

# Perancangan Aplikasi "Plement" bagian Pembelian dengan Studi Kasus NottSleepwear

Anastasia Levina<sup>#1</sup>, Adelia<sup>\*2</sup>

<sup>#</sup>Program Studi SI Teknik Informatika, Universitas Kristen Maranatha  
Jl. Surya Sumantri No. 65 Bandung, 40174

<sup>1</sup>1772020@maranatha.ac.id

<sup>2</sup>adelia@it.maranatha.edu

**Abstract** — Plement stands for Simple Management. Plement is a Mini ERP System Application based website. Plement made with case studies called NottSleepwear where NottSleepwear is a mini business which sell pajamas. There are two main features from Plement which is Sales Feature and Purchase Feature. Purchase Feature divided into three part which is Purchase, Good Receipt, and Invoice. Purchase features are divided based on the existing process in the provisions of an Mini ERP and follow the standar of Mini ERP and made using PHP programming language for back-end and HTML for front-end with the help of Laravel Framework and Bootstrap.

**Keywords**— ERP, Laravel, Purchase, PHP, Plement

## I. PENDAHULUAN

Plement adalah program Mini ERP dimana Plement merupakan singkatan dari Simple Management yang mengartikan program ini merupakan program untuk memanager proses bisnis dari suatu perusahaan secara simple sehingga lebih mudah digunakan. Plement akan dibuat se-efektif mungkin agar pengguna dapat menikmati fitur-fitur yang dibutuhkan secara maksimal dan lebih mudah dipahami untuk orang-orang awam yang tidak mempelajari ERP.

Aplikasi Plement ini dibuat dengan berdasarkan Studi Kasus dari sebuah bisnis baju tidur yang sedang membangun brandnya sendiri. Bisnis baju tidur itu adalah NottSleepwear, saat ini penjualannya dilakukan melalui Instagram dan Shopee. Proses pengambilan barang untuk dijual oleh NottSleepwear adalah NottSleepwear membeli produk-produk baju tidur sesuai yang diinginkan NottSleepwear ke dalam sebuah pabrik dalam jumlah banyak dengan sekali pembayaran. NottSleepwear sudah memiliki vendor untuk masing-masing produk yang dibutuhkan untuk penjualannya dan dalam penjualannya NottSleepwear tidak menerima reseller dan mengirimkan barangnya sesuai dengan prosedur penjualan dengan satu kali pembayaran dan pengiriman.

## II. KAJIAN TEORI

### A. Enterprise Resource Planning (ERP)

*Enterprise Resource Planning* merupakan kepanjangan dari ERP yang merupakan suatu konsep yang mengintegrasikan proses bisnis dan manajemen perusahaan sebagai sudut pandang untuk pengefisienan dan atau pengefektifan sumber daya dalam suatu perusahaan. ERP sendiri mencakup berbagai macam kebutuhan perusahaan dalam berbagai aspek, seperti keuangan, produksi, logistik, dan HRD yang dikemas dalam sebuah *software*. ERP diperuntukan bagi perusahaan barang maupun jasa yang berperan untuk mengintegrasikan proses bisnis yang berhubungan dengan aspek operasi, produksi maupun distribusi di perusahaan atau dapat dikatakan bahwa ERP digunakan untuk mengelola seluruh aktivitas perusahaan termasuk *production, human resource, marketing, finance, supply chain, purchasing, dan logistic*. [1]

ERP sangat dibutuhkan pada saat ini karena semua aktivitas penjualan baik jasa dan logistik semua memerlukan pengelolaan yang baik dan tercatat dengan terstruktur agar proses kerja dapat berjalan lancar dan lebih praktis sehingga dalam pengelolannya perusahaan atau sebuah bisnis dapat terkelola dengan rinci dan menghindari kesalahan pengelolaan dalam penjualan maupun pembeliannya.

### B. Purchasing / Procurement Process

*Procurement Process* merupakan proses yang mengatur sistem pengadaan barang dalam ERP. *Procurement Process* tidak hanya berfokus pada pembelian barang namun *Procurement Process* juga meliputi permintaan (requisition),

meminta penawaran barang atau jasa yang dibutuhkan kepada pemasok sebagai penyedia barang dan jasa, menerima dan menyeleksi surat penawaran dari pemasok, dan menganalisa pemasok agar mendapatkan pemasok yang terbaik dalam pembelian barang. [2] Sebuah aplikasi sistem ERP tidak akan lengkap tanpa purchasing atau procurement proses. Purchasing merupakan salah satu fitur pendukung utama dari sebuah sistem ERP.

### C. Website

Website adalah salah satu aplikasi yang berisikan dokumen dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protocol HTTP (hypertext transfer protocol) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut browser. [3]

### D. Hypertext Preprocessor (PHP)

*Hypertext Preprocessor* (PHP) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis menurut (Nugroho, 2004) PHP banyak dipakai untuk pemrograman situs WEB dinamis. [4] PHP merupakan *server-side scripting*, sintaks dan perintah-perintah dari PHP akan dijalankan dalam server lalu hasilnya dikirim ke browser dalam format HTML. Sehingga kode program yang ditulis dalam PHP tidak akan terlihat oleh *user* yang menggunakan website sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membentuk suatu tampilan berdasarkan permintaan pengguna atau *developer*, seperti menampilkan isi dari basis data ke halaman *website*. Selain itu php juga mendukung akses ke MySQL.

### E. HyperText Markup Language (HTML)

*HyperText Markup Language* (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah Penjelajah web Internet dan *formatting hypertext* sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. [5] HTML seringkali menjadi dasar dari pembuatan suatu website karena tanpa tampilan yang terintegrasi dan rapi pengguna dari sebuah website tetap akan kesulitan menggunakan websitenya meskipun fungsinya sangat bagus. Penggunaan HTML biasanya dimulai dengan penggunaan tag `<>` dan ditutup dengan penggunaan tag `</>`.

### F. Model-View-Controller (MVC)

*Model-View-Controller* (MVC) adalah sebuah konsep yang diperkenalkan oleh penemu Smalltalk (Trygve Reenskaug) untuk meng-enkapsulasi data bersama dengan pemrosesan (*model*), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada sebuah *user interface*. Pola MVC memecahkan sebuah aplikasi menjadi tiga modul asosiasi: *model*, *view*, dan *controller*. Model modul adalah logika bisnis dari aplikasi dan inti dari sebuah aplikasi. *View* adalah *user interface* dari *controller*. Model modul adalah logika bisnis dari aplikasi dan inti dari sebuah aplikasi. *View* adalah *user interface* dari *controller*. Yang merupakan muka umum untuk respon event pengguna. Komponen *controller* mengimplementasi *flow* yang mengontrol antara *view* dan *model*. [6] MVC juga banyak dipakai agar *file* codingan terlihat lebih rapi dan mudah dimengerti sehingga akan lebih memudahkan dalam mengerjakan sesuatu agar mudah mengingat dimana letak dari suatu *code* atau memudahkan untuk bekerja dalam team agar orang lain tidak bingung saat melihat rangkaian struktur sebuah *file* hasil dari sebuah codingan. bingung saat melihat rangkaian struktur sebuah *file* hasil dari sebuah codingan.

### G. Laravel Framework

Laravel adalah sebuah MVC *web development framework* yang didesain untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan serta meningkatkan produktifitas pekerjaan dengan sintak yang bersih dan fungsional yang dapat mengurangi banyak waktu untuk implementasi (Widodo & Purnomo, 2016). Laravel merupakan *framework* dengan versi PHP yang *up-to-date*, karena Laravel mensyaratkan PHP versi 5.3 keatas. Laravel merupakan *framework* PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya (Naista, 2016).[7] Pemakaian Laravel pun sedang tenar dalam pembuatan website-website karena tingkat pemakaiannya yang cukup simple dan metode pembelajaran Laravel sudah sangat banyak tersebar di internet sehingga memudahkan penggunaannya dalam mempelajari *framework* Laravel.

### H. Javascript

JavaScript adalah bahasa pemrograman web yang bersifat *Client Side Programming Language*. *Client Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*. Aplikasi *client* yang dimaksud merujuk kepada *web browser* seperti *Google Chrome*, *Mozilla Firefox*, *Opera Mini* dan sebagainya. JavaScript pertama kali dikembangkan pada pertengahan dekade 90'an. [8] Javascript sangat banyak digunakan dalam pembuatan

*website* khususnya untuk fitur-fitur yang membutuhkan bantuan javascript penambahan jumlah pada *button up down* dalam fitur memasukan jumlah atau pemanggilan nama tabel agar suatu tabel dapat diberikan *filter*.

#### I. *Json*

JSON (*Java Script Object Notation*) adalah format pertukaran data yang bersifat ringan, disusun oleh Douglas Crockford. Fokus JSON adalah pada representasi data di *website*. JSON dirancang untuk memudahkan pertukaran data pada situs dan merupakan perluasan dari fungsi-fungsi javascript. [9] Sebagai contoh *Json* memudahkan saat menggunakan *ajax* yang merupakan salah satu perluasan dari javascript. Data dapat dengan mudah dikonversikan dari *php* ke *ajax* dengan menggunakan bantuan *Json* terlebih dahulu. *Json* juga dapat mudah dikonversi ke bahasa lain seperti C+, C#, Dart, Java, dan lainnya.

#### J. *Jquery*

*jQuery* adalah JavaScript *library* yang dirancang untuk meringkas kode-kode JavaScript, sehingga dapat menyederhanakan penulisan skrip program, sesuai dengan slogan “*write less, do more*”. *jQuery* pertama kali dirilis oleh John Resig pada tahun 2006, pada perkembangannya *jQuery* tidak hanya sebagai *framework* JavaScript. [10]

#### K. *Bootstrap*

*Bootstrap* adalah *front-end framework* yang bagus dan luar biasa yang mengedapankan tampilan untuk *mobiledevice* (Handphone, smartphone dll.) guna mempercepat dan mempermudah pengembangan *website*. *Bootstrap* menyediakan HTML, CSS dan Javascript siap pakai dan mudah untuk dikembangkan. *Bootstrap* merupakan *framework* untuk membangun desain web secara responsif.[11] Penggunaan *Bootstrap* pun cukup mudah rata-rata cara penggunaannya adalah dengan memanggil nama-nama kelas yang sudah ada pada *Bootstrap* dan *source* dari nama-nama kelas dan contoh-contohnya pun sudah lengkap pada *website Bootstrap* sendiri.

#### L. *Cascading Style Sheet (CSS)*

CSS adalah kependekan dari *Cascading Style Sheet*. CSS merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/layout halaman web supaya lebih elegan dan menarik. CSS adalah sebuah teknologi internet yang direkomendasikan oleh *World Wide Web Consortium* atau W3C pada tahun 1996.[12] Umumnya CSS paling sering diaplikasikan dengan HTML dan dalam pembuatannya CSS pun cukup mudah untuk digabungkan dengan HTML dan bila ada kesulitan dalam penggunaannya, cara penggunaannya sudah sangat banyak pada internet.

#### M. *MySQL*

MySQL merupakan *software yang berbasis structure query language (SQL)* tergolong sebagai DBMS (*Database Management System*) yang bersifat *Open Source*. MySQL adalah sebuah implementasi dari sistem manajemen database relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License).[13] Pada penggunaannya MySQL dapat digunakan langsung dengan mendownload XAMPP dan membukanya pada *phpMyAdmin* sehingga MySQL sangat umum digunakan dan dipadukan dengan penggunaan PHP yang sama-sama merupakan bahasa umum dari pembuatan sebuah *website*.

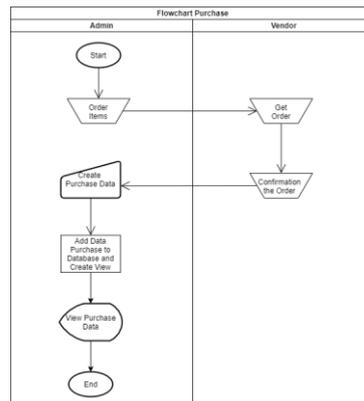
### III. ANALISIS DAN PERANCANGAN SISTEM

#### A. *Flowchart*

Pada Plement salah satu menu utama yang dimiliki adalah Fitur *Purchase* dimana pada pembuatannya diharapkan fitur ini dapat membantu pencatatan pembelian barang dari *vendor* lain agar lebih teratur dalam pencatatannya. Fitur *Purchase* sendiri dibagi menjadi tiga buah bagian dimana terdapat Fitur *Purchase*, *Good Receipt*, dan *Invoice* berikut akan dibahas mengenai proses penggunaan dari Aplikasi Plement.

##### 1) *Flowchart Purchase*

Fitur *Purchase* memiliki dua buah role yaitu role admin dan vendor

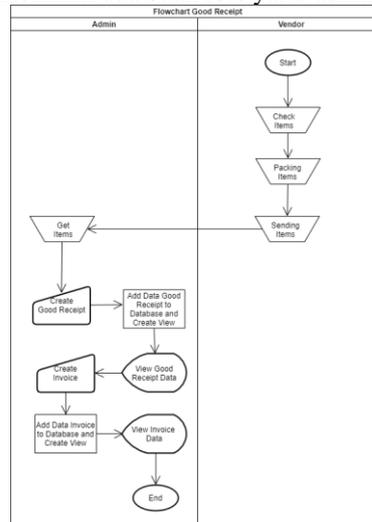


Gambar 1 Flowchart Purchase

Gambar 1 menggambarkan arahan-arahan penggunaan dari Fitur *Purchase*. Pada gambar 1 dijelaskan bahwa setelah admin memesan barang maka *vendor* akan menerima pesannya dan mengkonfirmasi jika pesannya sudah diterima lalu admin akan mulai membuat *create purchase*, sistem akan memasukan data ke dalam *database* dan menampilkan *view* lalu admin bisa melihat datanya pada halaman *view purchase* setelah itu proses pada fitur *purchase* sudah selesai.

## 2) Flowchart Good Receipt

Fitur Good Receipt dan Fitur Invoice memiliki dua buah role yaitu admin dan vendor.



Gambar 2 Flowchart Good Receipt

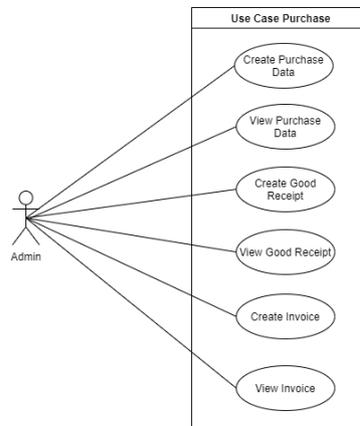
Gambar 2 menggambarkan *vendor* akan mengirimkan barang setelah barang terkirim admin akan menggunakan menu *create good receipt* untuk pembuatan data barang-barang yang sudah terkirim lalu sistem akan memasukan datanya ke dalam *database* dan admin dapat melihat laporannya lagi pada *view good receipt*. *Vendor* akan memeriksa apakah masih ada barang yang harus dikirimkan jika masih ada maka *vendor* akan mem-*packing* barang-barang yang diminta dan mengirimkannya lagi sampai pesannya sesuai, dan jika ada pengiriman barang berikutnya *admin* akan melakukan pembuatan *good receipt* lainnya lagi namun jika tidak ada pengiriman lagi admin akan langsung membuat *invoice*, sistem akan memasukan ke dalam *database* dan menampilkan *view* lalu admin dapat melihat *invoice* pada menu *view invoice*.

## B. Use Case Diagram

Dalam Aplikasi Plement ini user yang menggunakan Aplikasi ini hanya terdapat satu *role* yaitu *role* admin. Dimana admin sendiri adalah orang yang memiliki usaha atau bisnis yang menggunakan Aplikasi Plement atau pekerja dalam usaha atau bisnis tersebut. Berikut adalah gambar *Use Case Diagram* dari fitur-fitur dalam Aplikasi Plement.

### 1) Use Case Purchase

Dalam fitur Purchase terdapat beberapa fitur yang melengkapi fitur utamanya. Fitur-fitur tersebut adalah Create Purchase, Create Good Receipt, Create Invoice, View Purchase, View Good Receipt, dan View Invoice. Berikut merupakan gambar dari Use Case Purchase.

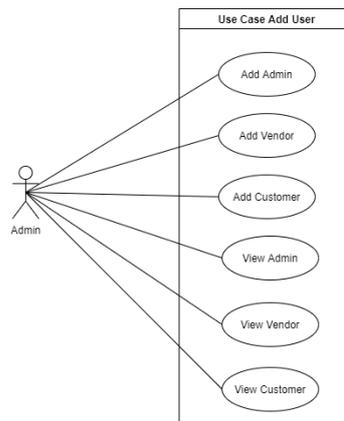


Gambar 3 Use Case Purchase

Pada gambar 3 digambarkan apa saja yang dapat dilakukan admin dalam fitur *Purchase*. Dalam setiap fitur di dalam *Purchase* setiap data yang dimasukkan melalui *create* dapat dilihat pada *menu view*.

### 2) Use Case Add User

Dalam fitur *Add User* terdapat beberapa fitur yang melengkapi fitur utamanya yaitu adalah *Add Admin*, *Add Vendor*, dan *Add Customer* dimana data *vendor* dan *customer* akan dibutuhkan untuk mencatat pembelian dan penjualan.

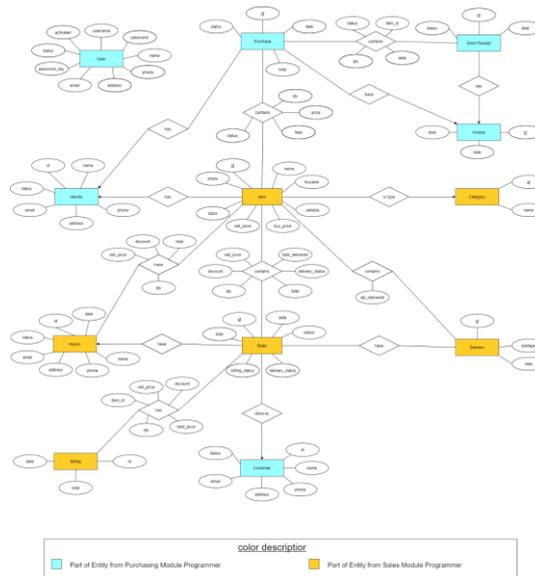


Gambar 4 Use Case Add User

Pada gambar 4 dijelaskan bahwa pengguna Aplikasi Plement dapat memasukkan data admin baru, *vendor*, dan *customer* dan dapat melihat data-datanya dalam halaman *view*.

### C. Database Design

Pada Aplikasi Plement *database* yang akan digunakan adalah *database* dari MySQL dengan menggunakan phpMyAdmin. *Database* ini sudah disusun sesuai dengan rancangan untuk UInya dan dilengkapi dengan beberapa tabel *log* agar data yang dihasilkan lebih akurat. *Database* ini juga dirancang dengan bentuk Diagram Chen. Berikut merupakan gambaran *Database* yang sudah disusun.



Gambar 5 Database Design

Gambar 5 merupakan gambar dari Rancangan *Database* Plement, dimana pada *database* ini diberikan rancangan keseluruhan dari Aplikasi Plement, baik untuk bagian *purchasing* maupun *sales and distribution*. Tidak lupa juga pada *database* ini berisi tabel *user* seperti *vendor*, *customer*, dan *admin* yang dimana tabel *user* merupakan tabel untuk *admin*. Pada gambar 5 pun terdapat tabel *item* dimana tabel *item* merupakan sebuah tabel yang hampir memiliki hubungan dengan semua tabel lainnya.

#### IV. IMPLEMENTASI

##### A. Database

Aplikasi Plement menggunakan *database* MySQL yang dirancang dalam phpMyAdmin. Namun dalam perancangannya *database* dari Aplikasi Plement dibuat dalam kode-kode program melalui bantuan Framework Laravel.



Gambar 6 Database

Tabel-tabel entitas *database* pada gambar 6 tidak memiliki *relationship*. Hal ini dikarenakan dengan menggunakan Framework Laravel pengintegrasian dapat menggunakan *code* langsung di dalam program dan tidak perlu direlasikan melalui *database* aslinya sehingga menjadi lebih efektif dan lebih mudah untuk digunakan. Hal seperti ini dinamakan dengan *database* menggunakan *migration* dalam Laravel penggunaannya dimaksudkan juga agar programmer yang bekerja sama membuat program dapat dengan mudah mengintegrasikan databasenya masing-masing tanpa perlu mengirim file *database* terus menerus dan menyesuaikan secara manual.

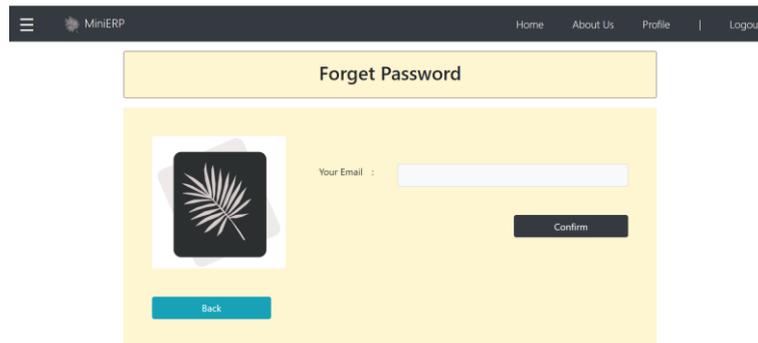
Terdapat 20 entitas dalam *database* Aplikasi Plement dimana di dalamnya sudah terintegrasi pula *database* dari fitur *Sales* dan *Purchase* dan terdapat tabel migration yang dibuat langsung oleh Laravel dan terdapat *field created\_at* dan *updated\_at* yang merupakan *field* wajib dan otomatis dari Laravel. Terdapat pula sebuah tabel *log* pada tabel *invoice* yang digunakan untuk membantu programmer dalam menampilkan data.

## B. Implementasi Aplikasi Plement

Aplikasi Plement tentunya memiliki tampilan yang sudah dibuat senyaman mungkin untuk penggunanya. Selain itu dalam pembuatan Aplikasi Plement juga tentunya tidak luput dari fungsi-fungsi yang ada di belakang tampilannya.

### 1) Forgot Password

Hampir seluruh aplikasi tentunya memiliki fitur *forgot password* dimana fitur ini sangat dibutuhkan ketika pengguna dari aplikasi lupa akan *password*-nya atau ingin mengganti *password*-nya karena satu san lain hal.

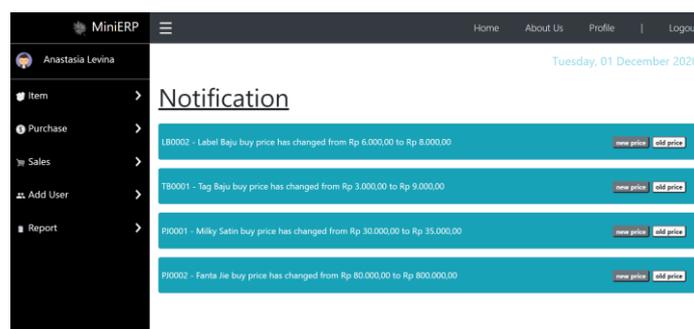


Gambar 7 Forgot Password

Dengan menekan tombol biru bertuliskan “*Forgot Password?*” admin akan di arahkan ke halaman *forgot password*. Admin harus memasukan *email* yang digunakan untuk mendaftar dalam *website* ini setelah itu fungsi dari *forgot password* akan mengecek apakah *email* yang dimasukan sudah aktif dan terdaftar bila belum admin harus mengaktifkan *email*-nya dahulu melalui *email* yang sudah dikirim via *email* untuk mengganti *password* pertama kali dan mengaktifkan *email*-nya. Bila email tidak ada fungsi akan menolak email dan menampilkan tulisan bahwa *email* tidak ada, namun bila berhasil fungsi akan mengirimkan *email* ke *email* yang bersangkutan untuk mengganti *password* admin.

### 2) Home

Halaman *home* dalam aplikasi plement berisikan notifikasi-notifikasi pengingat bahwa harga dari barang yang ada dalam *list item* sedang naik atau turun. Admin dapat memilih apakah harga dalam *list item* mau diubah atau tetap sama dengan harga lama.

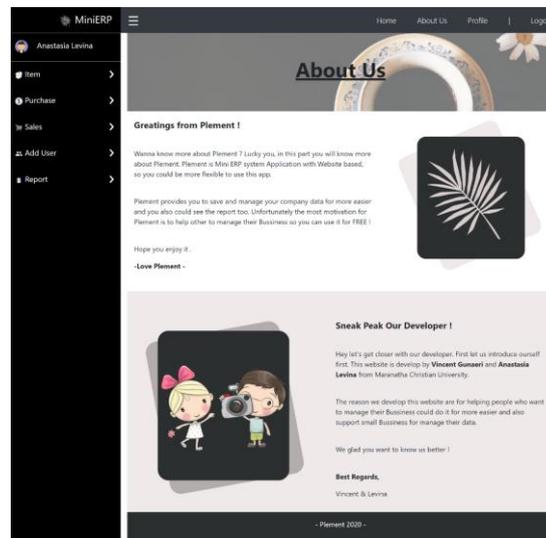


Gambar 8 Home

Pada gambar 8 digambarkan bahwa terdapat tombol *new price* dan *old price* dimana kegunaan untuk admin, tombol *new price* dan *old price* merupakan pilihan untuk admin apakah harga barang mau disesuaikan dengan update terbaru atau tidak.

### 3) About Us

Dalam halaman *about us* dijelaskan banyak hal mengenai Aplikasi Plement dan mengenai *programmer* yang telah membuat Aplikasi Plement dengan tujuan dari pembuatan aplikasi.

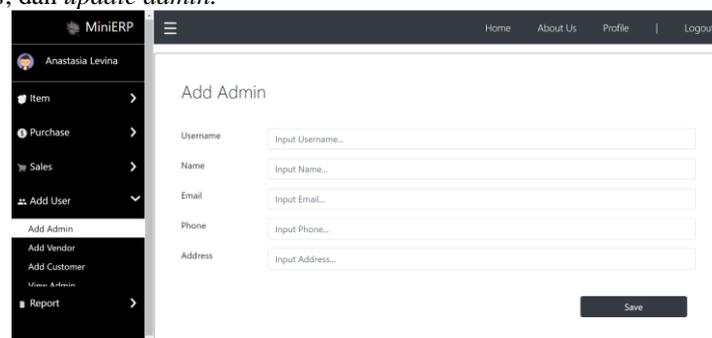


Gambar 9 About Us

Sesuai dengan gambar 9 pada halaman *about us* selain dijelaskan mengenai Aplikasi Plement ada juga penjelasan mengenai *programmer* yang membuat Aplikasi Plement dan tujuan dari pembuatan Aplikasi Plement. Disertakan juga logo dari Aplikasi Plement sebuah logo daun yang kompleks yang mengartikan bahwa Aplikasi Plement merupakan sebuah aplikasi yang kompleks namun tetap satu meski memiliki banyak cabang fitur.

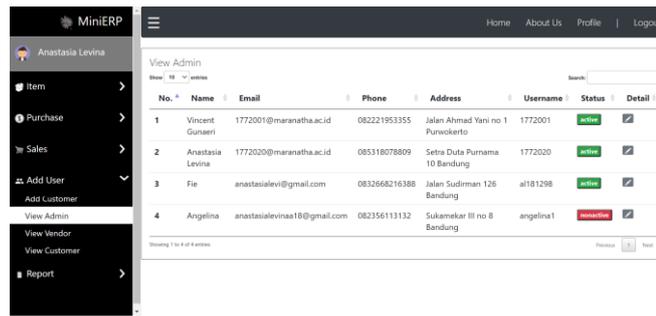
### 4) Master Data Admin

Pada *master data* admin terdapat beberapa buah fitur pelengkap di dalamnya yaitu *add admin*, *view admin*, *view detail admin*, *change status*, dan *update admin*.



Gambar 10 Add Admin

Pada gambar 10 digambarkan bahwa admin dapat menambah admin dengan cara memasukkan data admin baru mulai dari *username*, *name*, *email*, *phone*, *address* dan mereka dapat menekan tombol *save* untuk menambah admin baru. Selain itu program akan mengirimkan *email* kepada admin baru untuk mengaktifkan akunnya dan untuk mengubah *password*-nya.

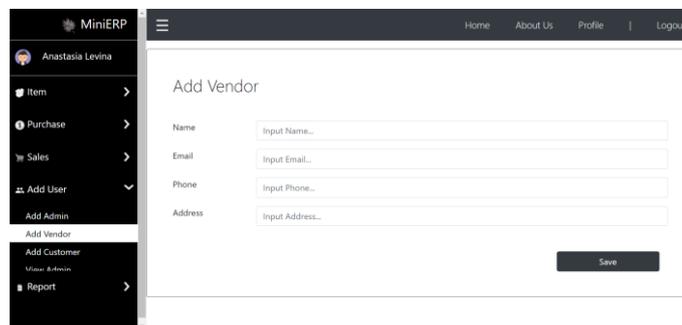


Gambar 11 View Admin

Pada gambar 11 digambarkan bahwa admin dapat melihat data admin lainnya yang dapat menggunakan Aplikasi Plement dan admin pula dapat mengubah status admin yang sudah tidak aktif menjadi *nonactive* sehingga admin yang *nonactive* tidak dapat memasuki Aplikasi Plement. Admin dapat melihat detail dari admin lainnya dan meng-update data admin pada tombol yang bergambarkan pena.

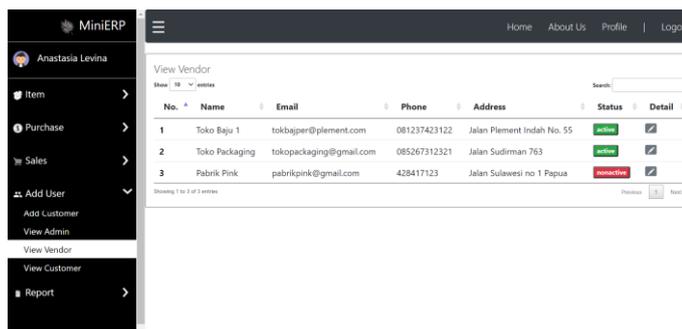
### 5) Master Data Vendor

Pada *master data vendor* terdapat beberapa buah fitur pelengkap di dalamnya yaitu *add vendor*, *view vendor*, *view detail vendor*, *change status*, dan *update vendor*.



Gambar 12 Add Vendor

Pada gambar 12 digambarkan bahwa admin dapat menambah *vendor* dengan cara memasukkan data *vendor* baru mulai dari *name*, *email*, *phone*, *address* dan mereka dapat menekan tombol *save* untuk menambah *vendor* baru.

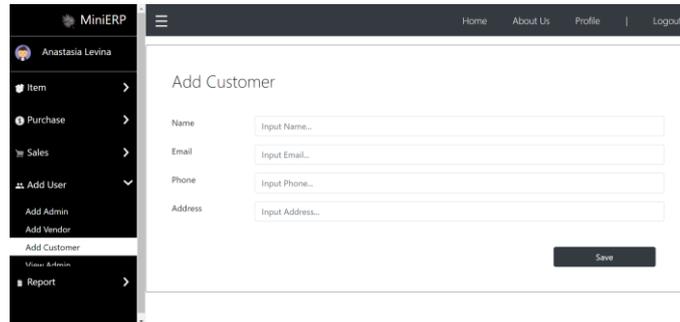


Gambar 13 View Vendor

Pada gambar 13 digambarkan bahwa admin dapat melihat data *vendor* yang dapat ada dalam Aplikasi Plement dan admin pula dapat mengubah status admin yang sudah tidak aktif menjadi *nonactive* sehingga *vendor* yang *nonactive* tidak dapat digunakan dalam Aplikasi Plement. Admin dapat melihat detail dari setiap *vendor* dan meng-update data *vendor* pada tombol yang bergambarkan pena.

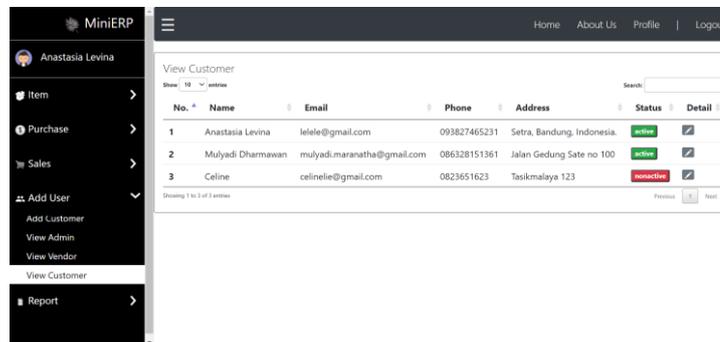
### 6) Master Data Customer

Pada *master data customer* terdapat beberapa buah fitur pelengkap di dalamnya yaitu *add customer*, *view customer*, *view detail customer*, *change status*, dan *update customer*.



Gambar 14 Add Customer

Pada gambar 14 digambarkan bahwa admin dapat menambah *customer* dengan cara memasukkan data *customer* baru mulai dari *name*, *email*, *phone*, *address* dan mereka dapat menekan tombol *save* untuk menambah *customer* baru.

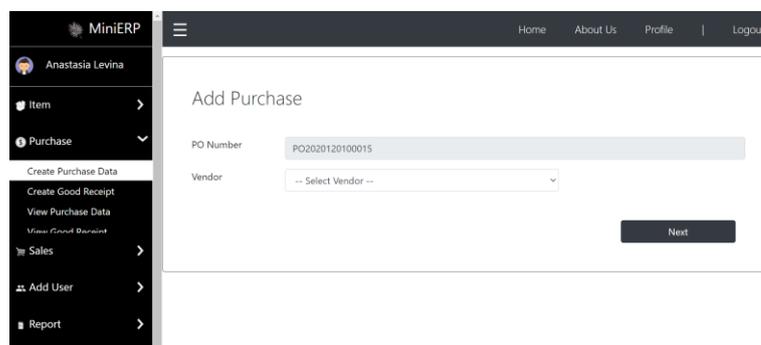


Gambar 15 View Customer

Pada gambar 15 digambarkan bahwa admin dapat melihat data *customer* yang dapat ada dalam Aplikasi Plement dan admin pula dapat mengubah status admin yang sudah tidak aktif menjadi *nonactive* sehingga *customer* yang *nonactive* tidak dapat digunakan dalam Aplikasi Plement. Admin dapat melihat detail dari setiap *customer* dan meng-update data *customer* pada tombol yang bergambarkan pena.

### 7) Purchase Data

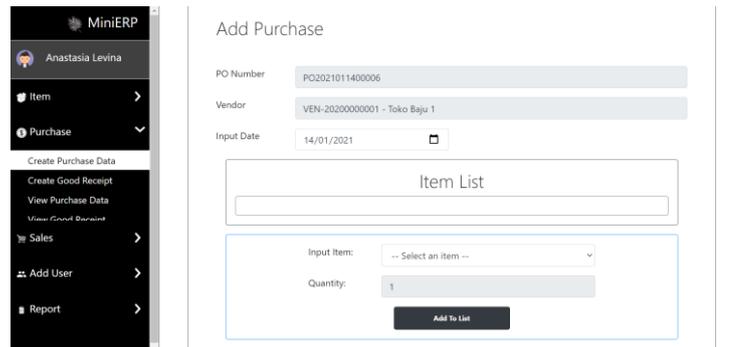
Dalam *Purchase Data* terdapat beberapa bagian dari fitur-fitur yang mendukung *Purchase Data*. Beberapa fitur tersebut adalah *Create Purchase Data*, *View Purchase Data*, *Detail Purchase Data*, dan *Edit Purchase Data*.



Gambar 16 Create Purchase

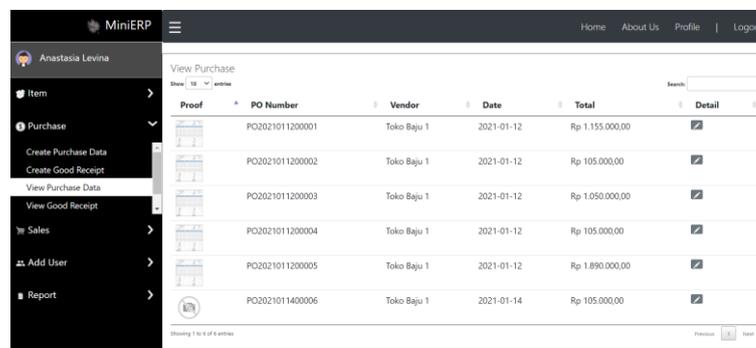
Pada gambar 16 menggambarkan tampilan awal dari *create purchase data* dimana saat awal akan membuat *purchase data* *PO Number* akan secara otomatis di *generate* oleh sistem dan admin hanya tinggal memilih *vendor* lalu

admin dapat menekan tombol *next* untuk ke halaman berikutnya untuk memasukan data-data dari item yang akan dibeli dan menyimpan data pada *database*.



Gambar 17 Save Create Purchase

Pada gambar 17 merupakan gambaran *page* setelah admin memilih *vendor* pada gambar 4.19 setelah itu admin harus memilih tanggal dan memasukan *item*, *price*, dan *quantity* dari barang yang ingin dibeli. *Combo box* pada *item* sudah menampilkan data yang sesuai dengan nama *vendor* yang dipilih di awal oleh admin.

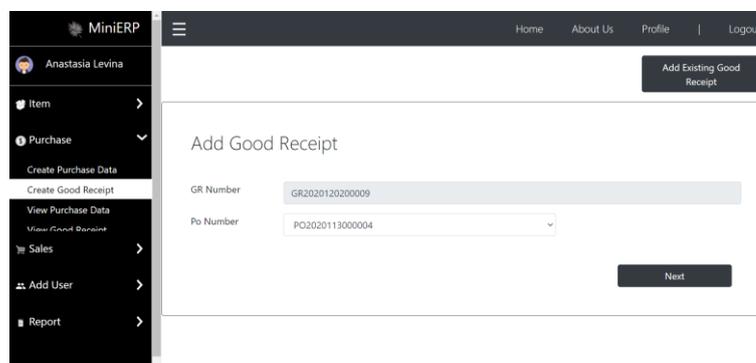


Gambar 18 View Purchase

Pada gambar 18 merupakan gambaran tampilan dari *view purchase data* dimana dalam *viewnya* akan menampilkan *PO number*, *vendor*, *date*, *total*, dan sebuah tombol untuk menuju ke halaman *detail* dari setiap *PO number*. Tampilan menggunakan *datatables* dimana dalam tampilan ini admin dapat memilih berapa banyak data yang ingin dilihat dan sudah terdapat *pagination* pada halamannya juga sebuah *input box* untuk *search* atau mencari data yang diinginkan.

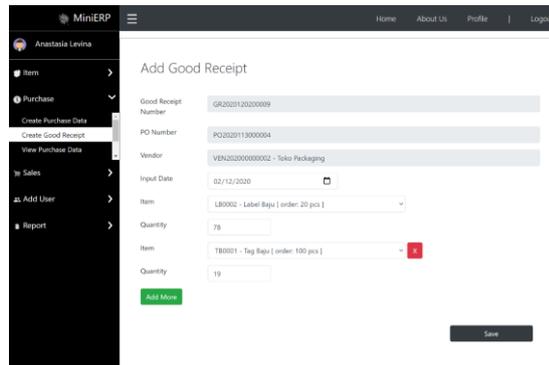
#### 8) Good Receipt

Dalam *Good Receipt* terdapat beberapa fitur pelengkap fitur *good receipt*. Beberapa dari fitur tersebut adalah *create good receipt*, *add existing good receipt*, *save add good receipt*, *view good receipt*, *detail good receipt*, *add to stock*, *delete good receipt item*, dan *add data good receipt*.



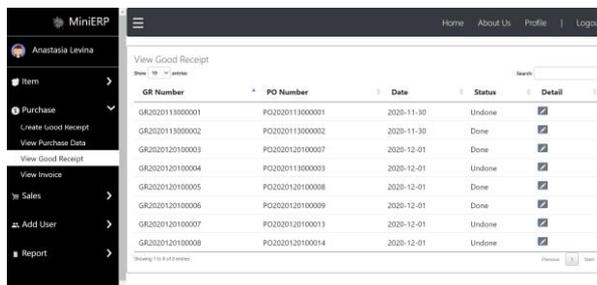
Gambar 19 Create Good Receipt

Pada gambar 19 admin dapat membuat *Good Receipt* yang baru dengan nomor yang sudah otomatis di *generate* oleh program lalu admin dapat memilih *PO Number* yang ingin dibuat *Good Receipt*nya setelah itu admin dapat menekan tombol *next* untuk ke halaman *save create good receipt*.



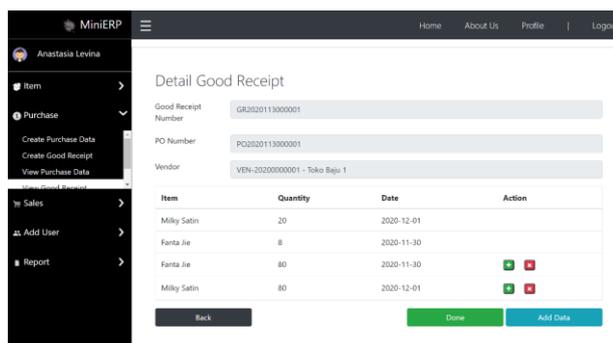
Gambar 20 Save Create Good Receipt

Pada gambar 20 admin dapat memasukkan data tanggal dan *item* yang diinginkan lalu admin dapat menambah item dengan menekan *add more* sesuai dengan berapa banyak jenis barang yang diinginkan. Admin juga harus melengkapi *quantity* yang sudah dikirimkan lalu menekan tombol *save* agar program dapat memasukkan data ke dalam *database*.



Gambar 21 View Good Receipt

Pada gambar 21 merupakan tampilan dari *good receipt* dan pada halaman ini tentunya admin dapat memilih berapa jumlah data yang ingin dilihat dan bisa mencari data yang diinginkan. Halaman ini juga sudah dilengkapi dengan pagination dan tombol pena untuk melihat *detail* dari *good receipt*nya.

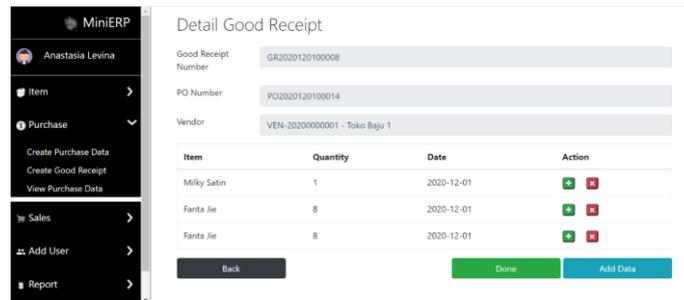


Gambar 22 Detail Good Receipt

Pada gambar 22 merupakan tampilan jika admin memasuki halaman *detail good receipt* dimana pada halaman ini admin dapat melihat *detail* dari *good receipt*nya, *item* yang sudah terkirim. Admin juga dapat memasukkan barang yang sudah dikirim ke dalam *stock* pada *item* dimana bila admin menekan tombol plus maka barang akan masuk ke dalam *stock* di *item* dan barang tidak dapat di *delete*. Bila admin menekan tombol *done* maka *good receipt* akan selesai dan *invoice* akan dibuat.

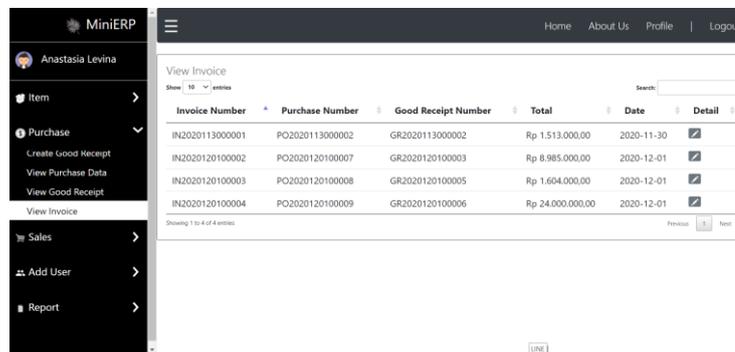
9) Invoice

Dalam *invoice* tampilan *invoice* dibuat lebih simple dengan menggunakan satu *button* saja dimana pada *button* tersebut akan otomatis membuat *invoiceny* ketika di tekan. Pada *Invoice* fitur-fitur yang ada adalah sebagai berikut: *create invoice*, *view invoice*, dan *view invoice detail*.



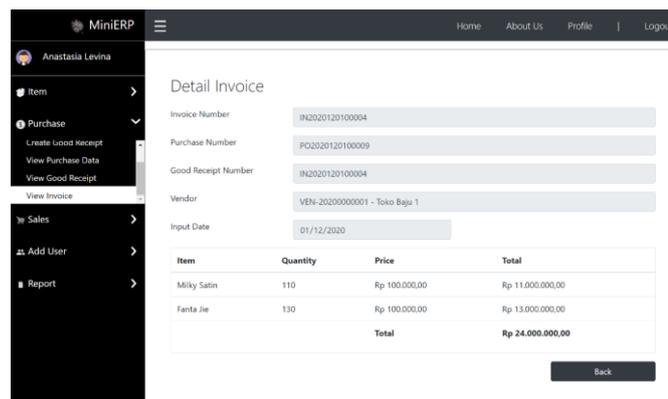
Gambar 23 Create Invoice

Pada gambar 23 diperlihatkan tampilan dari *detail good receipt* dimana terdapat tombol *done* di sebelah *add data*. Bila tombol *done* dipencet maka data tidak dapat diubah kembali kecuali menambahkan data ke dalam *item*.



Gambar 24 View Invoice

Pada gambar 24 merupakan tampilan dari *view invoice* dimana dalam *view invoice* admin dapat melihat detailnya lagi dari pembelian yang sudah dilakukan. Pada *view invoiceny* sendiri admin dapat melihat data sesuai dengan banyak data yang dipilih admin dan bisa mencari data yang diinginkan.



Gambar 25 Detail Invoice

Pada gambar 25 merupakan gambar dari *detail invoice* yang ada. Pada *Detail invoice* terdapat barang, *quantity*, *price*, *subtotal*, dan *total* dari semua *item* yang sudah datang. Terdapat pula tanggal *good receipt*, *vendor*, dan *detail* lainnya.

## V. KESIMPULAN

### A. Simpulan

Berdasarkan hasil dari rumusan masalah yang diangkat dalam bab pertama dan berdasarkan pengamatan dalam proses pembuatan aplikasi, ada beberapa hal yang dapat disimpulkan, antara lain:

- 1) Fitur *Purchase* dalam sebuah aplikasi Sistem ERP merupakan sebuah fitur utama yang mendukung jalannya proses utama dalam proses perdagangan. Bila sebuah Aplikasi Sistem ERP memiliki alur proses pembelian yang salah maka Sistem ERP dari Aplikasi akan menjadi tidak beraturan.
- 2) Aplikasi Plement menggunakan struktur database yang dibantu dengan Eloquent ORM Laravel dimana dalam sistem databasenya programmer dapat mengambil dan menyimpan data dalam setiap entitasnya dan data yang langsung dari kode program sehingga membantu proses pengerjaan sebuah program oleh dua *programmer* atau lebih.
- 3) Pembuatan fitur Sistem Mini ERP sangat kompleks dengan alur yang dapat berbeda-beda sesuai dengan kebutuhan dan kasus dari bisnisnya. Namun dalam Studi Kasus *Nott Sleepwear* sendiri sistem pembelian dibuat agar admin dapat memasukan data *purchase*, pengiriman yang dapat berulang kali sampai barang terkirim sesuai dengan jumlahnya, dan setelah selesai admin akan membayar barang sesuai dengan *invoice*-nya.

### B. Saran

Berdasarkan hasil penelitian yang telah dilakukan dan aplikasi yang telah dikerjakan, terdapat beberapa saran untuk pengembangan selanjutnya, yaitu:

- 1) Pengembangan Aplikasi Plement terutama dalam fitur *purchasing* masih dapat lebih dikembangkan dengan menambahkan lokasi penyimpanan barang.
- 2) Pengembangan Aplikasi Plement fitur *purchasing* bagian *invoice* masih dapat dikembangkan, dimana dapat ditambahkan *detail* seperti *tax amount*.

## DAFTAR PUSTAKA

- [1] P. M. Jannah, L. W. Suwarsono dan P. F. A, "Perancangan Enterprise Resource Planning Modul Benefits Aplikasi SAP dengan Metode SAP Activate Pada PT. XYZ," 2018.
- [2] M. N. Rahmadano, R. Sarno dan S. , "Rancang Bangun Sistem Enterprise Resource Planning pada Modul Procurement Process (Purchasing) Berorientasikan Multi-Tenancy dengan Sistem Basis Data Terdistribusi," vol. 5, 2016.
- [3] R. Erinton, R. M. Negara dan D. D. Sanjoyo, "ANALISIS PERFORMASI FRAMEWORK CODEIGNITER DAN LARAVEL MENGGUNAKAN WEB SERVER APACHE," 2017.
- [4] D. Lavarino dan W. Yustanti, "RANCANG BANGUN E – VOTING BERBASIS WEBSITE DI UNIVERSITAS NEGERI SURABAYA," 2016.
- [5] H. dan A. Syarif, "SISTEM INFORMASI GEOGRAFIS SARANA PADA KABUPATEN PASAMAN BARAT," 2016.
- [6] P. Simanjuntak dan A. Kasnady, "ANALISIS MODEL VIEW CONTROLLER (MVC) PADA BAHASA PHP," 2016.
- [7] F. Luthfi, "Penggunaan Framework Laravel Dalam Rancang Bangun Modul Back-End Artikel Website Bisnisbisnis.ID," 2017.
- [8] . O. Pahlevi, A. Mulyani dan M. Khoir, "SISTEM INFORMASI INVENTORI BARANG MENGGUNAKAN METODE OBJECT ORIENTED DI PT. LIVAZA TEKNOLOGI INDONESIA JAKARTA," 2018.
- [9] B. D. Wijaya, F. E. A dan A. Fiade, "IMPLEMENTASI JSON PARSING PADA APLIKASI MOBILE E-COMMERCE," 2015.
- [10] I. Y. B, "APLIKASI PENGOLAHAN CITRA BERBASIS WEB MENGGUNAKAN JAVASCRIPT DAN JQUERY," 2014.
- [11] F. Effendy dan B. Nuqoba, "PENERAPAN FRAMEWORK BOOTSRAP DALAM PEMBANGUNANSISTEM INFORMASI PENGANGKATAN DAN PENJADWALANPEGAWAI (STUDI KASUS:RUMAH SAKIT BERSALIN BUAH DELIMA SIDOARJO)," 2016.
- [12] A. Noor, "Aplikasi Kisah 25 Nabi Dan Rasul Berbasis Android," 2016.
- [13] A. Josi, "PENERAPAN METODE PROTOTIPING DALAM PEMBANGUNAN WEBSITE DESA," 2017.