilan setiap tabel (terdapat fitur *search*, *pagination*, dan *ordering*).
- 3) *Nott Sleepwear* sebagai usaha kecil yang diangkat menjadi studi kasus pengerjaan Tugas Akhir ini mampu untuk mengimplementasikan *Enterprise Resource Planning*. Meskipun usaha tersebut terbilang usaha kecil (tidak terlalu besar) sedangkan *ERP* sendiri merupakan sistem yang complex dan besar. Namun dengan pembuatan aplikasi *Plement* dimana *ERP* diminifikasi (*minified*) atau dikecilkan skalanya sehingga lebih sederhana dan praktis *Nott Sleepwear* dapat mengimplementasikan *website Plement* dan *ERP* untuk diterapkan pada usahanya.

- 4) Alur proses modul *sales* yang digunakan di dalam *Plement* tidak sama dengan yang diterapkan pada modul *sales* yang ada di sistem *ERP* atau *miniERP*. Pada aplikasi *Plement*, modul *sales* tidak memiliki fitur atau proses *quotation* yang dimana pada sistem *ERP* terdapat fitur *quotation* untuk penentuan kesepakatan harga antara pelanggan dengan perusahaan. Di dalam *Plement* data *inquiry* dapat digunakan sebagai referensi pembuatan transaksi *sales*, transaksi *sales* hanya memiliki satu buah pembayaran dan dapat dilakukan banyak (lebih dari satu kali) pengiriman atau *delivery*.

B. Saran

Berdasarkan hasil pengerjaan/pembangunan aplikasi, implementasi, pengujian, dan simpulan pada pengerjaan Tugas Akhir ini. Aplikasi *Plement* masih dapat dikembangkan dan diperluas skalanya dengan menambah fitur yang dapat membantu perkembangan sistem *MiniERP* seperti fitur *delivery tracking*, fitur *linear regression* untuk bagian *report*, dan lain-lain. Selain penambahan fitur, juga dapat dilakukan implementasi tambahan untuk modul ERP seperti *Inventory Management*, *Human Resource*, *Accounting*, dan lain-lain. *Plement* juga berpotensi untuk penambahan fitur *marketplace* sehingga dapat digunakan juga sebagai *online marketplace* atau *e-commerce* sehingga memungkinkan untuk penggunaan yang lebih luas dan bermanfaat.

Semoga dengan adanya aplikasi *Plement* dapat membantu mahasiswa atau pelajar yang ingin mengetahui konsep dasar atau *basic* dari sistem *MiniERP*. Karena di dalam aplikasi *Plement* sudah tercantum modul minimal yang terimplementasi dalam sistem ERP yang *simple*, diharapkan pelajar dapat mengerti dengan lebih mudah mengenai konsep-konsep dasar *MiniERP*.

DAFTAR PUSTAKA

- [1] W. S., "Enterprise Resource Planning (ERP) Solusi Sistem Informasi Terintegrasi", 2005.
- [2] A. Lestari C., "Implementasi Odoo dengan Modul Accounting And Finance di SD Islam Tunas Mandiri", 2017.
- [3] Muchtar, A. Z., Munir S., "Perancangan Web E-Commerce UMKM Restoran Bakso Arema Menggunakan Framework Laravel", 2019.
- [4] Sari, D. P., Wijanarko R., "Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi kasus Di Rumah Kamera Semarang)", 2019.
- [5] Munir S., Adidaya W., Riansyah E., Sasmita H., "Perancangan Sistem Informasi Akademik berbasis Web Menggunakan Framework MVC Pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri", 2016.
- [6] Firman A., Wowor H. F., Najooan X., "Sistem Informasi Perpustakaan Online Berbasis Web", 2016.
- [7] Zakir A., "Rancang Bangun Responsive Web Layout Dengan Menggunakan Bootstrap Framework", 2016.
- [8] M. H., "Analisis Dan Perancangan Sistem Replikasi Database MySQL Dengan Menggunakan VMWARE Pada Sistem Operasi Open Source", 2016.
- [9] M. A. K. M. Pahelvi O., "Sistem Informasi Inventori Barang Menggunakan Metode Object Oriented Di PT. LIVAZA TEKNOLOGI INDONESIA JAKARTA", 2018.
- [10] Perkasa M., L., Setiawan E., B., "Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token", 2018.