
**JURAGAN (JUALAN RAKYAT GABUNGAN
ONLINE) - PANDUAN PEMBUATAN DAN
MENGUNAKAN APLIKASI**

YUSNIAR NUR SYARIF SIDIQ

1.16.4.089



**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA
POLITEKNIK POS INDONESIA
BANDUNG
2020**

BAB I

PENDAHULUAN

Pada abad ke 21 dimana para pembisnis usaha kecil menengah atau yang biasa kita sebut dengan UKM telah menyadari mengenai penerapan sistem *E-Commerce* dalam meningkatkan daya saing bisnis usaha mereka [1]. *E-Commerce* merupakan salah satu sebuah sistem bisnis yang sangat populer untuk saat ini, dimana pada umumnya mengacu pada komunikasi bisnis dan transaksi, menjual, serta membulu produk melalui media internet (*online*) [2]. Dengan adanya penerapan *E-Commerce* kedalam dunia binis para pengusaha akan lebih terbantu dan mudah dalam melayani konsumen sehingga menimbulkan bisnis tersebut dapat mempertahankan para *customer*-nya [3]. Disamping itu efek dari penggunaan sistem *e-commerce* ini dapat menimbulkan hubungan atau kerjasama antara mitra bisnis dan pemasok [3]. Negara Indonesia merupakan salah satu negara yang sebagian besar masyarakatnya telah menggeluti usaha di bidang UKM. Hal ini menunjukkan dimana masyarakat negara Indonesia memiliki tingkat kreatifitas yang cukup tinggi. Terdapat 3 sektor kreatif di negara Indonesia yang pertama adalah kuliner dimana berupa usaha yang mengeluarkan produk berupa makan dan minuman. Adapun kota – kota yang dijadikan destinasi kuliner di Indonesia yaitu Kota Bandung, Jakarta, Bali, Yogyakarta, Solo, Semarang, dan masih banyak lagi. Sektor yang kedua adalah *fashion* dimana merupakan sebuah usaha yang mengeluarkan produk berupa pakaian, sepatu, assesoris dan sesuatu yang dapat menunjang gaya hidup. Sektor terakhir yaitu kerajinan dimana merupakan sebuah usaha yang menghasilkan produk – produk kreatif dari buatan

tangan atau keterampilan tangan sehingga menciptakan sebuah seni atau karya.

Tidak heran apabila saat ini semua kegiatan bisnis sudah memasuki jejaringan sosial. Banyak sekali para pengusaha yang memanfaatkan media sosial untuk melakukan promosi produknya. Hal ini dikarenakan negara Indonesia sudah memasuki RI (Revolusi Industri) 4.0. Di era RI 4.0 ini dimana sebuah pekerjaan sudah dilakukan secara modern seperti halnya sistem *e-commerce* yang membantu kegiatan bisnis. Namun pada kenyataannya penerapan sistem *e-commerce* ini masih jarang sekali digunakan oleh negara – negara berkembang [4]. Padahal studi telah mengatakan penerapan sistem *e-commerce* ini tidak hanya dapat diterapkan oleh industri – industri yang memiliki tingkatan tinggi, namun pada kenyataannya industri dengan tingkatan rendah pun dapat menerapkannya secara optimal [5]. Dengan penerapan sistem *e-commerce* tersebut terbukti dimana industri tersebut secara perlahan akan tumbuh [5]. Dengan adanya penerapan sistem *e-commerce* tersebut dimana dapat meningkatkan produktivitas tenaga kerja signifikan terkait positif [5]. Sistem *e-commerce* juga dapat meningkatkan daya saing bagi pengusaha yang menerapkannya [6].

Berdasarkan teori – teori tersebut dimana penulis akan menciptakan sebuah sistem *e-commerce* yang mampu menyatukan para pengusaha UMKM, UKM, dan PKL. *E-Commerce* tersebut akan diberi nama JURAGAN yang dimana merupakan arti dari “Jualan Rakyat Gabungan *Online*”. JURAGAN dirancang untuk membantu para pengusaha – pengusaha bersaing di era RI 4.0 ini. JURAGAN akan menggabungkan para pengusaha dengan status UMKN, UKM, hingga PKL kedalam satu

wadah dengan sistem *e-commerce* dimana para pengusaha tersebut dapat mempromosikan produknya melalui media internet atau secara *online*. JURAGAN juga berfungsi sebagai media komunikasi dan transaksi melalui media internet (*online*). JURAGAN tersebut merupakan sistem *e-commerce* dengan bentuk *website* dengan berbasis komunitas yang dimana dirancang dengan menggunakan *framework codeigniter* dengan bahasa pemrograman PHP dan *MariaDB* sebagai basis datanya.

BAB II

LANDASAN TEORI

2.1 Usaha Kecil Menengah (UKM)

Usaha Kecil Menengah (UKM) merupakan salah satu kegiatan bisnis atau usaha yang didirikan berdasarkan dari inisiatif sendiri [7]. UKM merupakan sekelompok usaha paling banyak dan terbesar di negara Indonesia [7]. Banyaknya UKM yang berdiri di negara Indonesia ini dikarenakan produk – produk yang dimilikinya sangat diminati oleh masyarakat [7]. Salah satu UKM yang sangat diminati oleh masyarakat Indonesia yaitu salah satu sektor di bidang kuliner. Hal ini dikarenakan makanan merupakan salah satu kebutuhan pokok bagi kelangsungan hidup makhluk hidup, sehingga berpeluang besar dalam pengambilan keuntungan. Sebelum melanjut ke pembahasan berikutnya, apakah kalian tahu apa perbedaan UKM dan UMKM ?. Berikut akan dijelaskan apa itu perbedaan UKM dan UMKM menurut UU No 28 Tahun 2008.

2.1.1 Usaha Mikro

Dimana suatu perusahaan dikategorikan menjadi usaha mikro apabila memiliki kriteria sebagai berikut :

1. Memiliki kekayaan bersih sebesar Rp. 50.000.000 tidak termasuk bangunan tempat usaha dan tanah.
2. Memiliki penghasilan tahunan sebesar Rp. 300.000.000.

2.1.2 Usaha Kecil

Dimana suatu perusahaan dikategorikan menjadi usaha kecil apabila memiliki kriteria sebagai berikut :

1. Memiliki kekayaan bersih dimana lebih dari Rp. 50.000.000 dan paling banyak Rp.500.000.000 bukan termasuk tanah dan bangunan usaha.
2. Memiliki penghasilan tahunan dimana lebih dari Rp. 300.000.000 dan paling banyak sebesar Rp. 2.500.000.000.

2.1.3 Usaha Menengah

Dimana suatu perusahaan dikategorikan menjadi usaha menengah apabila memiliki kriteria sebagai berikut :

1. Memiliki kekayaan bersih dimana lebih dari Rp. 500.000.000 dan paling banyak sebesar Rp. 10.000.000.000 bukan termasuk tanah dan bangunan usaha.
2. Memiliki penghasilan tahunan dimana lebih dari Rp. 300.000.000 dan paling banyak sebesar Rp. 50.000.000.000.

2.2 Pedagang Kaki Lima (PKL)

Pedagang Kaki Lima (PKL) dapat diartikan dimana seseorang yang melakukan usaha dengan menggunakan gerobak. Istilah ini muncul dari persepsi masyarakat yang dimana dua kaki milik pengusaha dan ditambah tiga kaki dari roda gerobak pengusaha sehingga disebut dengan pedagang kaki lima.

2.3 Promosi

Promosi merupakan salah satu kegiatan dalam proses bisnis yang dimana dengan tujuan memperkenalkan suatu produk dan jasa kepada konsumen [7]. Promosi dapat diartikan juga sebagai suatu komunikasi informasi antara penjual dengan pembeli yang bertujuan untuk

memberikan sebuah pengenalan produk sehingga membuat pelanggan yang awalnya tidak tahu menjadi tahu dan menarik minat untuk membeli produk tersebut.

2.4 Internet

Internet merupakan sistem informasi global yang terhubung secara logika oleh *address* yang unik secara global dan berbasis pada *internet protocol* (IP), mendukung komunikasi dengan menggunakan TCP/IP sehingga membuatnya dapat diakses baik secara umum atau khusus [8].

2.5 E-Commerce

E-Commerce didefinisikan sebagai proses pembelian, penjualan, mentransfer atau bertukar produk, jasa atau informasi dengan menggunakan jaringan komputer [9]. Dengan menggunakan bentuk-bentuk cara tradisional dari proses bisnis dan memanfaatkan jejaring sosial melalui internet, strategi bisnis dapat berhasil apabila dilakukan dengan benar, yang akhirnya dapat menghasilkan peningkatan *customer* [9]. Dengan adanya *e-commerce* tersebut dimana proses bisnis yang dilakukan akan semakin membaik dan dapat meningkatkan daya saing antar sesama industri [9]. *E-Commerce* merupakan sebuah bisnis yang populer untuk saat ini, dimana umumnya mengacu pada komunikasi bisnis dan transaksi melalui internet, menjual, dan membeli produk secara online [2].

2.6 Komunitas

Komunitas pada umumnya diartikan sebagai sebuah perkumpulan sosial dari beberapa organisme yang dimana saling berbagi lingkungan dan umumnya memiliki ketertarikan serta habitat yang sama [10]. Komunitas

juga dapat diartikan sebagai identifikasi serta interaksi sosial yang dibentuk dengan berbagai dimensi kebutuhan fungsional.

2.7 Aplikasi

Aplikasi adalah penggunaan atau penerapan dalam suatu konsep yang menjadi sebuah pokok pembahasan, dimana aplikasi juga bisa diartikan sebagai program komputer yang diciptakan dengan tujuan membantu dan mempermudah kegiatan manusia untuk melaksanakan sebuah tugas tertentu [11].

2.8 Framework

Framework berfungsi dalam memfasilitasi pemrograman web dan membuatnya menjadi lebih teratur [12]. Dimana *framework* akan meningkatkan produktivitas pemrograman karena menuliskan sepotong *source code* yang biasanya bersifat panjang dan membutuhkan waktu yang cukup lama kini bisa dikerjakan dalam hitungan menit [12]. *Framework* juga memiliki keunggulan dalam hal keamanan, hal ini dikarenakan *user* menggunakannya dalam jangka panjang [12]. *Framework* juga bersifat *free* sehingga banyak diminati oleh para developer karena dapat membantu developer bekerja lebih cepat [12].

2.9 CodeIgniter

Codeigniter merupakan *application development framework* untuk membangun aplikasi menggunakan PHP [13]. Tujuan dari *codeigniter* tersebut yaitu untuk memungkinkan mengembangkan proyek lebih cepat daripada menulis kode dari awal, dengan menyediakan serangkaian *library* yang biasanya dibutuhkan oleh para pengembang, antarmuka yang sederhana, dan struktur logis untuk mengakses *library* tersebut [13].

Codeigniter didasarkan pada pola pengembangan MVC (*Model View Controller*) [13]. MVC adalah pendekatan perangkat lunak yang memisahkan logika aplikasi dari presentasi [13]. Hal ini memungkinkan halaman web yang dibuat akan mengandung *source code* yang minim dikarenakan skrip PHP yang terpisah [13].

2.10 Hypertext Preprocessor (PHP)

Hypertext Preprocessor (PHP) merupakan bahasa pemrograman *open source* yang dibuat oleh Rasmus Lerdorf pada tahun 1995 sebagai serangkaian skrip *Perl Common Gateway Interface* (CGI). PHP memiliki perkembangan yang signifikan, sejak versi 3 dimana PHP merupakan bahasa pemrograman yang berorientasi objek dan pada versi 5 dimana PHP memiliki tujuan untuk menjadi bahasa pemrograman yang umum dalam pengembangan web. Seperti java, bahasa PHP menggabungkan antarmuka dan pewarisan tunggal. Namun pada versi 7, kinerja PHP menjadi dua kali lebih cepat dari PHP 5. Hingga saat ini dimana PHP 7 telah digunakan untuk mengembangkan sistem manajemen dan pelatihan [14].

2.11 Database (Basis Data)

Database secara sederhana dapat kita artikan sebagai data. Secara teori dimana *database* adalah sekumpulan data atau informasi yang kompleks, data – data tersebut disusun menjadi beberapa kelompok dengan tipe data yang sejenis. Dimana data tersebut akan saling berhubungan satu sama lain atau berdiri sendiri sehingga dapat dengan mudah untuk di akses [15].

2.12 *MariaDB*

MariaDB merupakan sistem manajemen basis data *relasional* yang dikembangkan oleh *MySQL*. *MariaDB* dikembangkan oleh komunitas pengembang yang dimana sebelumnya telah berkontribusi untuk basis data *MySQL*. Alasan dimana pengembang *MySQL* membangun *MariaDB* yaitu telah diakuinya *MySQL* oleh pihak *oracle* sehingga membuat *MySQL* menjadi sebuah produk yang berlisensi *proprietary* [16].

BAB III

PERSIAPAN

Sebelum melakukan pembuatan aplikasi *e-commerce* JURAGAN ada beberapa hal yang harus dipersiapkan terlebih dahulu. Ada beberapa hal yang perlu di *download*, apa saja yang perlu kalian siapkan ?, akan dibahas pada sub-sub bab berikut.

3.1 Xampp

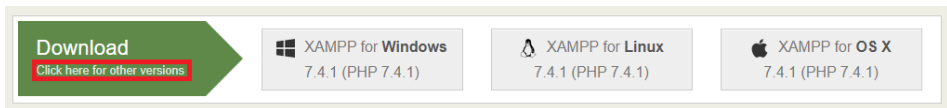
Xampp merupakan sebuah *software* yang memiliki fungsi sebagai server lokal dimana berguna sebagai membuat sebuah *website* yang sifatnya masih dikembangkan. *Xampp* bekerja tanpa menggunakan koneksi internet atau secara *offline* yang dimana layaknya *web hosting* namun tidak dapat diakses oleh banyak orang. Dikarenakan JURAGAN merupakan sebuah *website* yang sifatnya masih dikembangkan maka sebelum dilakukannya *web hosting* kita perlu mempersiapkannya terlebih dahulu. *Xampp* sangat berperan penting untuk memabntu kinerja pengembangan JURAGAN. Bagaimana cara melakukan *install software Xampp* tersebut ?, ikuti langkah – langkah nya sebagai berikut :

1. *Download* terlebih dahulu *sofwate Xampp*. Untuk melakukan *download*, dimana perlu mengunjungi *website* resminya adalah <https://www.apachefriends.org/index.html>. Maka akan terbuka seperti gambar 3.1.



Gambar 3.1 Halaman Utama Web Xampp

2. Klik pada bagian *text “Click here for other versions”*.



Gambar 3.2 Melihat Versions Xampp

3. Anda akan dibawa ke halaman seperti pada gambar 3.3. *download Xampp* versi berapapun. Perlu kalian ketahui dimana dalam *website resmi Xampp* merupakan versi terbaru yang dimana sudah menggunakan PHP 7. Jika anda belum terbiasa dengan PHP 7 silahkan anda mencari *Xampp* veris 3 – 5 keatas yang dimana masih menggunakan PHP 5. Apasih perbedaan PHP 7 dan PHP 5 ?, akan kita bahas di akhir *tutorial instalasi software Xampp*.

Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.

XAMPP for Windows 7.2.26, 7.3.13 & 7.4.1

Version	Checksum	Size
7.2.26 / PHP 7.2.26	What's Included? md5 sha1 Download (64 bit)	145 Mb
7.3.13 / PHP 7.3.13	What's Included? md5 sha1 Download (64 bit)	146 Mb
7.4.1 / PHP 7.4.1	What's Included? md5 sha1 Download (64 bit)	146 Mb

[Requirements](#) [Add-ons](#) [More Downloads](#) »

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

Documentation/FAQs

There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the Forums or Stack Overflow.

- [Linux FAQs](#)
- [Windows FAQs](#)
- [OS X FAQs](#)
- [OS X XAMPP-VM FAQs](#)

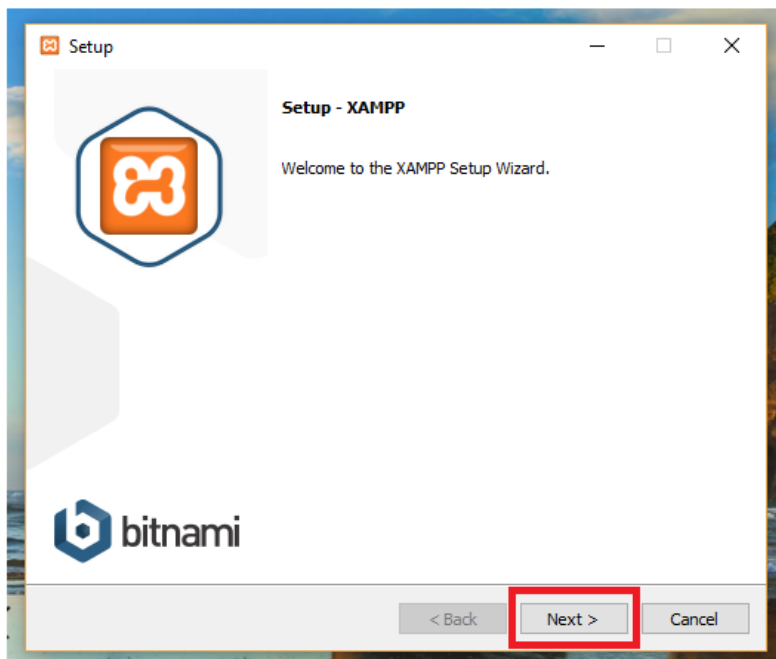
Add-ons and Themes



Bitnami provides a free all-in-one tool to install Drupal, Joomla!, WordPress.

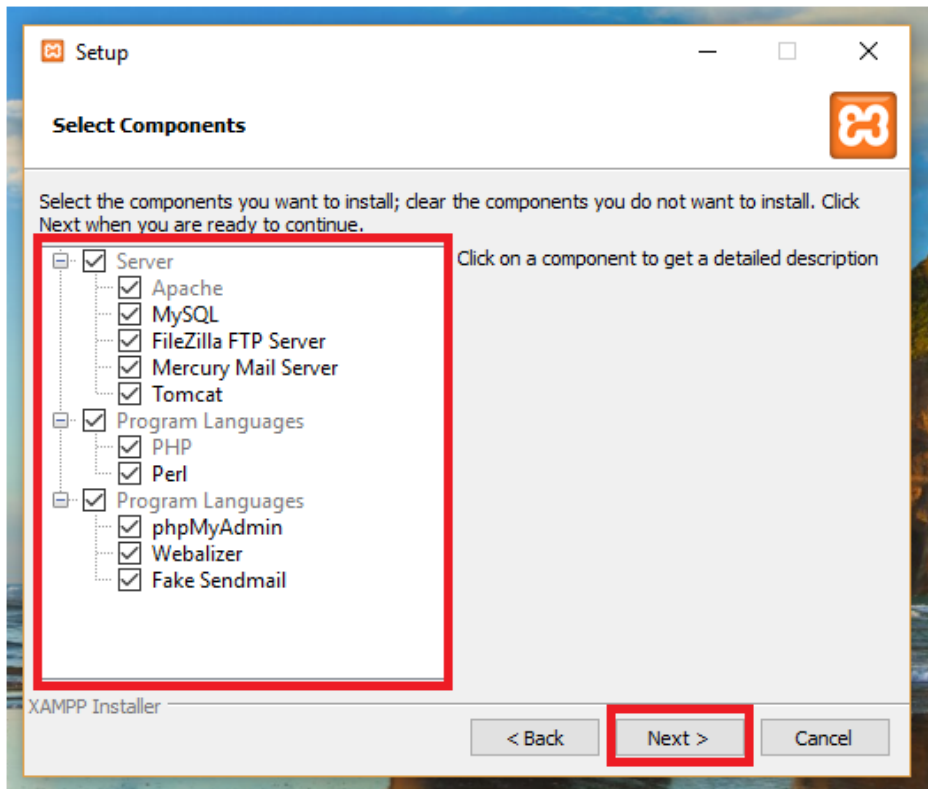
Gambar 3.3 Versi Xampp

4. Jalankan *file* yang sudah anda *download* sebelumnya.
5. Dimana akan muncul sebuah jendela baru yang merupakan indikasi dimulainya proses instalasi. Pilih *next*.



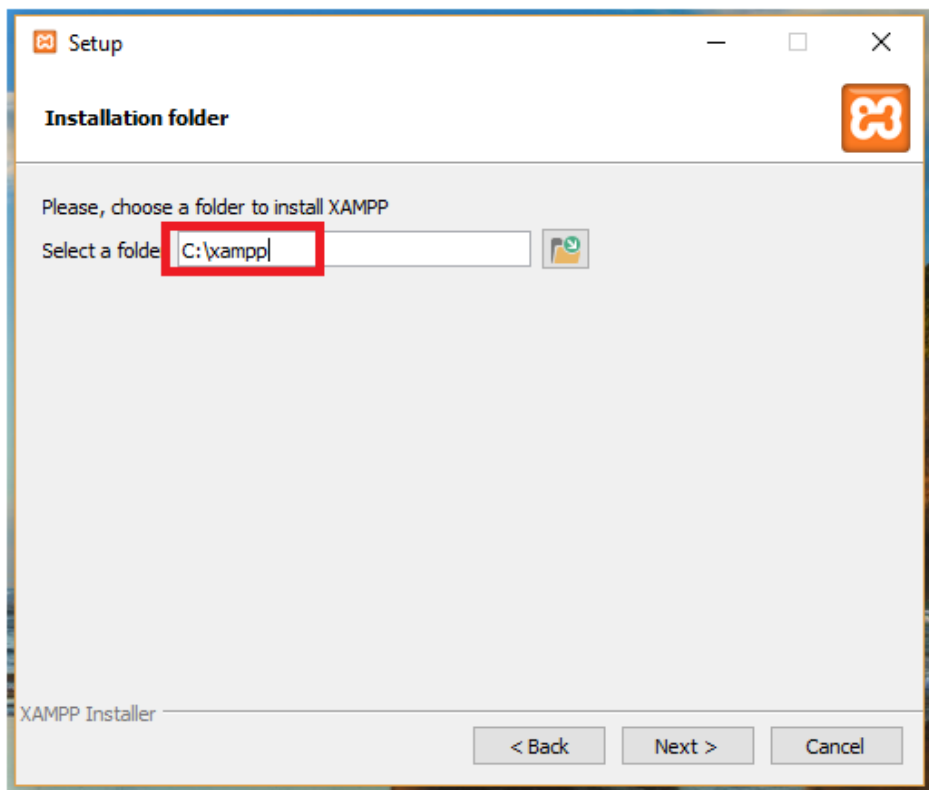
Gambar 3.4 Proses Instalasi Xampp Pertama

6. Pada halaman berikutnya dimana anda akan diminta untuk memilih komponen – komponen yang akan digunakan. Perhatikan gambar 3.5. Klik *next* untuk menuju ke tahap berikutnya.



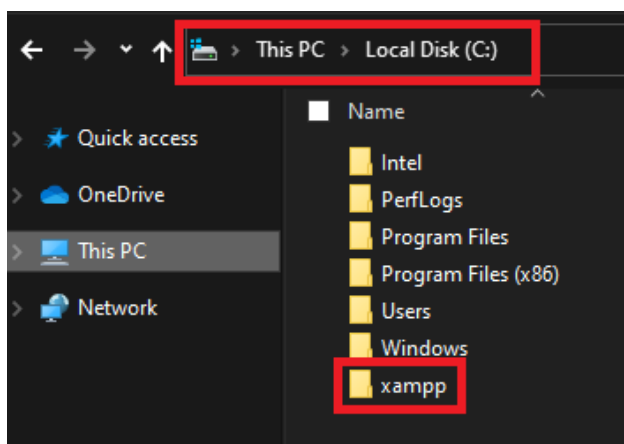
Gambar 3.5 Proses Instalasi Xampp Kedua

7. Pada halaman berikutnya dimana anda diminta untuk memilih tempat *software Xampp* tersebut disimpan. Simpan saja di *local disk C:* lalu klik *next*.



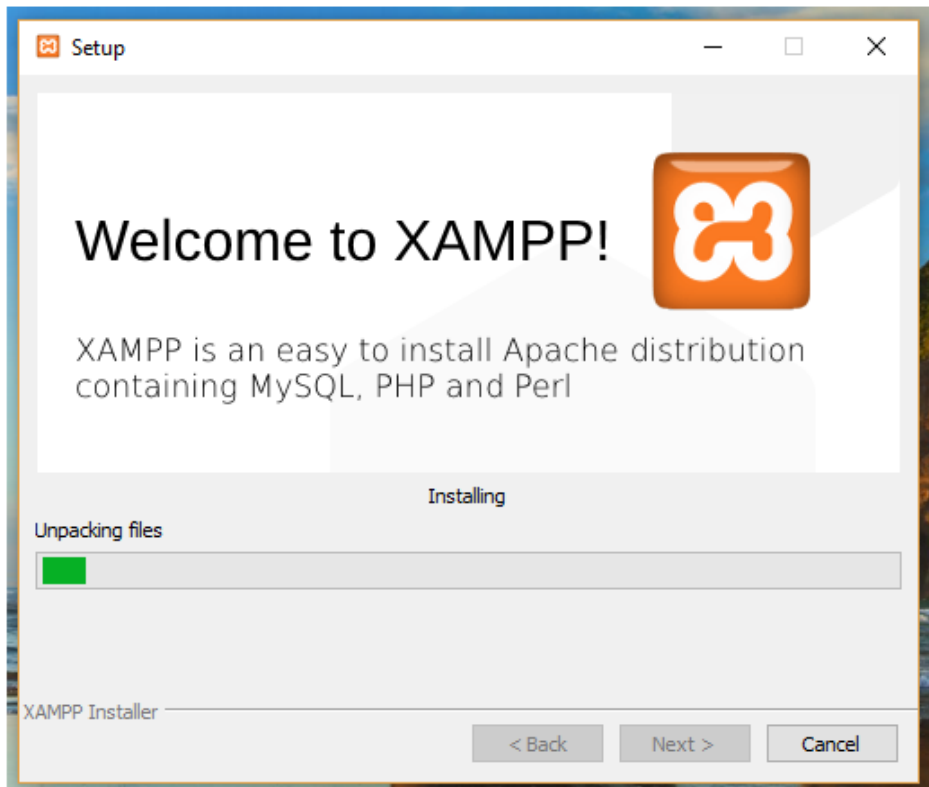
Gambar 3.6 Proses Instalasi Xampp Ketiga

8. Dimana *folder Xampp* anda akan tersimpan.



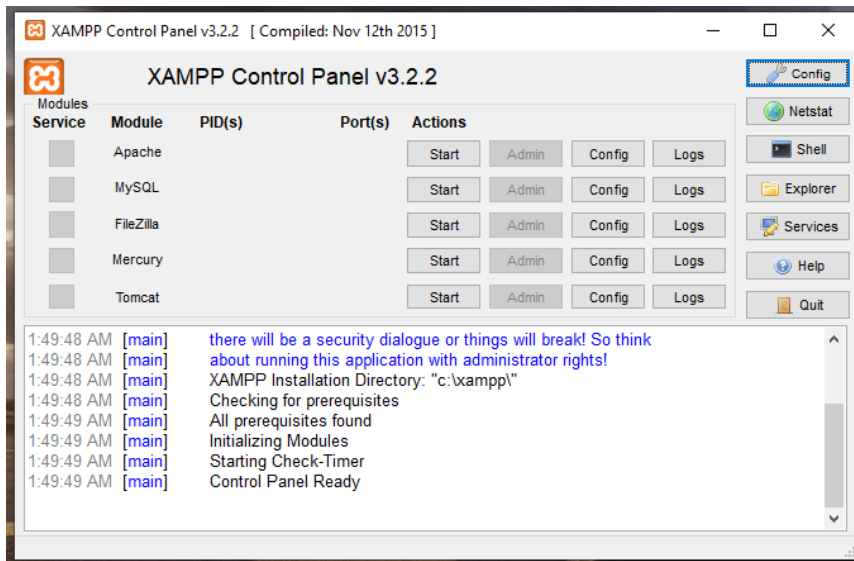
Gambar 3.7 Proses Instalasi Xampp Keempat

9. Tunggu beberapa menit hingga *bar progres* terisi penuh.



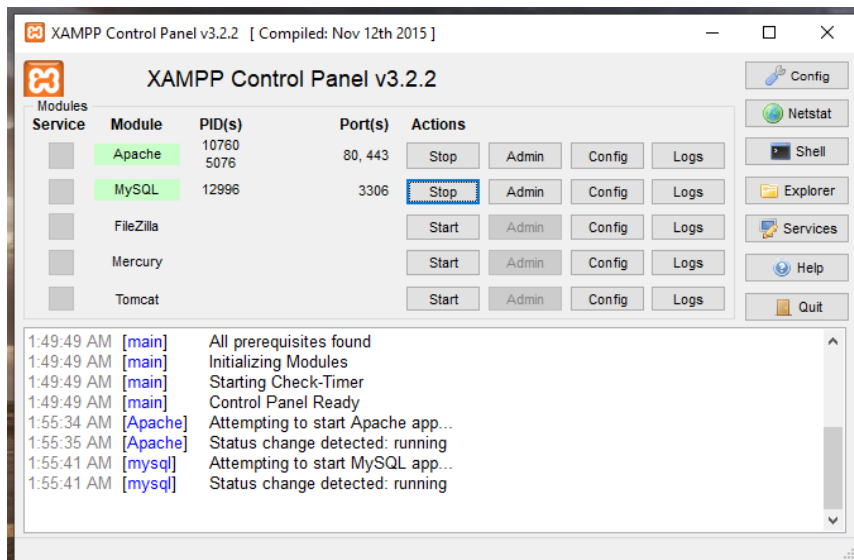
Gambar 3.8 Proses Instalasi Xampp Kelima

10. Jika sudah klik *next* dan coba jalankan *software Xampp* yang sudah terinstall pada laptop / PC anda.



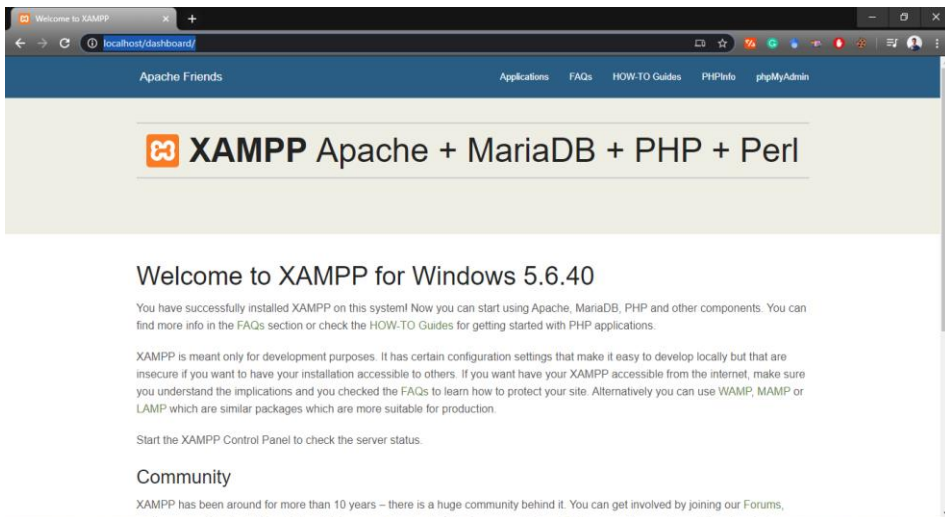
Gambar 3.9 Proses Instalasi Xampp Keenam

11. Berikan *actions start* pada *module Apache* dan *MySQL*. Apabila berganti menjadi wana hijau *software Xampp* anda sudah dapat digunakan.



Gambar 3.10 Proses Instalasi Xampp Ketujuh

12. Buka *web browser* anda dan kunjungi *http://localhost/dashboard/* dan apabila nampak seperti pada gambar 3.11 maka proses *instalasi software* anda berhasil dan siap untuk digunakan.



Gambar 3.11 Porses Instalasi Xampp Selesai

3.1.1 Perbedaan PHP 7 Dan PHP 5

Pada dasarnya dimana PHP 5 adalah sebuah evolusi yang berjalan pada PHP. PHP 5 telah menawarkan peningkatan dari segi fungsionalitas dan penambahan fitur baru yang dimana yaitu seperti dukungan terhadap XML dan juga *Web Service* yang menggunakan libxml2, dukungan terhadap basis data SQLite serta membuat *file swf* dan *applet java*.

Sedangkan untuk PHP 7 memiliki PHPNG (*PHP-Next-Gen*) dimana berfungsi untuk memberikan performa yang maksimal. Peningkatan performa pada PHP 7 ini dikarenakan sebuah *framework Zend* telah melakukan peningkatan kinerja yang sangat besar, dan dimana para developer dapat menggunakan patokan terhadap HHVM.

3.3 Visual Studio Code

Programmer merupakan sebuah pekerjaan yang dimana bertugas dalam menerapkan atau menulis *script code* kepada sistem yang akan dibuat atau dikembangkan. Dalam melakukan aktivitasnya dimana *programmer* memerlukan beberapa *tools* yang dapat mempermudah pekerjaannya, seperti sebuah *software* yang dapat menampung penulisan *scrip codenya*. Ada banyak sekali *tools* untuk membantu *programmer* dalam mengetikkan *script code*-nya seperti *notepad++*, *sublime*, *visual studio code*, dan masih banyak lagi.

Pada pembahasan kali ini dimana penulis menggunakan *visual studio code* atau *Vscode* untuk membangun sistem JURAGAN. *Vscode* merupakan salah satu *text editor* yang paling populer digunakan oleh para *programmer*. Hal ini dikarenakan *Vscode* memiliki beberapa fitur yang dapat mempermudah *programmer* dalam melakukan pengkodean. Salah satunya adalah fitur yang dimana dapat melakukan *copy paste code* secara instan dengan cara menekan tombol kombinasi “*CTRL + SHIFT + DOWN*” untuk meng *copy paster code* ke arah bawah dan “*CTRL + SHIFT + UP*” untuk ke arah atas.

Sebelum melakukan penginstallan *software Vscode* dimana kita harus mengetahui spesifikasi atau *requirements* yang dibutuhkan. Spesifikasi yang dibutuhkan dapat dilihat pada tabel 3.1.

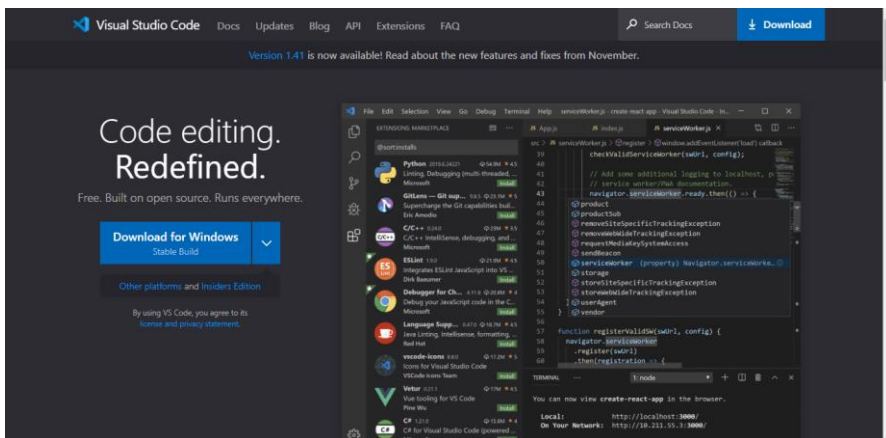
Tabel 3.1 Spesifikasi VScode

<i>Hardware</i>	<i>Operation System</i>	<i>Sarat</i>
<i>Processor 1.6 GHz</i> <i>1 GB RAM</i>	<i>Windows 7, 8.0, 8.1,</i> <i>10</i>	32/64 Bit

	<i>Linux Debian:</i> <i>Ubuntu 14.04, Debian</i> <i>7</i>	<ul style="list-style-type: none"> GLIBCXX Version 4.4.15 Or later
	<i>Linux Red Hat:</i> <i>Enterprise Linux 7,</i> <i>CentOS 7, Fedora 23</i>	<ul style="list-style-type: none"> GLIBC Version 2.15 Or later

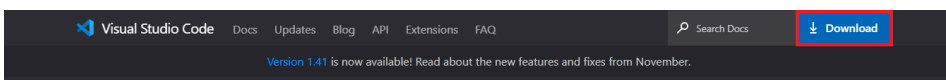
Pada kesempatan kali ini dimana penulis akan memberikan *tutorial* bagaimana cara melakukan *instalasi software Vscode* dengan menggunakan *sistem operation windows*. Ikuti langkah – langkah berikut ini :

1. Kunjungi *link* berikut <https://code.visualstudio.com/>



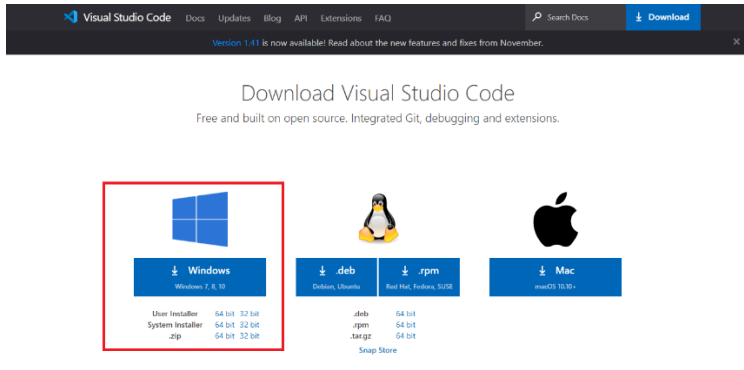
Gambar 3.12 Proses Instalasi Vscode Pertama

2. Pilih *download* pada bagian pojok kanan atas



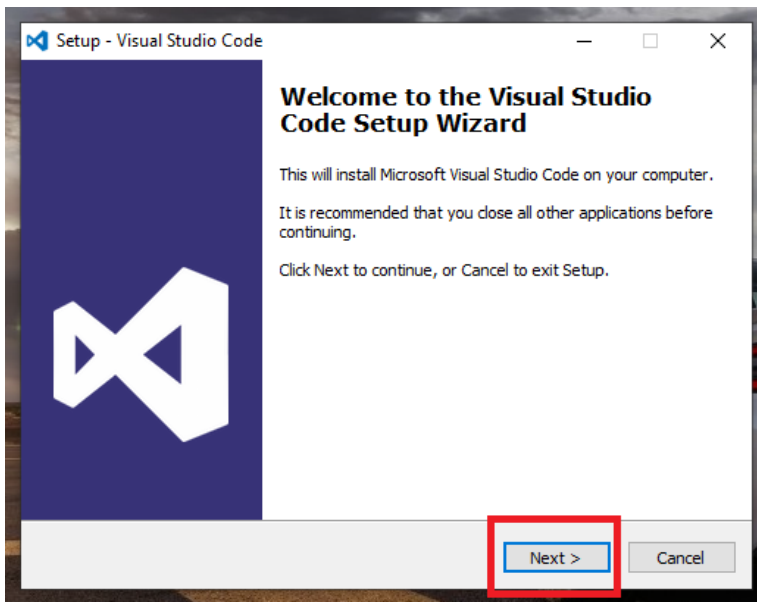
Gambar 3.13 Proses Instalasi Vscode Kedua

3. Karena kita disini akan menggunakan *operatios system windows* silahkan pilih *windows*. Maka sistem akan melakukan proses *download*, tunggu hingga selesai.



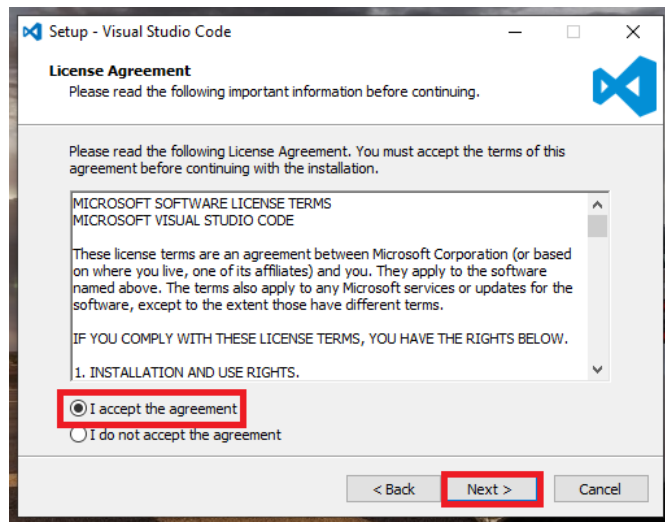
Gambar 3.14 Proses Instalasi Vscode Ketiga

4. Apabila sudah *download* sudah selesai silahkan jalankan *file* tersebut dengan cara “*Run Administrator*” lalu klik *next*.



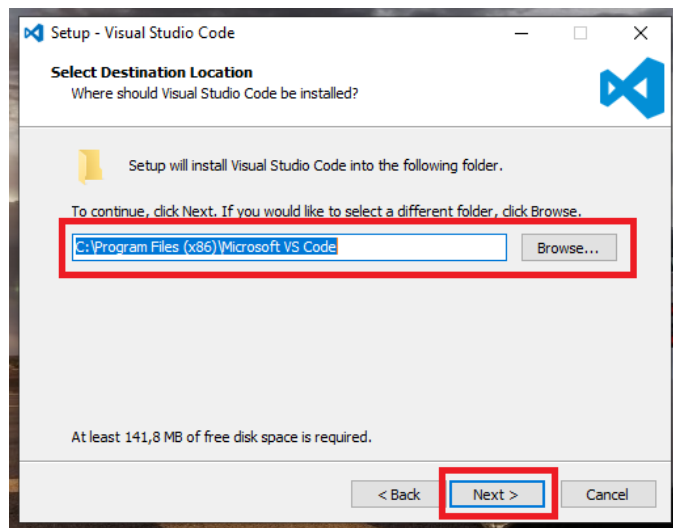
Gambar 3.15 Proses Instalasi Vscode Kempat

5. Pada tahap selanjutnya silahkan pilih “*I accept the agreement*” lalu klik *next*.



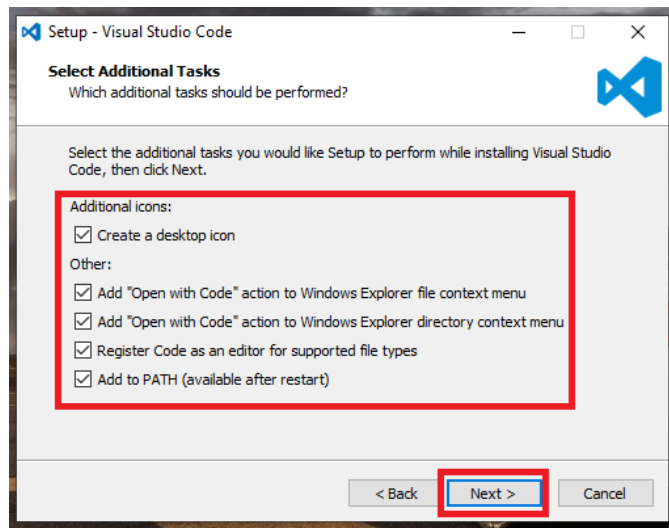
Gambar 3.16 Proses Instalasi Vscode Kelima

6. Silahkan pilih dimana anda akan menyimpan *file Vscode* tersebut lalu klik *next*.



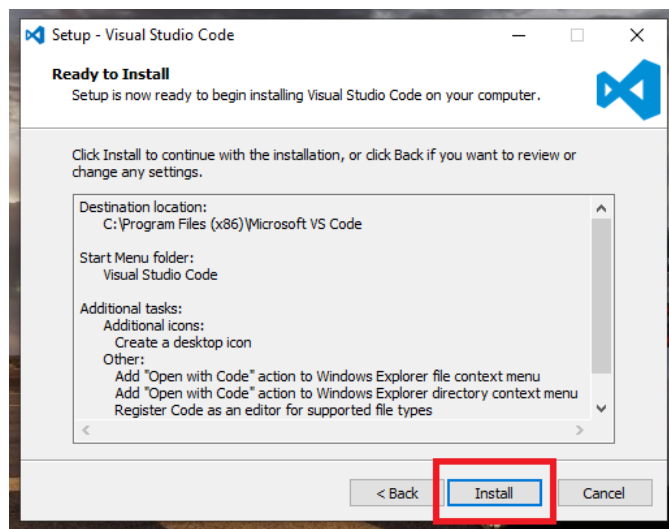
Gambar 3.17 Proses Instalasi Vscode Keenam

7. Pada bagian *select additional task* silahkan centang semua dan klik *next*.



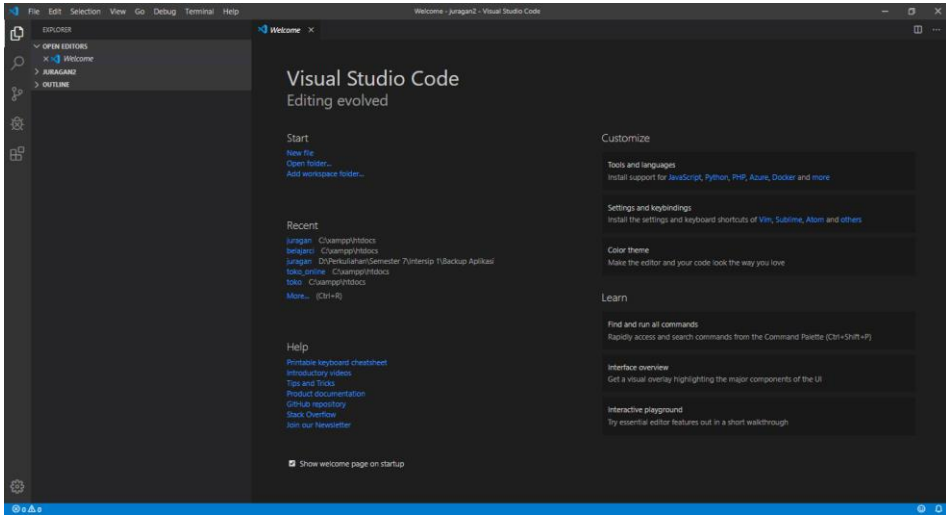
Gambar 3.18 Proses Instalasi Vscode Ketujuh

8. Klik *install* untuk memulai proses. Dimana akan muncul progres bar, silahkan tunggu hingga penuh.



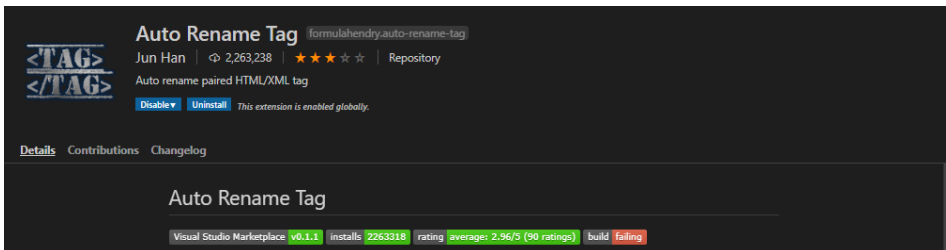
Gambar 3.19 Proses Instalasi Vscode Kedelapan

9. Apabila progres bar sudah penuh maka proses *install* telah selesai dan *Vscode* akan berada pada halaman *desktop* anda. Buka *Vscode* yang sudah ter-*install*. Tampilan *Vscode* akan terlihat seperti pada gambar 3.20.



Gambar 3.20 Proses Instalasi Vscode Selesai

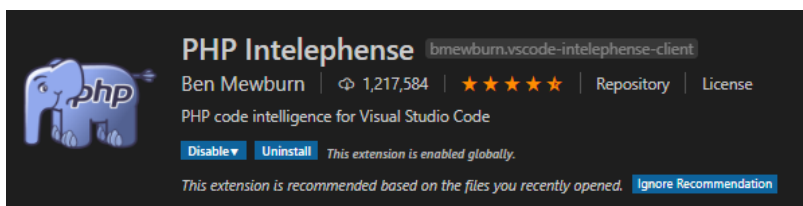
Apabila proses *install software Vscode* sudah selesai dimana anda perlu melakukan *install* beberapa *extensions* untuk membantu proses pengkodean. Apa saja *extensions* tersebut ?, yang perlu anda *install* adalah “*Auto Rename Tag*”.



Gambar 3.21 Extensions Auto Rename Tag

Extensions tersebut berfungsi untuk merapihkan *scrip code* anda saat melakukan *save*. Ketika anda melakukan *save* dengan cara menekan tombol kombinasi “*CTRL + S*” dimana *script code* yang anda ketik akan otomatis dirapihkan. Hal tersebut sangat bermanfaat karena *script code* yang anda ketikan akan terlihat lebih rapih.

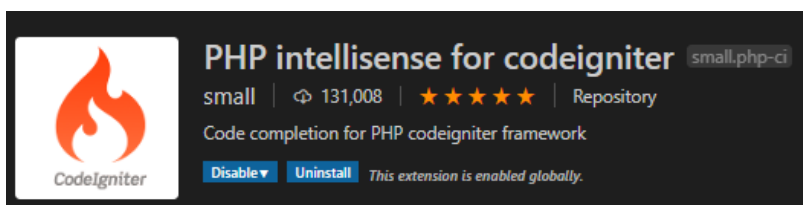
Extensions yang kedua dimana anda perlu meng-*install* “*PHP Intelephense*”.



Gambar 3.22 *Extensions PHP Intelephense*

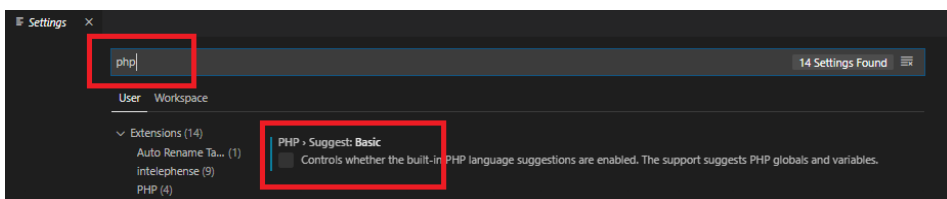
Extensions tersebut dimana memiliki fungsi sebagai memberikan fitur lengkap terhadap *PHP* dengan saran yang sangat detail dengan dukungan “*Go To*” langsung kepada sumbernya. Dengan adanya *extensions* tersebut dimana anda tidak perlu mengingat semua *sintaks* perintah yang akan anda ketik, hal ini dikarenakan fitur *intelephense* dapat bekerja tanpa harus melakukan konfigurasi.

Ektensions yang terakhir dimana anda perlu melakukan *install* “*PHP Intellisense for codeigniter*”.



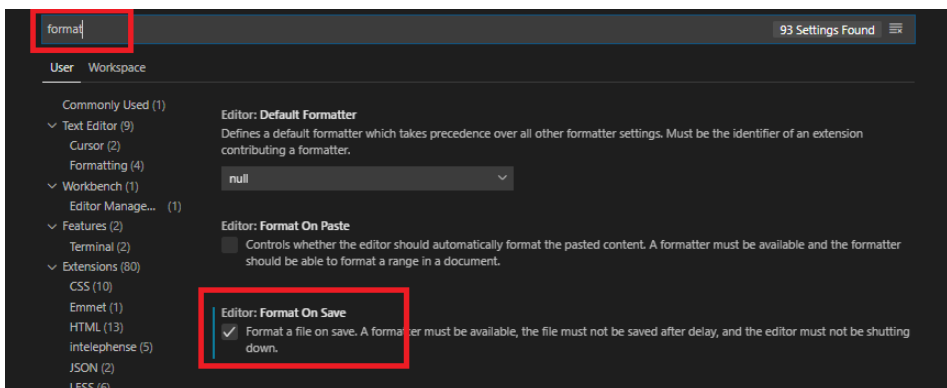
Gambar 3.23 *Ekstensions PHP Intellisense For CodeIgniter*

Extensions tersebut memiliki fungsi untuk memudahkan anda dalam melakukan *script code* pemanggilan *PHP* yang berada pada *codeigniter*. Untuk menggunakan semua *extensions* yang telah anda *install* dimana perlu dilakukan *setting* kembali pada *software Vscode*. Pilih *preferences* → *Setting*, maka akan terbuka seperti pada gambar 3.24. Ketikkan ”*PHP*” pada kolom pencarian, lalu hilangkan centang pada bagian “*PHP Suggest Basic*” agar proses *extensions PHP Intelephense* yang akan berjalan.



Gambar 3.24 Setting Vscode 1

Langkah yang kedua silahkan ketik “*format*” pada kolom pencarian, berikan centang pada bagian “*format on save*” agar saat anda melakukan *save script code*-nya akan otomatis dirapihkan.



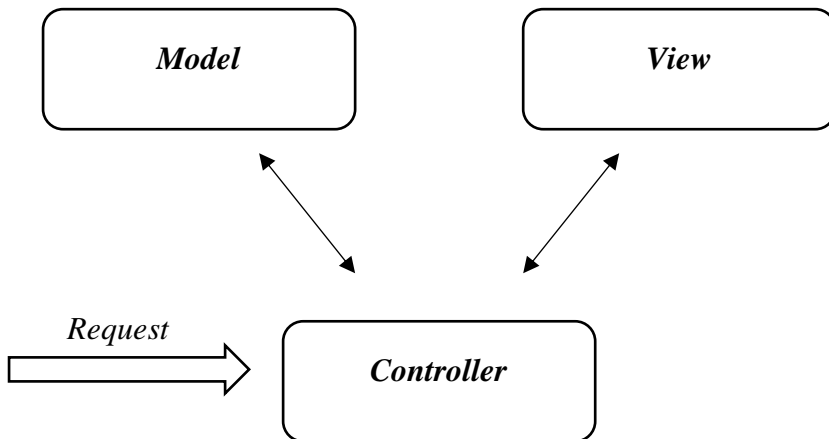
Gambar 3.25 Setting Vscode 2

3.4 CodeIgniter

Codeigniter merupakan *application development framework* untuk membangun aplikasi menggunakan PHP [13]. Tujuan dari *codeigniter* tersebut yaitu untuk memungkinkan mengembangkan proyek lebih cepat daripada menulis kode dari awal, dengan menyediakan serangkaian *library* yang biasanya dibutuhkan oleh para pengembang, antarmuka yang sederhana, dan struktur logis untuk mengakses *library* tersebut [13]. *Codeigniter* didasarkan pada pola pengembangan MVC [13]. MVC adalah pendekatan perangkat lunak yang memisahkan logika aplikasi dari presentasi [13]. Hal ini memungkinkan halaman web yang dibuat akan mengandung *source code* yang minim dikarenakan skrip PHP yang terpisah [13]. MVC akan melakukan pembagian aplikasi menjadi tiga bagian yang fungsional yaitu *model*, *view*, dan *controller* dimana memiliki pengertian sebagai berikut :

- **Model** merupakan tempat berkumpulnya *script code* untuk memulai proses bisnis dan data. Dimana *model* akan berhubungan langsung dengan basis data (*database*) untuk melakukan eksekusi sebuah *query insert, update, delete*. *Model* juga akan menangani proses validasi pada bagian *controller* akan tetapi *model* tidak dapat berhubungan langsung dengan bagian *view*.
- **View** adalah tempat untuk menangani logika presentasi yang didalamnya terdapat *script code* untuk membuat desain interaksi atau tampilan dari sistem yang akan dibuat.
- **Controller** adalah penghubung antara *model* dan *view* yang dimana akan menerima sebuah *request – request* dan data dari pengguna

(*user*). Dimana *controller* akan menentukan apa yang akan di proses oleh sistem tersebut.



Gambar 3.26 Konsep MVC

3.4.1 Sejarah *CodeIgniter*

Rick Ellis yang merupakan seorang musisi musik dengan genre rock yang dimana profesinya beralih menjadi seorang *programmer* yang dimana pada sebuah riset kecil – kecilan telah menghasilkan sebuah *framework PHP* dengan ukuran yang cukup kecil dan ringan serta dapat memenuhi fitur umum pada aplikasi *PHP*. Rick Ellis dialah seseorang yang pertama kali menuliskan *codeigniter*. Pada tanggal 28 Februari 2016 dimana *codeigniter* pertama kali dirilis, akan tetapi pada tahun 2014 dimana *framework* tersebut telah dimiliki oleh BCIT (*British Columbia Institute Of Technology*).

3.4.2 Tutorial install Codeigniter

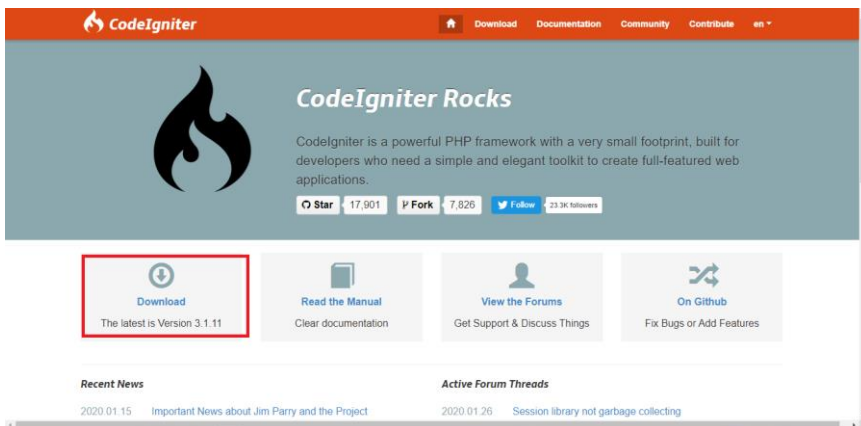
Bagaimana cara melakukan *instalasi frameword codeigniter* ?, kali ini penulis akan memberikan *tutorial download framework codeigniter*. Ikuti langkah – langkah berikut :

1. Kunjungi *website resmi codeigniter* di <https://codeigniter.com/>.



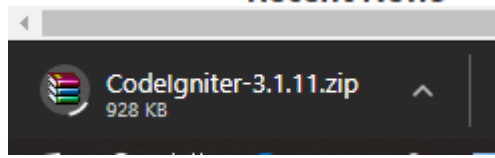
Gambar 3.27 CI Tahap 1

2. Klik pada bagian *download* seperti pada gambar 3.28.



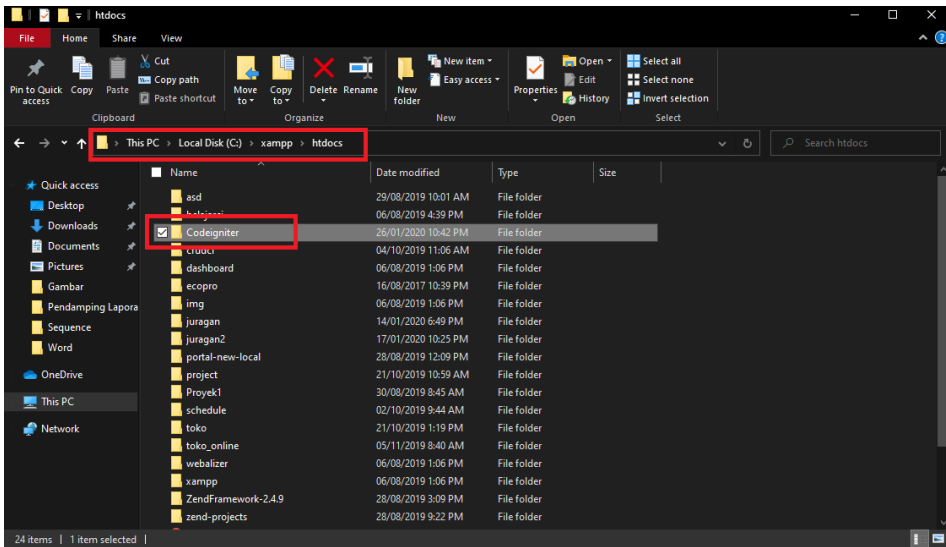
Gambar 3.28 CI Tahap 2

3. Tunggu beberapa menit hingga proses *download* selesai.



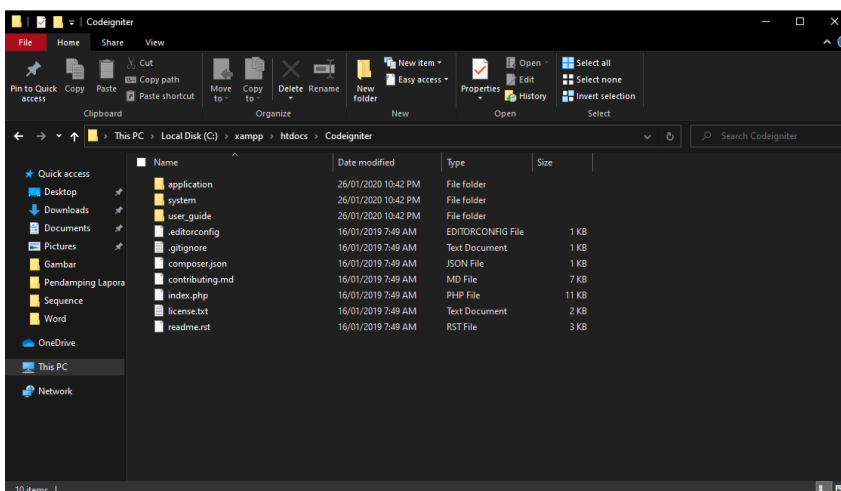
Gambar 3.29 CI Tahap 3

4. Lakukan *extract* pada *file zip* yang sudah anda *download* tersebut dan simpan pada *folder "Xampp/htdocs/"*. *Rename* saja namanya menjadi *Codeigniter*.



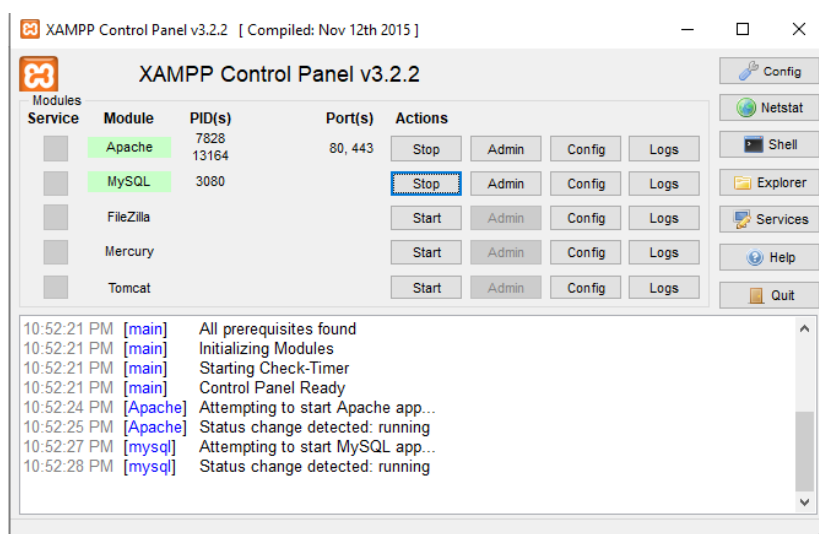
Gambar 3.30 CI Tahap 4

5. Buka *folder codeigniter* tersebut pastikan isinya sama seperti pada gambar 3.31.



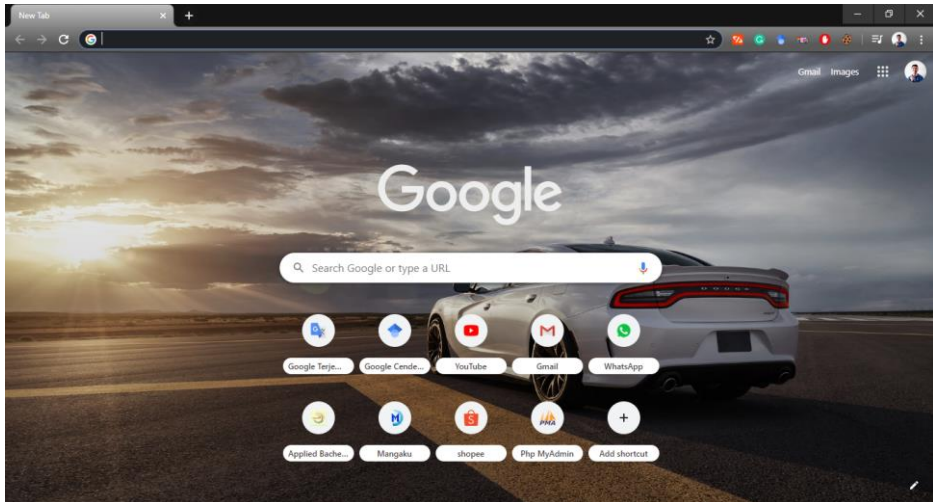
Gambar 3.31 CI Tahap 5

6. Untuk menguji *framework codeigniter* tersebut bisa digunakan atau tidak dimana and perlu melakukan pengujian terlebih dahulu dengan menggunakan bantuan *software Xampp* yang sudah di *download* dengan mengikuti langkah dari gambar 3.1 – 3.11.
7. Bula *software Xampp* tersebut dan jalankan *Apache* serta *MySQL*.



Gambar 3.32 CI Tahap 6

8. Buka *web browser* yang ada gunakan.



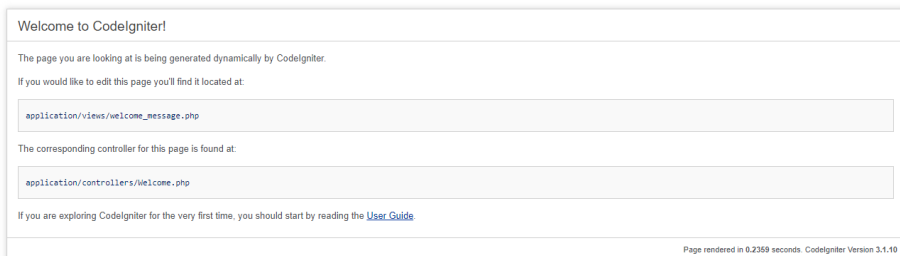
Gambar 3.33 CI Tahap 7

9. Ketikkan *http://localhost/codeigniter/* pada bagian pencarian.



Gambar 3.34 CI Tahap 8

10. Apabila tampilan yang muncul terlihat seperti pada gambar 3.35 maka *framework codeigniter* anda sudah berhasil ter-*install*.



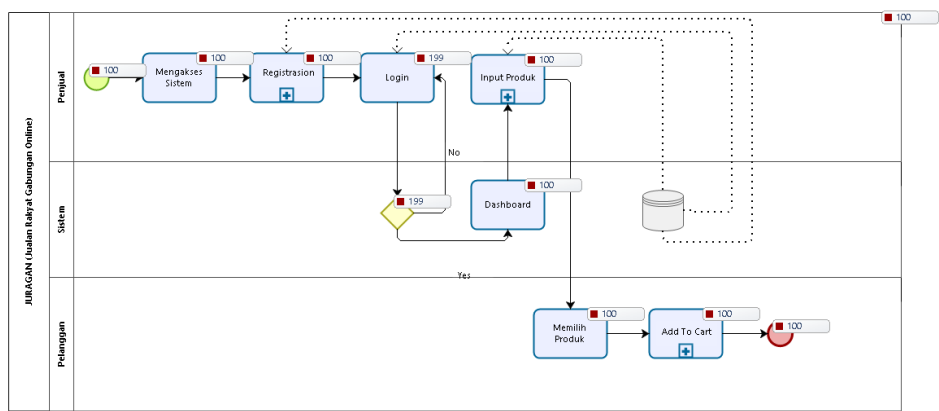
Gambar 3.35 CI Selesai

BAB IV

PERANCANGAN

4.1 BPMN Sistem JURAGAN

Dalam membuat sebuah sistem dimana perlu dilakukan tahap perancangan terlebih dahulu. Perancangan sistem dibuat dengan tujuan untuk menghindari kesalahan dalam pembuatan sistem sehingga sistem dapat berjalan dengan baik. Dimana penulis telah membuat sebuah analisis *business process model and notation (BPMN)* pada sistem JURAGAN adalah sebagai berikut :



Gambar 4.1 BPMN Sistem JURAGAN

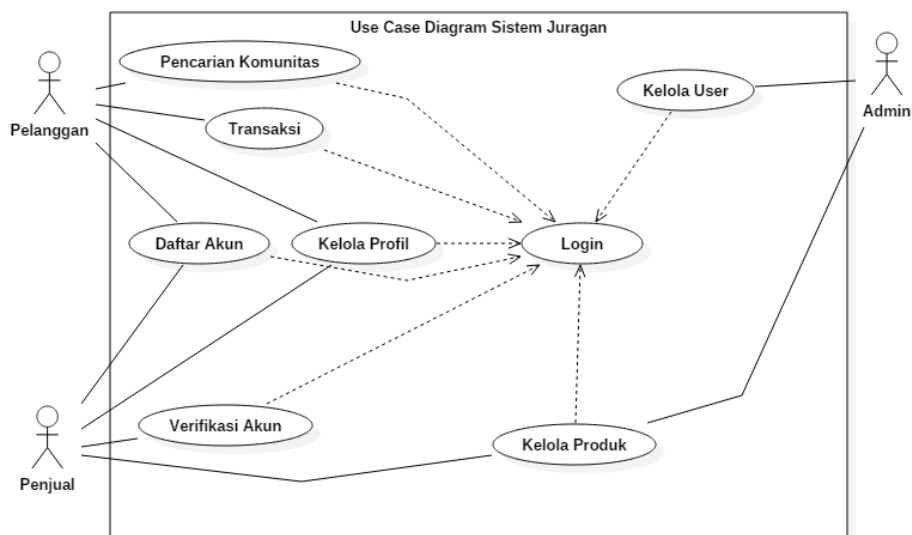
Tabel 4.1 Hasil Result BPMN Sistem JURAGAN

<i>Name</i>	<i>Type</i>	<i>Instances Completed</i>
Sistem JURAGAN	<i>Process</i>	<i>100</i>
<i>NoneStart</i>	<i>Start Event</i>	<i>100</i>

Mengakses Sistem	<i>Task</i>	<i>100</i>
<i>Login</i>	<i>Task</i>	<i>199</i>
<i>ExclusiveGateway</i>	<i>Gatewat</i>	<i>199</i>
<i>Dashboard</i>	<i>Task</i>	<i>100</i>
Memilih Produk	<i>Task</i>	<i>100</i>
<i>NoneEnd</i>	<i>End Event</i>	<i>100</i>
<i>Registration</i>	<i>Task</i>	<i>100</i>
Input Produk	<i>Task</i>	<i>100</i>
<i>Add To Cart</i>	<i>Task</i>	<i>100</i>

4.2 Use Case JURAGAN

Use Case Diagram merupakan salah satu diagram yang diklasifikasikan ke dalam aspek perilaku, dimana deskripsi perilaku dari setiap *use case* dijelaskan secara detail dan terpisah dengan menggunakan *document* secara *tekstual* yaitu *user case scanario*. Adapun *use case* dari sistem JURAGAN tersebut sebagai berikut :



Gambar 4.2 Use Case Sistem JURAGAN

4.2.1 Definisi Aktor

Tabel 4.2 Definisi Aktor Use Case

No.	Aktor	Deskripsi
1.	Pelanggan	a. Daftar Akun b. <i>Login</i> c. Kelola Profil d. Transaksi e. Pencarian Komunitas
2.	Penjual	a. Daftar Akun b. <i>Login</i> c. Kelola Profil d. Verifikasi Akun e. Kelola Produk

3	Admin	<ul style="list-style-type: none"> a. Kelola User b. Kelola Produk
---	-------	--

4.2.2 Definisi Use Case

Tabel 4.3 Definisi Use Case

No.	Use Case	Deskripsi
1.	Daftar Akun	a. Dapa melakukan sebuah proses pendaftaran atau <i>registrasion</i> akun pada sistem JURAGAN.
2.	<i>Login</i>	a. Melakukan proses <i>login</i> terhadap akun yang sudah melakukan <i>registrasion</i> atau sudah terdaftar di dalam sistem JURAGAN.
3.	Kelola Profil	a. Melakukan pengelolaan profil terhadap akun <i>user</i> .
4.	Pencarian Komunitas	a. Dapat melakukan pencarian sebuah produk berdasarkan komunitas.
5.	Transaksi	<ul style="list-style-type: none"> a. Melakukan penambahan produk kedalam keranjang. b. Melakukan proses transaksi.
6.	Verifikasi Akun	a. Mengaktifkan akun toko yang sudah terdaftar dengan

		menginputkan No KTP dan <i>upload</i> KTP
7	Kelola Produk	a. Melakukan pengelolaan produk terhadap akun level toko. b. Melakukan <i>view</i> produk terhadap akun level pelanggan.
8	Kelola User	a. Melakukan pengelolaan akun <i>user</i> oleh admin.

4.2.3 Use Case Scenario

Tabel 4.4 Use Case Scenario Daftar Akun

Identifikasi	
No.	JR1
Nama	Daftar Akun
Tujuan	Mendaftarkan akun pada sistem JURAGAN.
Deskripsi	Melakukan proses pendaftaran akun guna dapat melakukan proses <i>login</i> pada sistem JURAGAN.
Aktor	Pelanggan Dan Penjual

Skenario	
Kondisi Awal	Halaman registrasi
Aksi Aktor	Reaksi Sistem
1. Melakukan <i>input</i> data registrasi yang dibutuhkan sistem.	a. -
2. Menekan <i>button</i> daftar akun.	b. Melakukan validasi terhadap data yang dikirim.
3. -	c. Memberikan notifikasi akun berhasil terdaftar dan harus di aktivasi melalui <i>email</i> .
4. Membuka <i>email</i> yang digunakan untuk mendaftar.	d. Mengirimkan <i>email</i> kepada <i>user</i> yang mendaftar.
5. Membuka <i>email</i> dan melakukan aktivasi.	e. Mengirimkan <i>token</i> kepada tabel <i>user_token</i> .
6. -	f. Memberikan nortifikasi akun sudah aktif.

Tabel 4.5 Use Case Scenario Login

Identifikasi	
No.	JR2

Nama	<i>Login</i>
Tujuan	Membuka akses lebih luas pada sistem JURAGAN.
Deskripsi	Mengakses sistem JURAGAN secara menyeluruh dengan kategori <i>user</i> tertentu.
Aktor	Pelanggan Dan Penjual
Skenario	
Kondisi Awal	Halaman login
Aksi Aktor	Reaksi Sistem
1. Melakukan <i>input email</i> dan <i>password</i> yang sudah terdaftar pada sistem	a. -
2. Memberikan <i>action</i> pada <i>button login</i>	b. Memulai proses validasi
3. -	c. Menampilkan halaman utama sistem JURAGAN.

Tabel 4.6 Use Case Scenario Kelola Profil

Identifikasi	
No.	JR3
Nama	Kelola Profil

Tujuan	Melengkapi data profil.
Deskripsi	Melakukan pengelolaan terhadap data profil <i>user</i> .
Aktor	Pelanggan dan Penjual
Skenario	
Kondisi Awal	Halaman Edit Profil
Aksi Aktor	Reaksi Sistem
1. Melakukan <i>input</i> data profil secara lengkap.	a. -
2. Memberikan <i>action</i> terhadap <i>button edit profile</i>	b. Melakukan validasi terhadap data yang dikirim.
3. -	c. Memberikan notifikasi edit profil berhasil.

Tabel 4.7 Use Case Scenario Pencarian Komunitas

Identifikasi	
No.	JR4
Nama	Pencarian Komunitas
Tujuan	Mencari produk berdasarkan komunitas.
Deskripsi	Melakukan pencarian produk dari <i>keyword</i> komunitas yang dikirim.

Aktor	Pelanggan
Skenario	
Kondisi Awal	<i>Topbar</i> halaman utama
Aksi Aktor	Reaksi Sistem
1. Melakukan <i>input keyword</i> komunitas.	a. Menampilkan produk berdasarkan <i>keyword</i> komunitas yang dikirim.

Tabel 4.8 Use Case Scenario Transaksi

Identifikasi	
No.	JR5
Nama	Transaksi
Tujuan	Menambahkan produk ke dalam keranjang belanja.
Deskripsi	Menambahkan produk ke dalam keranjang belanja untuk melakukan proses transaksi.
Aktor	Pelanggan
Skenario	
Kondisi Awal	Halaman Produk
Aksi Aktor	Reaksi Sistem

1. Memberikan aksi terhadap <i>button</i> tambah ke keranjang	a. Mengirimkan <i>id</i> produk yang ditambahkan.
2. -	b. Melakukan validasi
3. -	c. Menambahkan produk ke dalam keranjang.

Tabel 4.9 Use Case Scenario Verifikasi Akun

Identifikasi	
No.	JR6
Nama	Verifikasi Akun
Tujuan	Mengaktifkan akun toko.
Deskripsi	Mengaktifkan akun toko dengan memasukkan No KTP dan <i>upload</i> foto KTP.
Aktor	Penjual
Skenario	
Kondisi Awal	Halaman Verifikasi Akun
Aksi Aktor	Reaksi Sistem
1. Menginputkan no KTP	a. -
2. <i>Upload</i> foto KTP	b. -

3. Memberikan aksi pada <i>button</i> verifikasi akun	c. Melakukan validasi terhadap data yang dikirim
4. -	d. Memberikan notifikasi akun toko sudah aktif.

Tabel 4.10 Use Case Scenario Kelola Produk Sub Penjual

Identifikasi	
No.	JR7
Nama	Kelola Produk
Tujuan	Mengelola data produk
Deskripsi	Melakukan pengelolaan terhadap data produk.
Aktor	Penjual
Skenario	
Kondisi Awal	Halaman Inventori
Aksi Aktor	Reaksi Sistem
1. Memberikan aksi terhadap <i>button</i> tambah produk	a. Memunculkan <i>pop up form input</i> produk
2. Melakukan <i>input</i> data produk yang dibuthkan oleh sistem	b. -

3. Memberikan aksi pada <i>button</i> simpan produk	c. Melakukan validasi terhadap data yang dikirim
4. -	d. Menyimpan data ke dalam tabel barang
5. -	e. Melakukan <i>view</i> produk
6. Memberikan aksi pada <i>button</i> edit produk	f. Menampilkan <i>form</i> edit produk
7. Melakukan edit data produk.	g. -
8. Memberikan aksi pada <i>button</i> edit produk	h. Melakukan validasi terhadap data yang dikirim
9. -	i. Memberikan notifikasi produk berhasil di edit

Tabel 4.11 Use Case Scenario Kelola Produk Sub Admin

Identifikasi	
No.	JR8
Nama	Kelola Produk
Tujuan	Mengelola data produk
Deskripsi	Melakukan pengelolaan terhadap data produk.

Aktor	Admin
Skenario	
Kondisi Awal	Halaman Admin
Aksi Aktor	Reaksi Sistem
1. Memilih menu data produk pada <i>sidebar</i>	a. Menampilkan halaman data produk
2. -	b. Mengambil data produk dari tabel barang
3.	c. Menampilkan produk

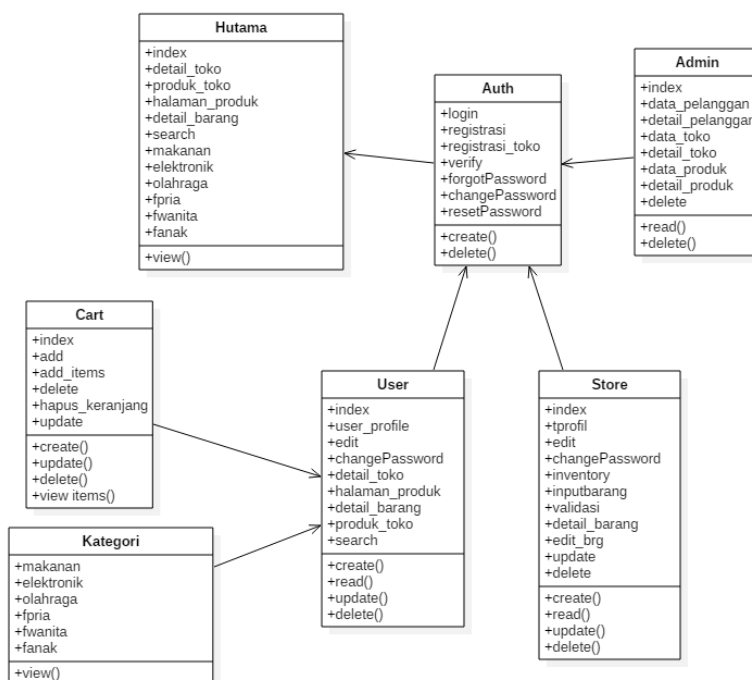
Tabel 4.12 Use Case Scenario Kelola User

Identifikasi	
No.	JR9
Nama	Kelola User
Tujuan	Mengelola user yang telah terdaftar
Deskripsi	Melakukan pengelolaan data terhadap <i>user</i> yang terdaftar
Aktor	Admin
Skenario	
Kondisi Awal	Halaman Admin
Aksi Aktor	Reaksi Sistem

1. Memilih menu <i>user</i> pada <i>sidebar</i>	a. Menampilkan halaman user
2. -	b. Mengambil data dari tabel <i>user</i>
3. -	c. Menampilkan data <i>user</i>

4.3 Class Diagram Sistem JURAGAN

Class diagram adalah sebuah spesifikasi yang jika tidak diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek. Adapun *class* diagram dari sistem JURAGAN adalah sebagai berikut :

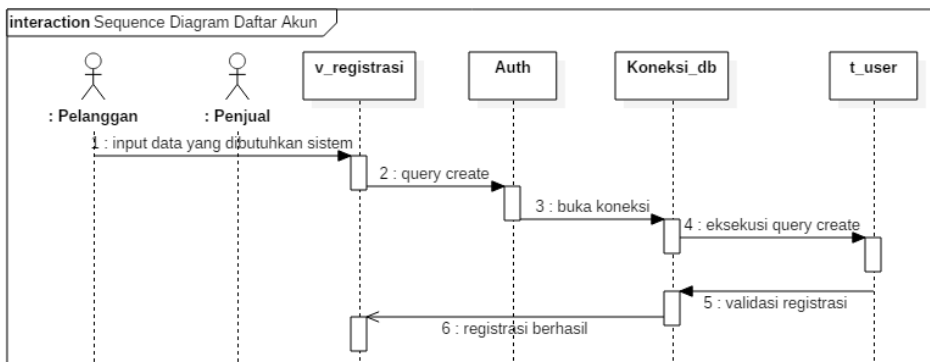


Gambar 4.3 Class Diagram Sistem JURAGAN

4.4 Sequence Diagram Sistem JURAGAN

Sequence diagram adalah suatu diagram yang memperlihatkan atau menampilkan interaksi – interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Adapun *sequence* diagram dari sistem JURAGAN adalah sebagai berikut:

1). *Sequence* Diagram Daftar Akun

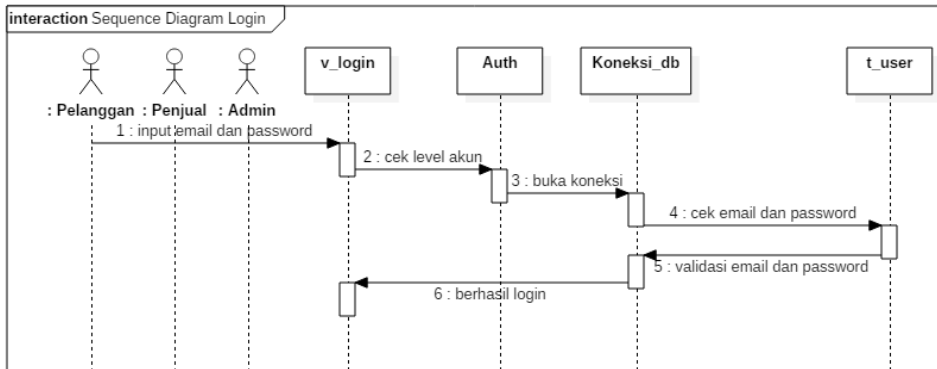


Gambar 4.4 *Sequence* Daftar Akun

Keterangan :

1. Pelanggan dan Penjual akan melakukan *input* data yang dibutuhkan.
2. *View registrasi* akan membuat *query create* melalui *controller Auth*.
3. *Controller Auth* akan membuka koneksi ke *database*.
4. Eksekusi *query create* akan di lakukan kepada tabel *user*.
5. Validasi *registrasi* akan di lakukan oleh tabel *user*.
6. *Registrasi* berhasil

2). Sequence Login

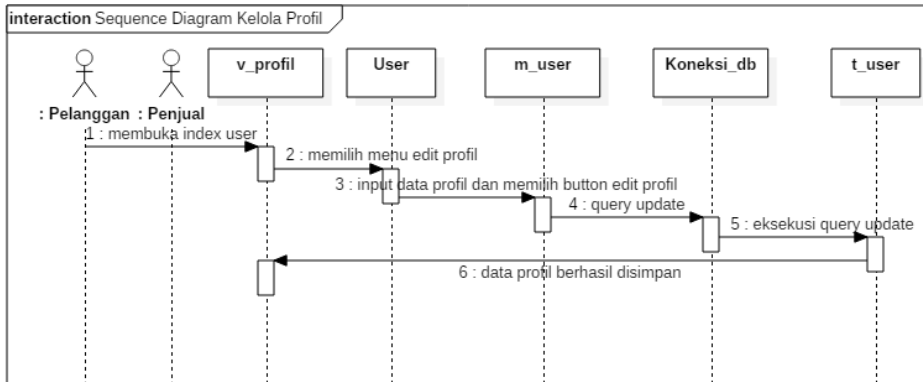


Gambar 4.5 Sequence Login

Keterangan :

1. Pelanggan, Penjual, dan admin akan melakukan *input email dan password*.
2. Sistem akan melakukan cek level akun pada *controller Auth*.
3. *Controller Auth* akan membuka koneksi *database*.
4. Di dalam *database* akan dilakukan proses pengecekan *email* atau *password* yang sudah terdaftar.
5. *Database* juga akan melakukan *validasi* terhadap tabel *user*.
6. Apabila lolos maka *user* berhasil melakukan *login*.

3). Sequence Kelola Profil

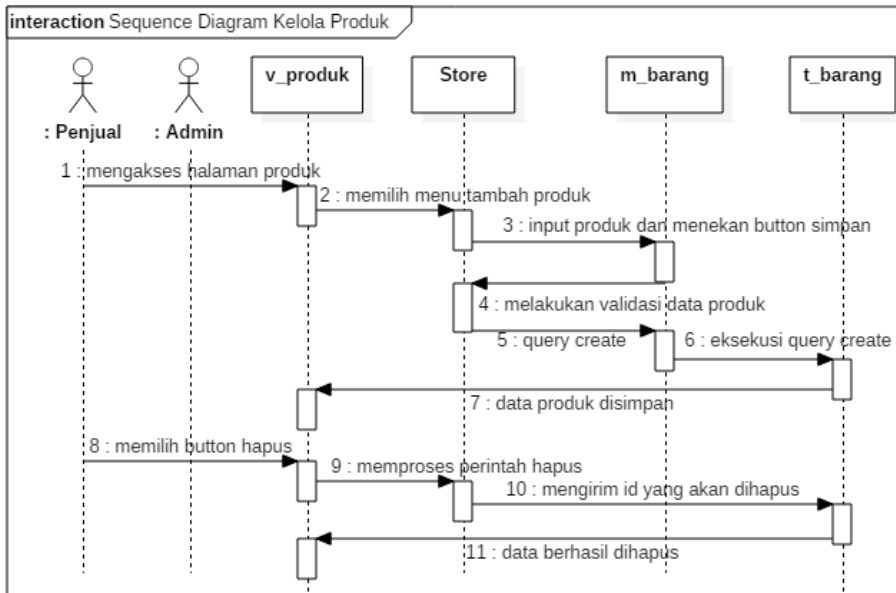


Gambar 4.6 Sequence Kelola Profil

Keterangan :

1. *User* akan mengakses halaman *index* nya masing - masing
2. *User* akan memilih menu edit profil
3. *User* melakukan *input* data profil dan apabila sudah *user* akan memilih *button* edit profil
4. *Model user* akan memberikan *query update* kepada *database*
5. *Database* akan mengeksekusi *query update*
6. Data profil berhasil disimpan.

4). Sequence Kelola Produk



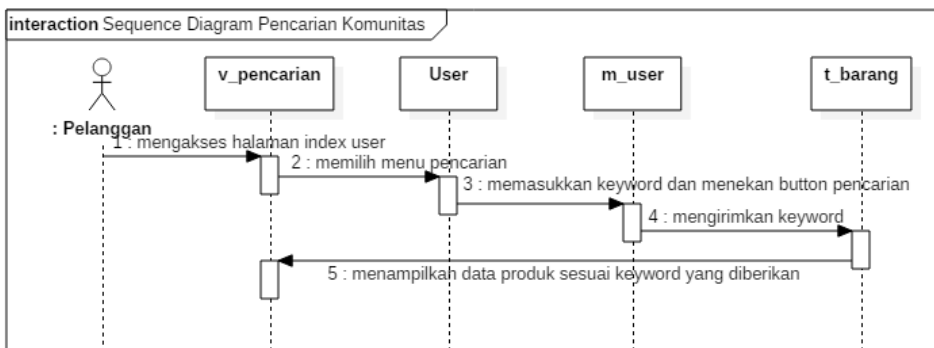
Gambar 4.7 Sequence Kelola Produk

Keterangan :

1. Penjual akan mengakses halaman produk
2. Penjual akan memilih menu tambah produk
3. Penjual akan melakukan *input* data produk dan menekan button simpan
4. *Model* barang akan memproses data yang dikirim dan akan di validasi
5. *Controller* mengirimkan *query create*
6. *Model* barang akan melakukan eksekusi terhadap *query create* tersebut kedalam tabel barang
7. Data produk berhasil di simpan admin dan penjualan dapat melihat pada halaman produk

8. Penjual dan Admin memilih *button* hapus
9. Perintah hapus akan di proses
10. *Controller* akan mengirimkan *id* yang akan di hapus
11. Data berhasil di hapus

5). *Sequence Pencarian Komunitas*

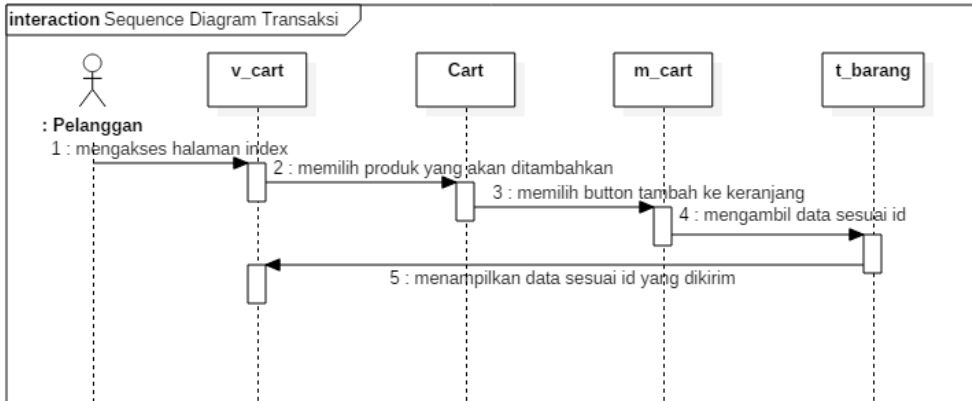


Gambar 4.8 *Sequence Pencarian Komunitas*

Keterangan :

1. Pelanggan akan mengakses halaman *index*
2. Pelanggan akan memilih menu pencarian
3. Pelanggan menginputkan *keyword* komunitasnya dan menekan *button* pencarian
4. *Model user* akan mengirimkan *keyword* kepada tabel barang
5. Tabel barang akan menampilkan data barang sesuai *keyword* yang diberikan

6). Sequence Transaksi

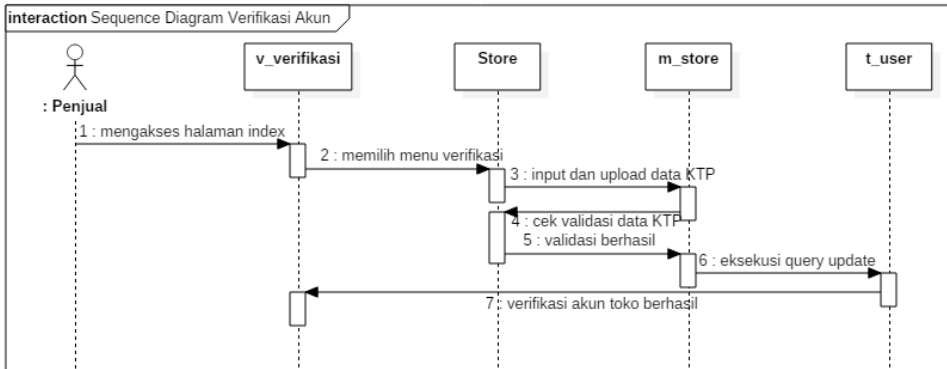


Gambar 4.9 Sequence Transaksi

Keterangan :

1. Pelanggan akan mengakses halaman *index*
2. Pelanggan akan memilih produk yang akan ditambahkan kedalam keranjang belanja
3. Pelanggan akan memilih *button* tambah ke keranjang berdasarkan *id* produk yang dikirim
4. *Mode cart* akan mengambil data dari tabel barang sesuai *id* yang dikirim
5. Produk berhasil ditambahkan kedalam *cart*

7). Sequence Verifikasi Akun

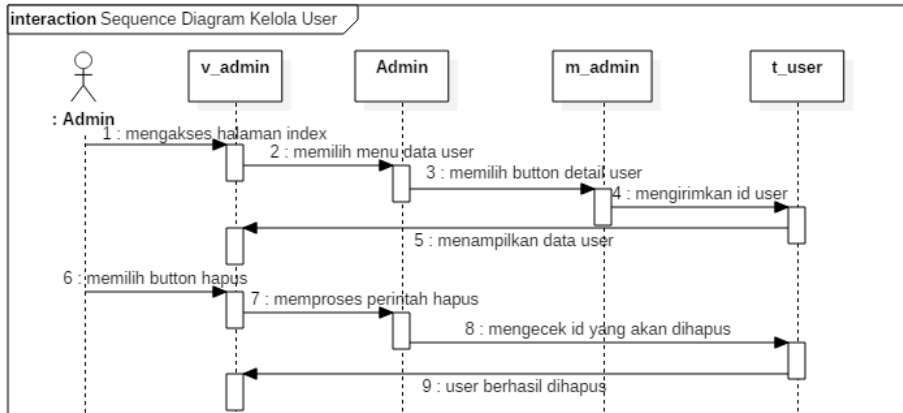


Gambar 4.10 Sequence Verifikasi Akun

Keterangan :

1. Penjual akan mengakses halaman *index*
2. Penjual akan memilih menu verifikasi
3. Penjual akan melakukan *input* nomor KTP dan melakukan *upload* KTP
4. *Controller store* akan melakukan validasi terhadap data KTP tersebut
5. Validasi berhasil akan dikirimkan kepada *model store*
6. *Model Store* akan mengeksekusi *query update* kepada tabel *user*
7. Verifikasi akan toko berhasil

8). Sequence Diagram Kelola User



Gambar 4.11 Sequence Diagram Kelola User

Keterangan :

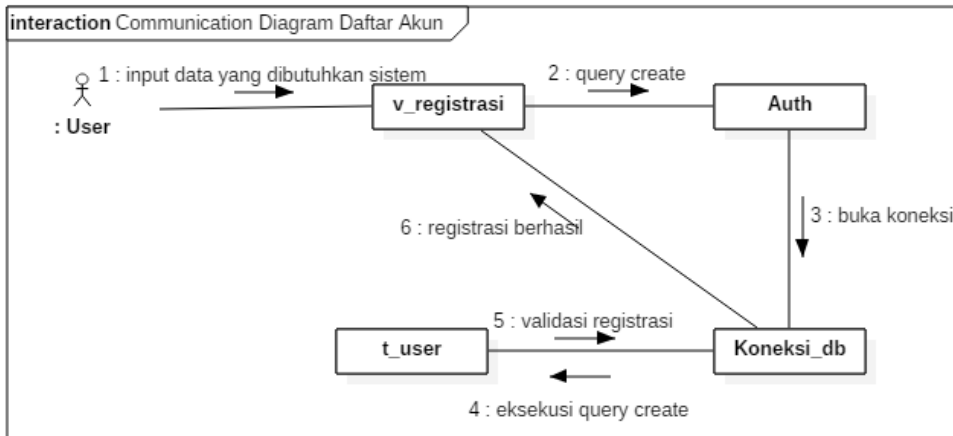
1. Admin akan mengakses halaman *index*
2. Admin akan memilih menu data *user*
3. Admin akan memilih *button* detail *user*
4. *Model admin* akan mengirimkan *id user* kepada tabel *user*
5. Tabel *User* akan menampilkan data *user*
6. Admin memilih *button* hapus
7. *Controller* akan memproses perintah hapus
8. *Controller* akan melakukan cek *id* yang akan dihapus
9. *User* berhasil di hapus

4.5 Collaboration Diagram Sistem JURAGAN

Collaboration diagram yaitu dimana sebuah pengelompokkan pesan terhadap kumpulan diagram sekuen menjadi sebuah diagram. Pada *collaboration* diagram dimana terdapat *method* yang dijalankan antara objek yang satu dan objek lainnya. Dalam *collaboration* diagram tersebut diaman objek harus melakukan sinkronasi pesan dengan serangkaian pesan

– pesan lainnya. Adapun *Collaboration* diagram pada sistem JURAGAN adalah sebagai berikut :

1). *Collaboration* Daftar Akun

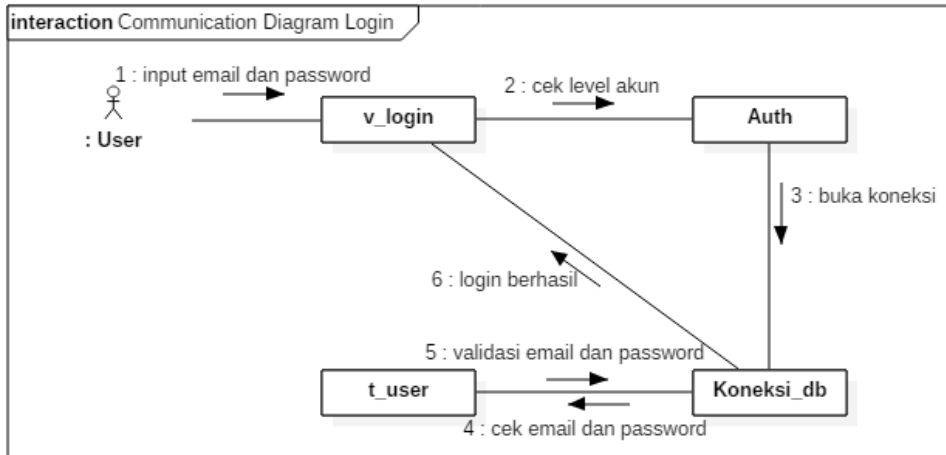


Gambar 4.12 *Collaboration* Daftar Akun

Keterangan :

1. Pelanggan dan Penjual akan melakukan *input* data yang dibutuhkan.
2. *View registrasi* akan membuat *query create* melalui *controller Auth*.
3. *Controller Auth* akan membuka koneksi ke *database*.
4. Eksekusi *query create* akan di lakukan kepada tabel *user*.
5. Validasi *registrasi* akan di lakukan oleh tabel *user*.
6. *Registrasi* berhasil.

2). Collaboration Login

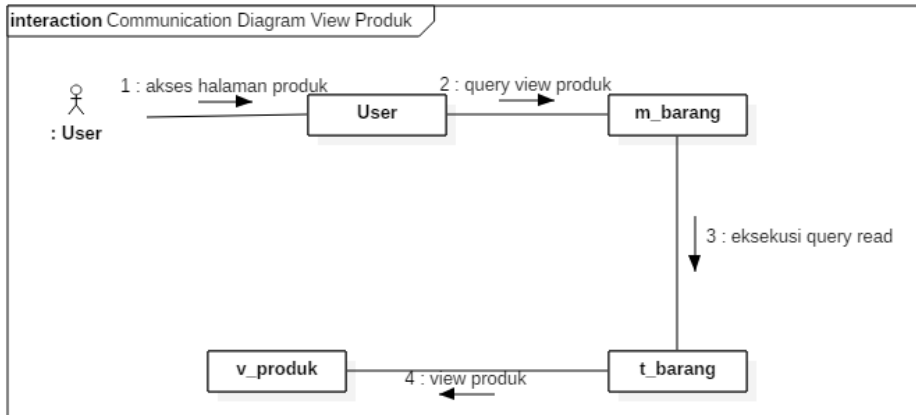


Gambar 4.13 Collaboration Login

Keterangan :

1. Pelanggan dan Penjual akan melakukan *input email* dan *password*.
2. Sistem akan melakukan cek level akun pada *controller Auth*.
3. *Controller Auth* akan membuka koneksi *database*.
4. Di dalam *database* akan dilakukan proses pengecekan *email* atau *password* yang sudah terdaftar.
5. *Database* juga akan melakukan *validasi* terhadap tabel *user*.
6. Apabila lolos maka *user* berhasil melakukan *login*.

3). Collaboration Kelola Produk

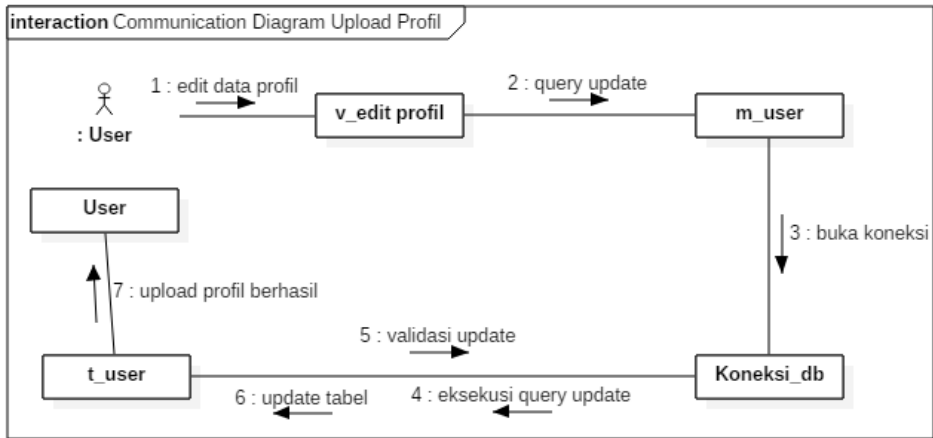


Gambar 4.14 Collaboration Kelola Produk

Keterangan :

1. *User* akan melakukan akses terhadap halaman produk.
2. *Controller* auth akan menampilkan halaman produk dan mengirimkan *query view* produk.
3. Pada model barang dimana *query view* tersebut di eksekusi kepada tabel barang.
4. Tabel barang akan melakukan *view* produk ke halaman *view* produk

4). Collaboration Kelola Profil

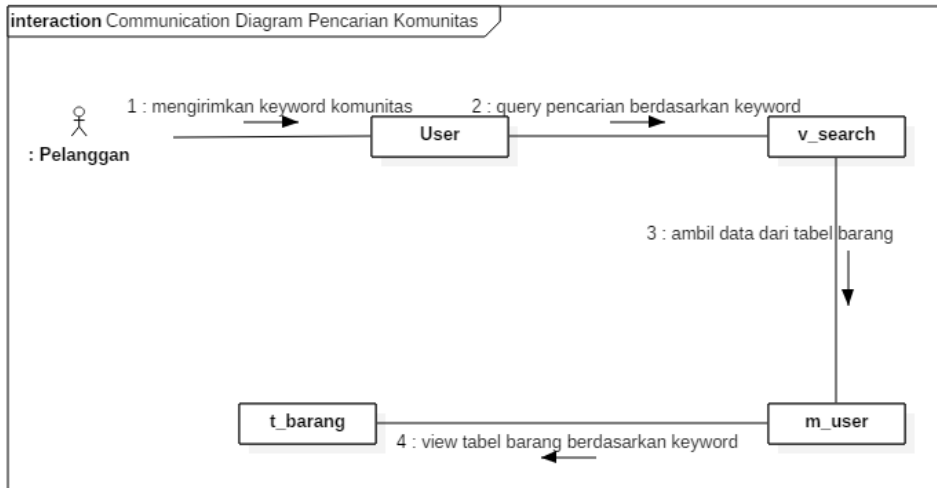


Gambar 4.15 Collaboration Kelola Profil

Keterangan :

1. *User* akan mengakses halaman *edit* profil dan merubah profil mereka pada halaman tersebut.
2. *User* akan menekan *button edit* dan *query update* akan dikirim kepada model *user*.
3. *Model user* akan melakukan akses terhadap *database*.
4. *Database* akan melakukan eksekusi *query update* kepada tabel *user*.
5. Tabel *user* akan melakukan validasi.
6. *Database* akan merubah data pada tabel *user* melalui *query update*.
7. Data akan tersimpan pada tabel *user* dan *update* profil berhasil.

5). Collaboration Pencarian Komunitas

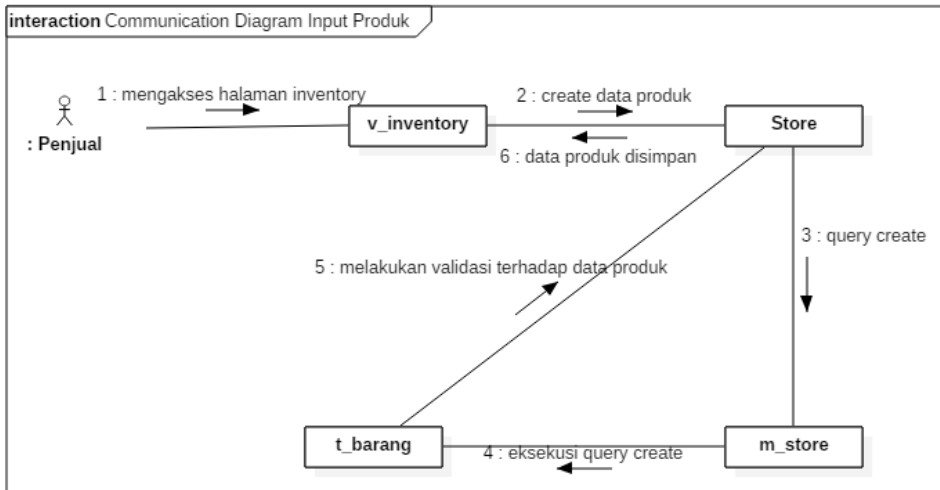


Gambar 4.16 Collaboration Pencarian Komunitas

Keterangan :

1. Pelanggan akan mengirimkan *keyword* komunitasnya.
2. Pada *controller user* dimana akan mengirimkan *query view* dengan menambahkan *keyword* yang di cari.
3. *Model user* akan mengambil data dari tabel barang.
4. Tabel barang akan melakukan *view* sesuai *keyword* yang diberikan.

6). Collaboration Kelola Produk

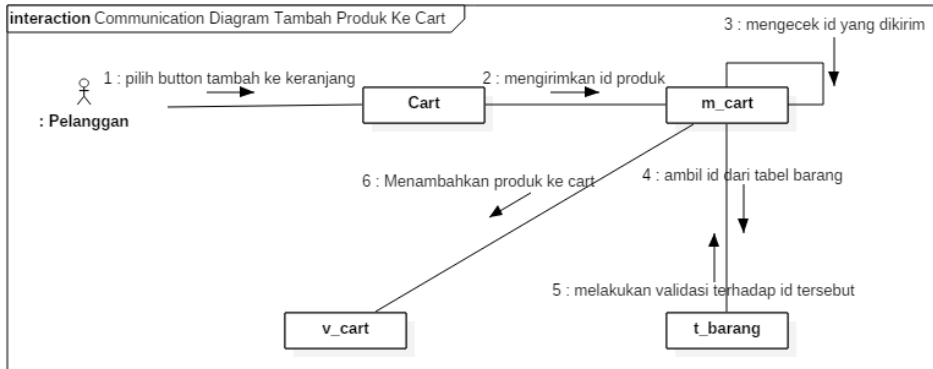


Gambar 4.17 Collaboration Kelola Produk

Keterangan :

1. Penjual akan melakukan akses terhadap halaman *inverntory*.
2. Pada halaman *inventory* tersebut pelanggan akan melakan *create* produk.
3. *Controller* akan melakukan mengirimkan *query create* kepada model *store*.
4. *Query create* tersebut akan dieksekusi oleh model *store* kepada *tabel barang*.
5. *Controller Store* akan melakukan validasi terhadap data produk yang dikirim.
6. Produk akan tersimpan di tabel barang dan ditampilkan pada halaman *inventory*.

7). Collaboration Transaksi

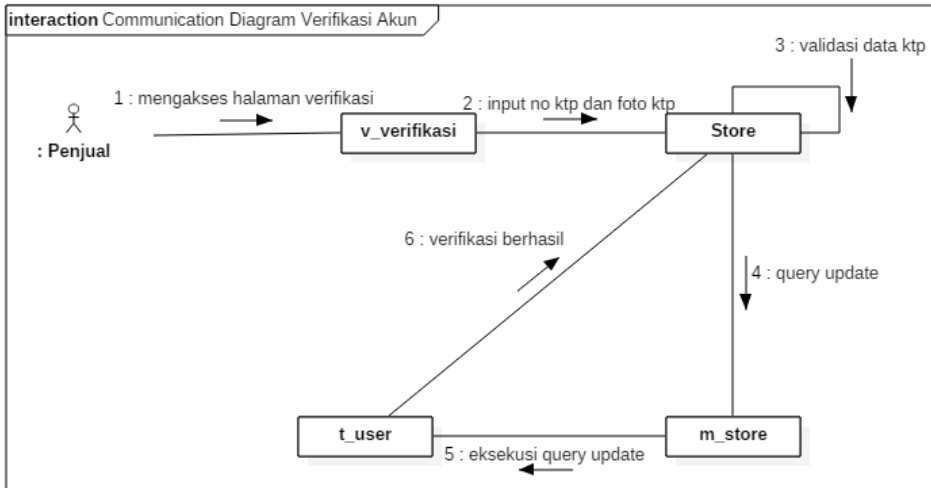


Gambar 4.18 Collaboration Transaksi

Keterangan :

1. Pelanggan akan menekan *button* tambah ke keranjang pada sebuah produk yang *id* dari produk tersebut akan dikirim.
2. *Controller cart* akan mengirimkan *id* ke dalam *model cart*.
3. *Model cart* akan mengecek *id* yang dikirim.
4. *Model cart* akan mengambil *id* tersebut dari tabel barang
5. Validasi pada *id* tersebut akan dilakukan.
6. Apabila lolos maka *model cart* akan menambahkan produk tersebut ke halaman *cart*.

8). Collaboration Verifikasi Akun



Gambar 4.19 Collaboration Verifikasi Akun

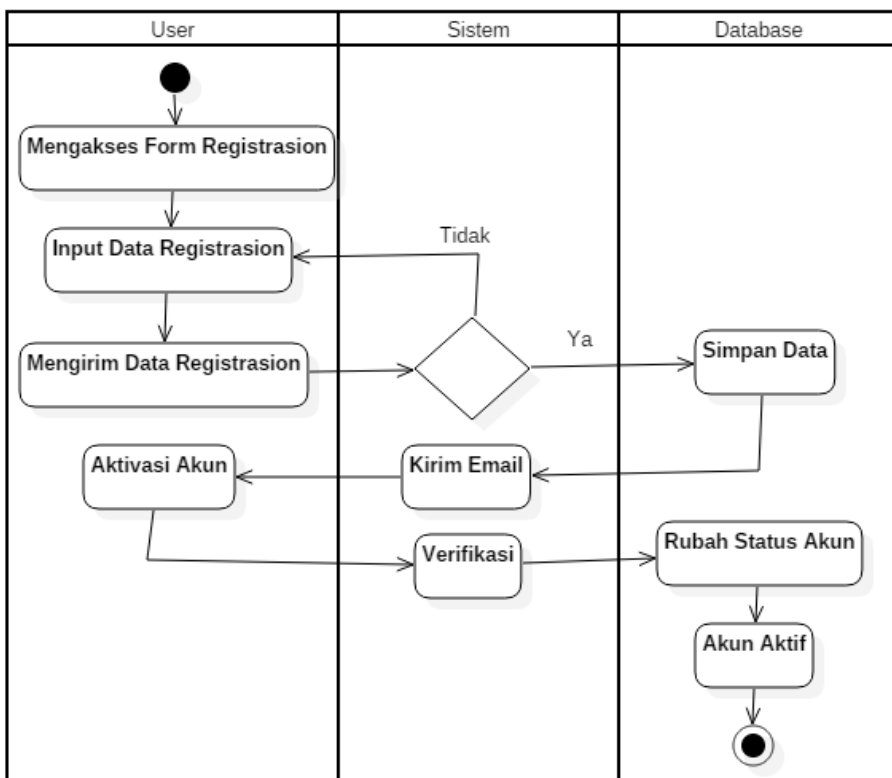
Keterangan :

1. Penjual akan mengakses halaman verifikasi.
2. Pada halaman verifikasi dimana penjual perlu mengisi No KTP dan melakukan *upload* foto KTP.
3. Pada *controller store* dimana akan terjadinya proses validasi data KTP.
4. *Controller Store* akan mengirimkan *query update* kepada *model store*.
5. *Model store* akan melakukan eksekusi *query update* terhadap tabel *user*.
6. Verifikasi akun toko telah berhasil dilakukan.

4.6 Activity Diagram Sistem JURAGAN

Secara umum *activity* diagram merupakan gambaran alur dari suatu sistem yang dibuat, sehingga pengguna mengerti kegunaan sistem yang akan dibangun. *Activity* diagram merupakan gambaran *workflow* atau aktivitas dari sebuah sistem dalam proses bisnis.

1) Activity Diagram Registrasion

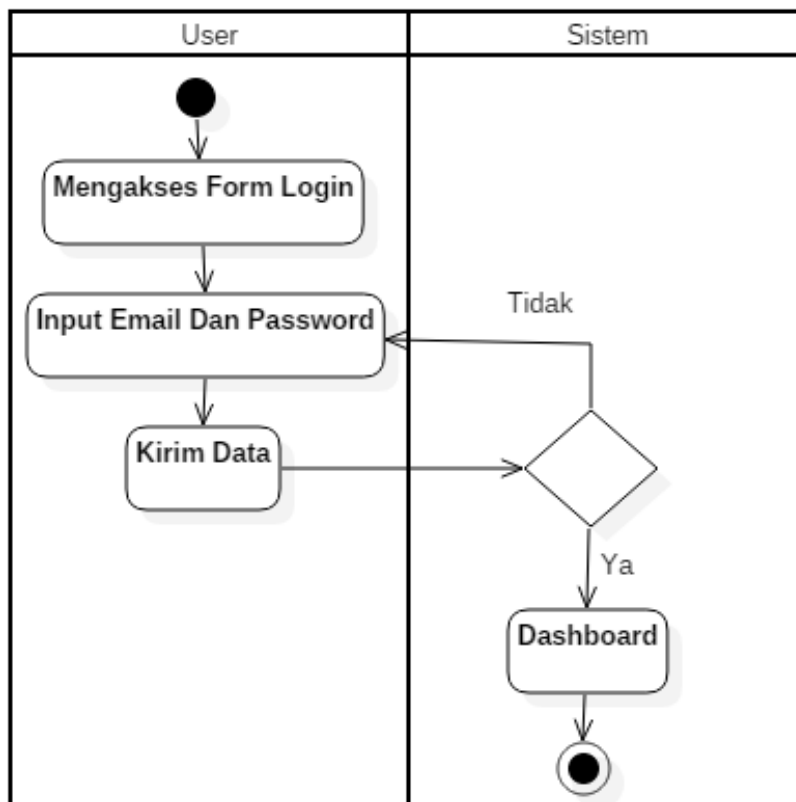


Gambar 5.24 Activity Diagram Registrasion

Pada *activity* diagram tersebut dimana dijelaskan bahwa *user* akan mengakses terlebih dahulu *form registrasion*. Setelah *form registrasion* terbuka maka langkah selanjutnya dimana *user* akan melakukan *input data registrasion* pada *form* tersebut sesuai dengan

ketentuan yang diberikan oleh sistem. Data yang di *nput* kan tersebut akan dikirim kepada sistem oleh *user*. Sistem akan melakukan proses validasi dan apabila berhasil maka data akan di simpan kedalam *database* namun akun belum aktif sehingga *user* tidak dapat melakukan *login*. Untuk mengaktifkan akun tersebut dimana *user* perlu melakukan aktivasi akun pada *email* yang telah dikirimkan oleh sistem. Sistem akan melakukan verifikasi. Data yang berada pada *database* akan di *update* sehingga status akun berubah menjadi aktif.

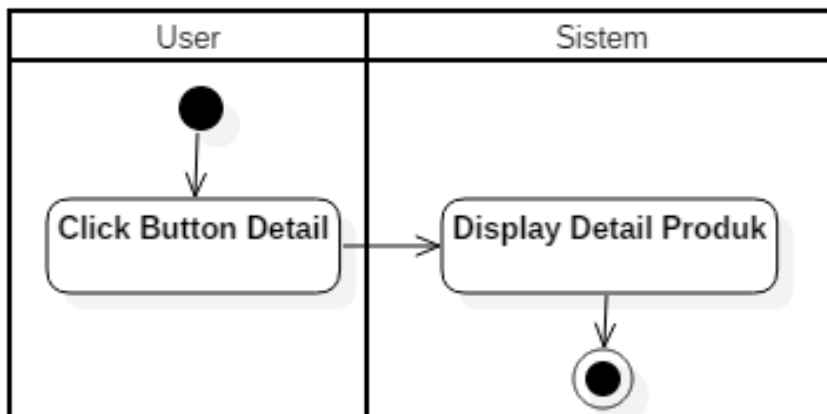
2) Activity Diagram Login



Gambar 5.25 Activity Diagram Login

Pada *activity* diagram tersebut dimana *step* pertama yang *user* lakukan adalah mengakses *form login*. Ketika *form login* sudah terakses maka *step* berikutnya *user* akan melakukan *input email* dan *password* yang sudah terdaftar pada sistem. Data *email* dan *password* yang sudah di *input* kan tersebut akan dikirimkan kepada sistem. Dimana sistem akan melakukan proses validasi dan apabila berhasil maka sistem akan mengarahkan *user* ke dalam *dashboard* yang menandakan bahwa *login* telah berhasil di lakukan.

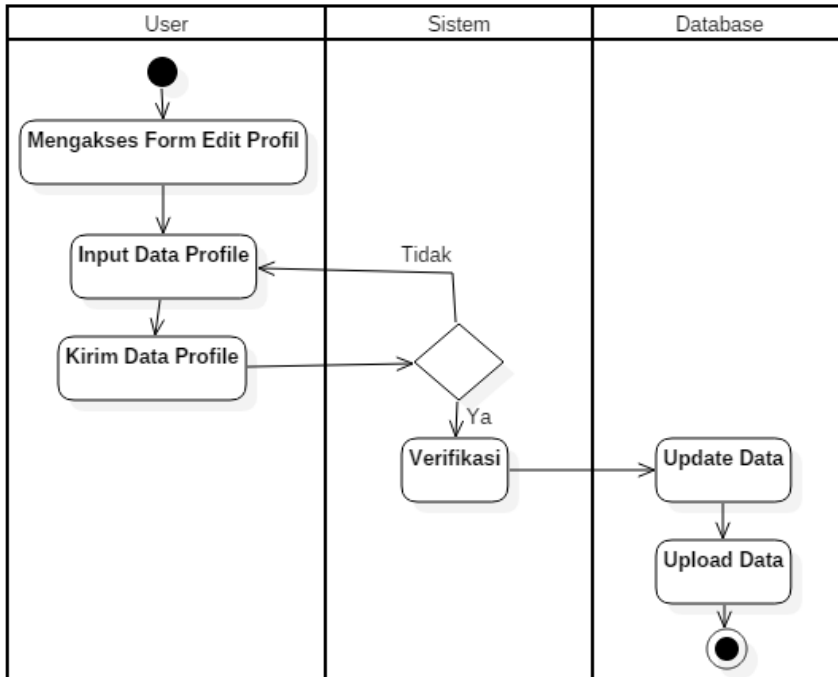
3) Activity Diagram View Produk



Gambar 5.26 Activity Diagram View Produk

Pada *activity* diagram tersebut dimana menunjukkan aktivitas *user* dalam melakukan *view* produk. Hal pertama yang harus *user* lakukan adalah memberikan *action* terhadap *button* detail dengan cara meng *click button* tersebut. Ketika *button* tersebut di *click* dimana sistem akan mengirim *id* dari produk tersebut dan memunculkan *display* detail produk. Dalam *display* detail produk dimana akan menampilkan informasi – informasi secara detail pada produk tersebut.

4) Activity Diagram Edit Profile

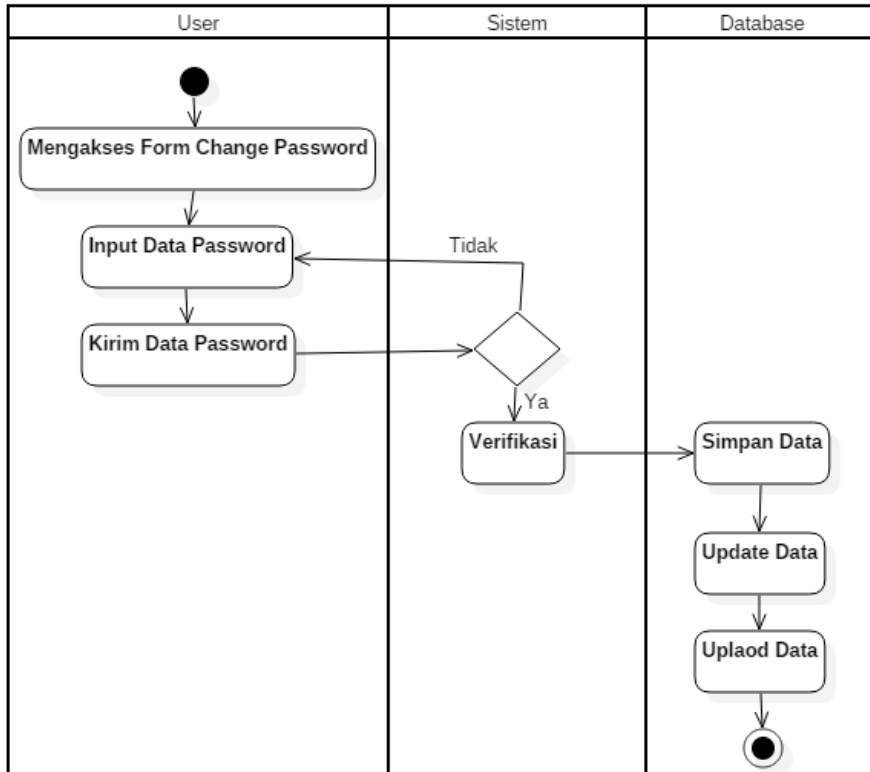


Gambar 5.27 Activity Diagram Edit Profile

Pada *activity diagram* tersebut dimana menunjukkan sebuah aktifitas *edit profile*. Dimana *step* pertama dalam melakukan aktifitas *edit profile user* perlu mengakses *form edit profile*. Pada *form edit profile* tersebut *user* akan melakukan perubahan data *profile*-nya. Ketika data sudah dirubah maka *user* akan mengirimkan data tersebut kepada sistem. Sistem akan melakukan proses validasi dimana untuk mengecek data tersebut apakah sudah benar. Jika data sudah benar maka sistem akan melakukan verifikasi. Setelah data terverifikasi sistem akan mengirimkan data tersebut kepada *database*. Di dalam *database* dimana terjadi fungsi *get* yang dimana untuk melakukan

update data. Data yang sudah di *update* akan di *upload* agar *user* dapat melihat perubahannya.

5) Activity Diagram Change Password

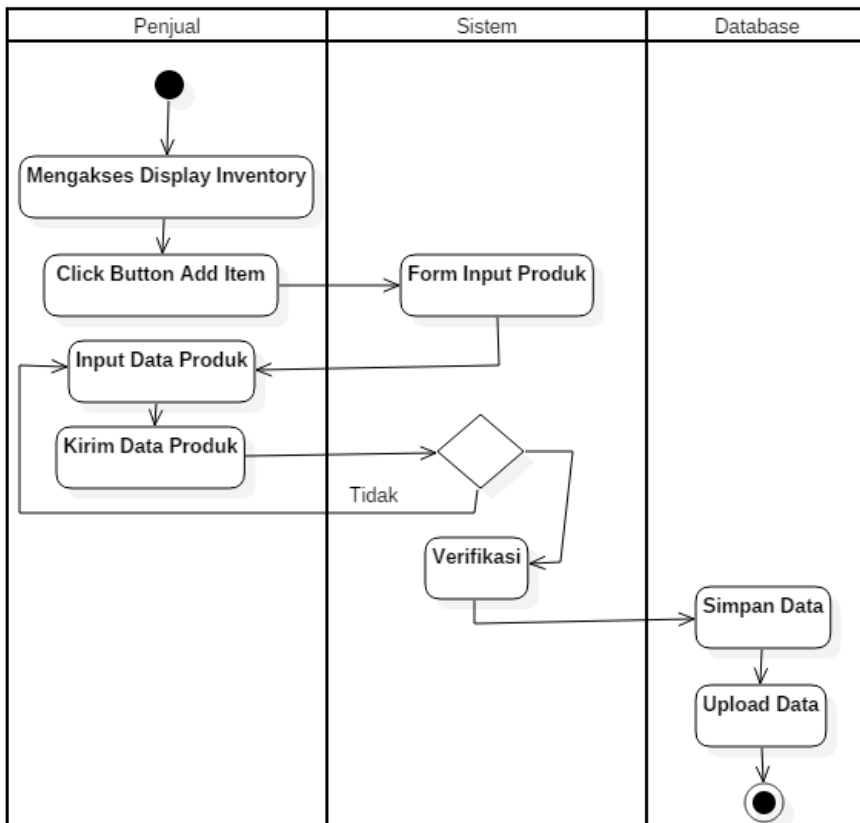


Gambar 5.28 Activity Diagram Change Password

Pada *activity* diagram tersebut dimana menjelaskan mengenai aktifitas dalam melakukan *change password*. *Step* pertama yang perlu *user* lakukan adalah mengakses *form change password*. Pada *form* tersebut dimana dibutuhkan data berupa *password* lama dan *password* baru yang dimana *user* perlu melakukan *input* data tersebut. Data yang di *input* kan tersebut akan dikirimkan kepada sistem oleh *user*. Sistem akan melakukan validasi untuk mengecek data apakah sudah benar.

Apabila sudah benar sistem akan memverifikasi data tersebut. Data yang sudah terverifikasi akan dikirimkan ke dalam *database*. Pada *database* akan dilakukan sebuah fungsi *get* yang berguna untuk melakukan *update* data. Data yang sudah di *update* akan di *upload* oleh *database*.

6) Activity Diagram Input Produk

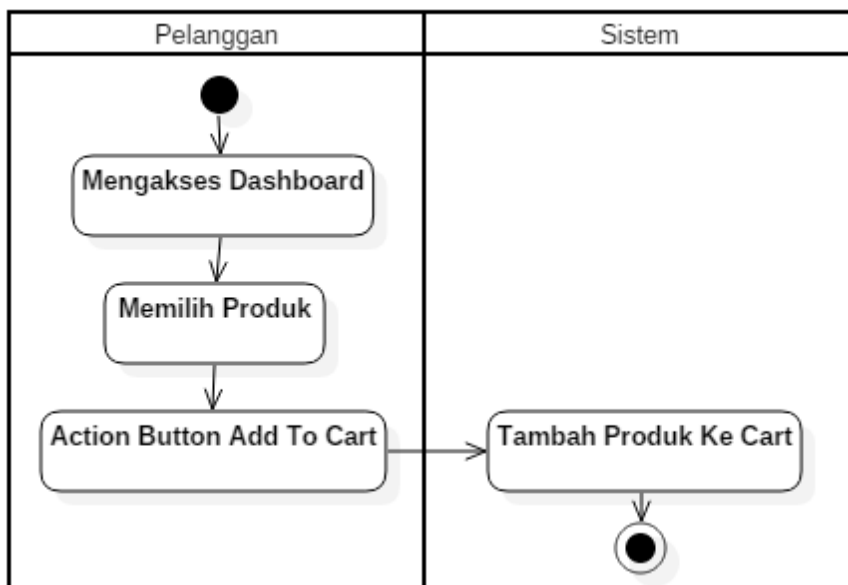


Gambar 5.29 Activity Diagram Input Produk

Pada *activity* diagram tersebut dimana menjelaskan bagaimana aktifitas pada proses *input* data produk dan aktor yang berperan adalah *user* dengan level penjual. Hal pertama yang harus dilakukan oleh penjual dimana perlu mengakses *display inventory*. Pada *display*

inventory terdapat satu *button* dengan nama *add item* dan apabila diberikan *action* akan memunculkan *form input produk* yang diberikan dengan fungsi modal. Pada *form* tersebut dimana penjual akan melakukan *input* data produk yang dibutuhkan oleh sistem. Data yang di *input* kan tersebut akan dikirim kepada sistem. Sistem akan melakukan proses validasi guna mengecek data tersebut apakah sudah benar. Apabila data yang dikirim tidak ada masalah maka sistem akan melakukan verifikasi. Data yang sudah terverifikasi akan disimpan kedalam *database*. Pada *database* akan berjalan fungsi *PUT* yang dimana berguna untuk meng-*input*-kan data. Data yang sudah ter *input* akan di *upload* oleh *database* agar tampil pada *Display My Inventory*.

7) Activity Diagram Tambah Produk Ke Cart



Gambar 5.30 Activity Diagram Tambah Produk Ke Cart

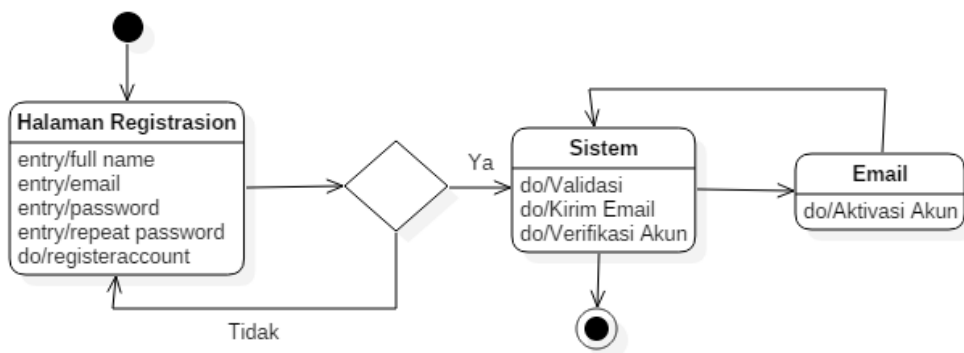
Pada *activity* diagram tersebut dimana menjelaskan bagaimana aktifitas dalam melakukan penambahan data ke *cart* dan aktor yang

berperan adalah *user* dengan level *pelanggan*. Hal pertama yang harus dilakukan pelanggan adalah mengakses *dashboard* terlebih dahulu. Dengan begitu pelanggan dapat memilih produk yang diinginkan. Pelanggan juga dapat melihat informasi – informasi produk tersebut dengan menekan *button* detail. Sistem akan memunculkan *display* detail produk. Apabila pelanggan merasa cocok dengan produk tersebut maka pelanggan dapat menambahkan produk tersebut kedalam *cart* dengan cara meng *click button add to cart*. Sistem akan secara otomatis menambahkan produk tersebut kedalam *cart*.

4.7 Statechart Diagram Sistem JURAGAN

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari suatu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimulus yang diterima. *Statechart* diagram mendeskripsikan bagaimana suatu objek mengalami perubahan status adanya *trigger* dari *event-event*. Adapun *statechart* diagram pada sistem JURAGAN adalah sebagai berikut :

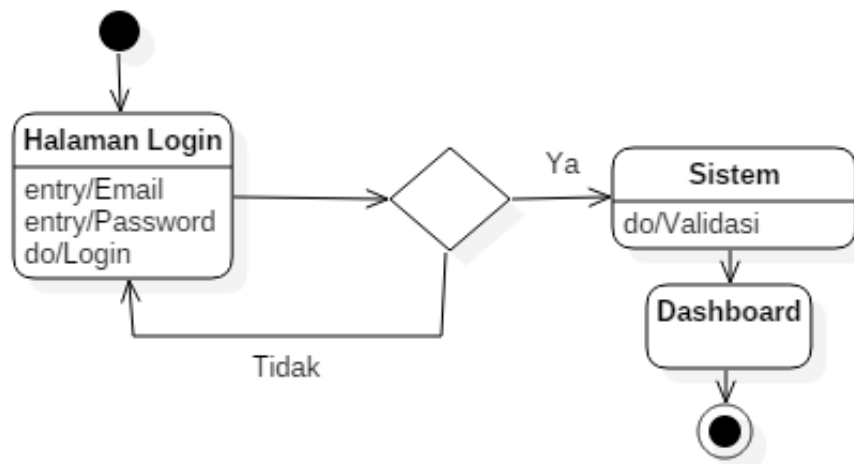
1) *Statechart* Diagram Registrasion



Gambar 5.31 Statechart Diagram Registrasion

Pada *statechart* diagram tersebut dimana menjelaskan *state – state* yang dilakukan oleh *user* untuk *registrasion* akun. *State* pertama yang perlu dilakukan adalah dengan mengakses halaman *registrasion* terlebih dahulu. Pada halaman *registrasion* dimana *user* perlu melakukan *input* data berupa *full name*, *email*, *password*, dan *repeat password*. Ketika data sudah di *input* kan maka *user* perlu memberikan *action* terhadap *button register accouunt*. *State* kedua dimana sistem akan melakukan proses validasi dan mengirimkan email kepada *user*. *State* ketiga dimana *user* akan mengakses *email* dan melakukan aktivasi akun. *State* keempat dimana sistem akan melakukan verifikasi akun sehingga akun menjadi aktif.

2) *Statechart* Diagram Login



Gambar 5.32 *Statechart* Diagram Login

Pada *statechart* diagram tersebut dimana menjelaskan *state – state* yang perlu dilakukan dalam proses *login*. Dimana untuk melakukan *login state* pertama ialah dengan mengakses halaman *login* terlebih dahulu. Pada halaman *login* dimana *user* perlu melakukan *input email*

dan *password* yang sudah terdaftar pada sistem. Ketika data sudah di *input* maka *user* perlu memberikan *action* terhadap *button login*. *State* kedua dimana sistem akan melakukan validasi guna melakukan pengecekan terhadap data yang dikirim. Apabila data yang dikirim tidak terjadi masalah maka sistem akan melanjutkan ke proses berikutnya. *State* ketiga dimana sistem akan memunculkan *dashboard* yang menunjukkan bahwa *login* telah sukses.

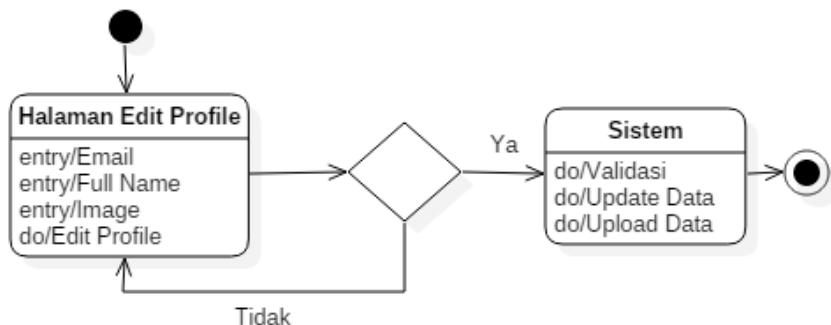
3) *Statechart Diagram View Produk*



Gambar 5.33 *Statechart Diagram View Produk*

Pada *statechart* diagram tersebut dimana menunjukkan *state – state* dalam melakukan *view* produk. *State* pertama yang perlu dilakukan adalah dengan memberikan *action* terhadap *button detail*. *State* kedua yaitu dimana sistem akan memberikan respon dengan cara memunculkan halaman detail produk.

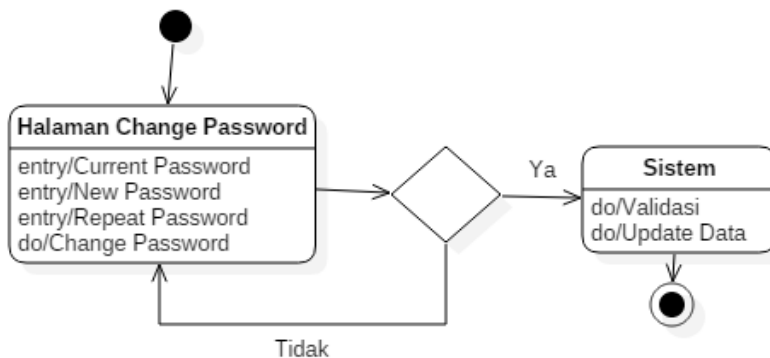
4) *Statechart Diagram Edit Profile*



Gambar 5.34 *Statechart Diagram Edit Profile*

Pada *statechart* diagram tersebut dimana menunjukkan *state – state* dalam melakukan *edit profile*. *State* pertama yang perlu dilakuka *user* adalah dengan mengakses halaman *edit profile*. Dalam halaman *edit profile* ersebut dimana *user* perlu menginputkan data *full name* dan *image*. *Email* disini bersifat *read only* sehingga tidak dapat diubah. Jika data tersebut sudah di *input* kan dimana *user* perlu memberikan *action* terhadap *button edit profile*. *State* kedua dimana sistem akan melakukan proses validasi guna mengecek data apakah sudah sesuai dengan ketentuan sistem. Apabila data sudah sesuai maka data akan dikirim ke dalam *database* untuk di *update*. Ketika sudah di *update* maka data tersebut akan di *upload* agar dapat dilihat perubahannya oleh *user*.

5) *Statechart Diagram Change Password*

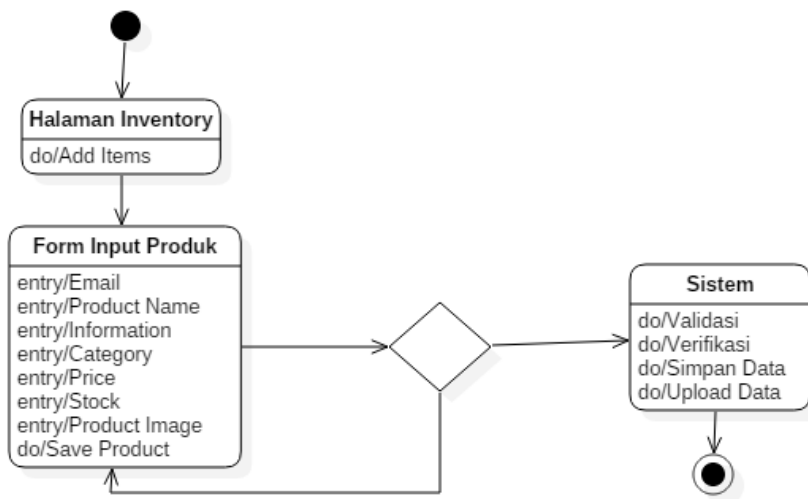


Gambar 5.35 Statechart Diagram Change Password

Pada *statechart* diagram tersebut dimana menunjukkan *state – state* dalam melakukan *change password*. *State* pertama yang perlu dilakukan adalah dengan mengakses halaman *change password*. Pada halaman tersebut dimana *user* perlu melakukan *input* data *current password*, *new password*, dan *repeat password*. Ketika data sudah di

input kan dimana *user* perlu memberikan *action* terhadap *button change password*. *State* kedua yaitu dimana sistem akan melakukan proses validasi guna mengecek data apakah sudah sesuai dengan ketentuan sistem. Apabila data sudah sesuai maka sistem akan mengirim data tersebut ke dalam *database*. *Database* akan melakukan proses *get* dimana berfungsi untuk *update* data.

6) *Statechart Diagram Input Produk*

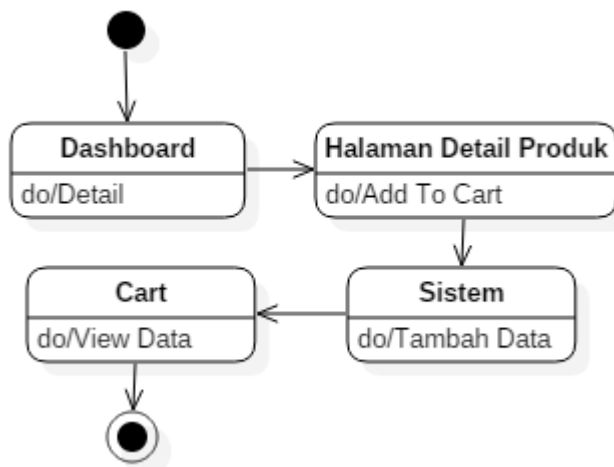


Gambar 5.36 Statechart Diagram Input Produk

Pada *statechart* diagram tersebut merupakan *state – state* dalam melakukan proses *input produk* dimana aktor yang berperan adalah *user* dengan level penjual. *State* pertama yang perlu dilakukan adalah dengan mengakses halaman *inventory* terlebih dahulu, dimana pada halaman tersebut terdapat satu *button add item* yang perlu di berikan *action*. *State* kedua yaitu sebuah modal berupa *form input product* akan muncul dan penjual perlu melakukan *input* data berupa *email*, *product name*, *information*, *category*, *price*, *stock*, dan *product image*. Ketika

data sudah di *inputkan* dimana penjual perlu memberikan *ation* terhadap *button save produt*. *State* ketiga dimana sistem akan melakukan proses validasi guna mengecek data apakah sudah sesuai. Apabila data sudah sesuai maka sistem akan melakukan verifikasi data tersebut dan menyimpannya kedalam *database*. Pada *database* dimana akan berjalan fungsi *PUT* yang berperan dalam meng *input* kan data dan data yang sudah ter *input* akan di *upload* oleh *database*.

7) *Statechart* Diagram Tambah Produk Ke *Cart*



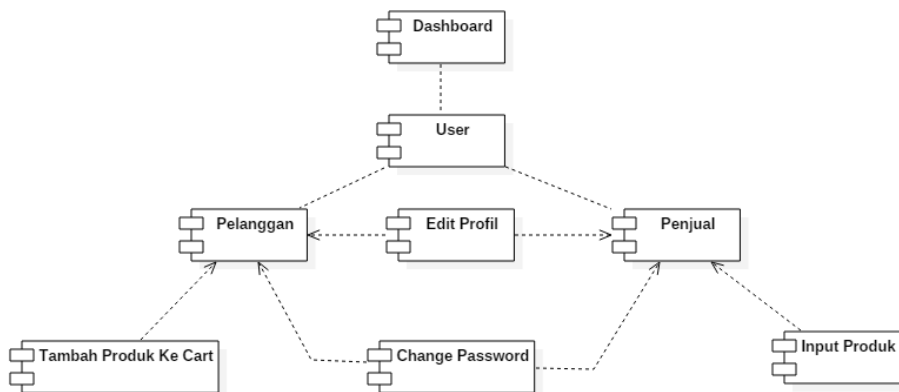
Gambar 5.37 Statechart Diagram Tambah Produk Ke Cart

Pada *statechart* diagram tersebut dimana akan menjelaskan *state – state* yang dilalui dalam proses menambahkan priduk ke dalam *cart* yang dimana aktor disini adalah *user* dengan level pelanggan. *State* pertama diamana pelanggan perlu mengakses *dashboard* terlebih dahulu. Pada *dashboard* pelanggan dapat memilih produk – produk yang telah di *input* kan oleh penjual. Untuk melihat *informasion* produk tersebut dimana pelanggan dapat memberikan *action* terhadap

button detail. *State* kedua dimana sistem akan merespon dengan memunculkan halaman detail produk. Pada halaman detail produk dimana pelanggan dapat menambahkan produk ke *cart* secara langsung dengan cara memberikan *action* terhadap *button add to cart*. *State* ketiga dimana sistem akan melakukan proses tambah data ke dalam halaman *cart*. *State* keempat dimana produk yang sudah ditambahkan dapat di *view* pada halaman *cart*.

4.8 Component Diagram

Berikut ini merupakan *component* diagram yang menjelaskan semua *component* yang terdapat pada sistes JURAGAN.

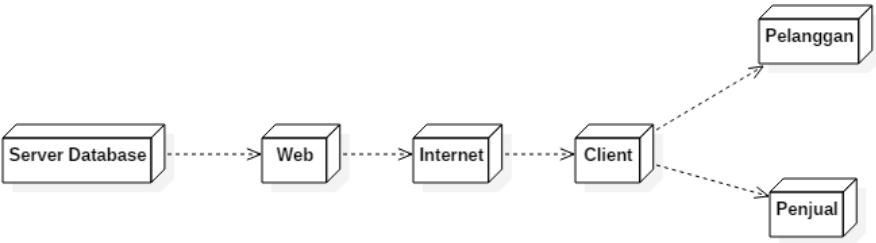


Gambar 5.38 Component Diagram

4.9 Diployment Diagram

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakan bagian – bagian *software* yang berjalan pada bagian – bagian *hardware* yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen – komponen *hardware* tersebut. *Deployment* diagram dapat digunakan pada bagian – bagian awal

proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem. Berikut *deployment* diagram yang menggambarkan susunan fisik, perangkat lunak dan sistem.



Gambar 5.39 Deployment Diagram

4.10 Perancangan Basis Data/Database

Tabel 5.20 User

No.	Field	Type	Wide	Information
1	<i>Id</i>	<i>Integer</i>	11	<i>Id user</i> yang merupakan <i>primary key</i>
2	<i>Name</i>	<i>Varchar</i>	128	Nama lengkap <i>user</i>
3	<i>Email</i>	<i>Varchar</i>	128	<i>Email</i> dari <i>user</i>
4	<i>Password</i>	<i>Varchar</i>	300	<i>Password</i> dari akun <i>user</i>
5	<i>Image</i>	<i>Varchar</i>	128	<i>Image profile</i> pada <i>user</i> pelanggan

6	<i>Role_id</i>	<i>Integer</i>	11	Merupakan id level <i>user</i> . 1. Admin 2. User
7	<i>Is_active</i>	<i>Integer</i>	11	Id akun yang aktif. 0. Belum Aktif 1. Aktif
8	<i>Date_created</i>	<i>Integer</i>	11	Tanggal <i>user</i> mendaftar akun
9	Kebijakan	<i>Varchar</i>	15	Validasi daftar toko
10	Tlpn	<i>Varchar</i>	15	Nomor telpon <i>user</i>
11	Alamat	<i>Varchar</i>	600	Alamat dari <i>user</i>
12	Kecamatan	<i>Varchar</i>	25	Kecamatan <i>user</i>
13	Kelurahan	<i>Varchar</i>	25	Kelurahan <i>user</i>
14	Kota	<i>Varchar</i>	25	Kota <i>user</i>
15	Kode_pos	<i>Varchar</i>	15	Kode pos alamat <i>user</i>
16	Status_toko	<i>Varchar</i>	25	Status verifikasi toko
17	Cat_toko	<i>Text</i>		Berisikan catatan toko

18	No_ktp	<i>Varchar</i>	20	No KTP <i>user</i>
19	Img_ktp	<i>Varchar</i>	25	Foto KTP <i>user</i>

Tabel 5.21 tb_barang

No.	<i>Field</i>	<i>Type</i>	<i>Wide</i>	<i>Information</i>
1	Id_brg	<i>Integer</i>	11	Merupakan id dari barang dan merupakan <i>primary key</i>
2	<i>Email</i>	<i>Varchar</i>	128	<i>Forgent key</i> dari tabel <i>user</i>
3	Nama_brg	<i>Varchar</i>	120	Nama dari produk yang di <i>input</i>
4	Keterangan	<i>Varchar</i>	225	Keterangan dari sebuah produk
5	Kategori	<i>Varchar</i>	60	Kategori dari sebuah produk
6	Harga	<i>Integer</i>	11	Harga dari sebuah produk
7	Stok	<i>Integer</i>	4	Merupakan jumlah dari sebuah produk yang ada

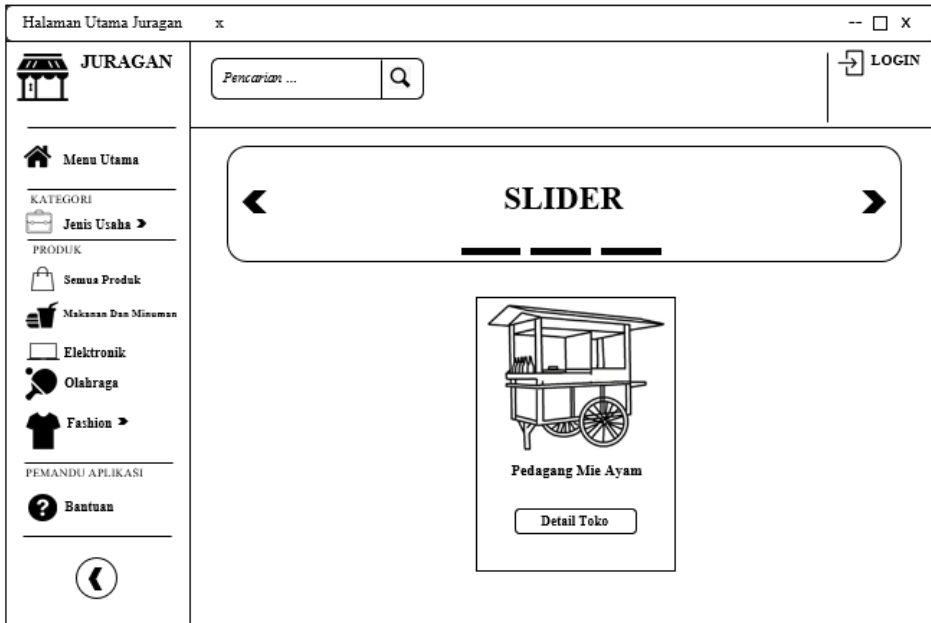
8	Gambar	<i>Varchar</i>	60	Merupakan gambar dari sebuah produk
---	--------	----------------	----	-------------------------------------

Tabel 5. 22 User_token

No.	<i>Field</i>	<i>Type</i>	<i>Wide</i>	<i>Information</i>
1	<i>Id</i>	<i>Integer</i>	11	Merupakan <i>id</i> dari token yang masuk dan sebuah <i>primary key</i>
2	<i>Email</i>	<i>Varchar</i>	128	Merupakan <i>forgent key</i> dari tabel <i>user</i>
3	<i>Token</i>	<i>Varchar</i>	128	Token yang akan dikirim saat aktivasi akun
4	<i>Date_created</i>	<i>Varchar</i>	128	Tanggal token di <i>input</i> kan

4.11 User Interface Sistem

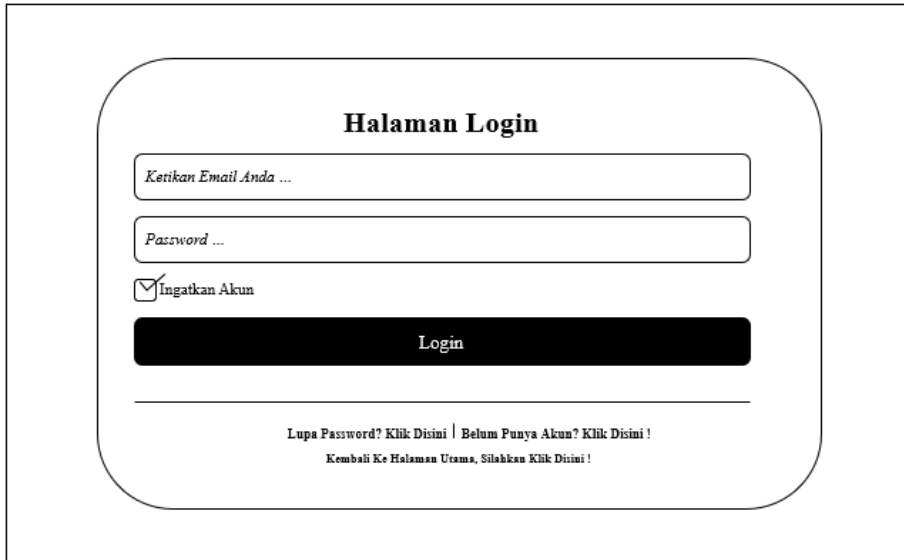
1). User Interface Halaman Utama (Sebelum Login)



Gambar 5.40 UI Halaman Utama Sebelum Login

Pada gambar 5.40 dimana merupakan UI halaman utama dengan kondisi sebelum melakukan *login*. Dalam *topbar* pada UI tersebut terdapat sebuah *form* pencarian yang berguna untuk mencari sebuah toko. Terdapat juga sebuah *icon login* yang disertai dengan *text* dengan fungsi *link*, dan apabila di klik akan menuju ke halaman *login*. Pada bagian *sidebar* terdapat *icon-icon* yang dibutuhkan dalam sistem JURAGAN tersebut yang diberikan dengan fungsi *link*. Pada bagian *index* nya terdapat *slider* yang akan berpindah – pindah secara otomatis. Terdapat juga *card* yang di isi dengan toko – toko yang sudah terdaftar pada sistem JURAGAN tersebut.

2). *User Interface Login*



The image shows a login form titled "Halaman Login" inside a rounded rectangular card. The form contains two input fields: "Ketikan Email Anda ..." and "Password ...". Below these is a checkbox labeled "Ingatkan Akun" with a checked mark. A black button labeled "Login" is positioned below the checkbox. At the bottom of the card, there is a horizontal line followed by the text "Lupa Password? Klik Disini | Belum Punya Akun? Klik Disini !" and "Kembali Ke Halaman Utama, Silahkan Klik Disini !".

Gambar 5.41 UI Login

Pada gambar 5.41 tersebut merupakan UI *login* dari sistem JURAGAN. Dimana UI *login* tersebut di masukkan kedalam *card* dan diberikan sebuah *form* guna menginputkan *email* yang terdaftar dan *password*. Terdapat juga pengingat akun untuk melakukan save pada akun kita. Terdapat juga *buuton login* yang dimana apabila diklik akan memulai proses *login*.

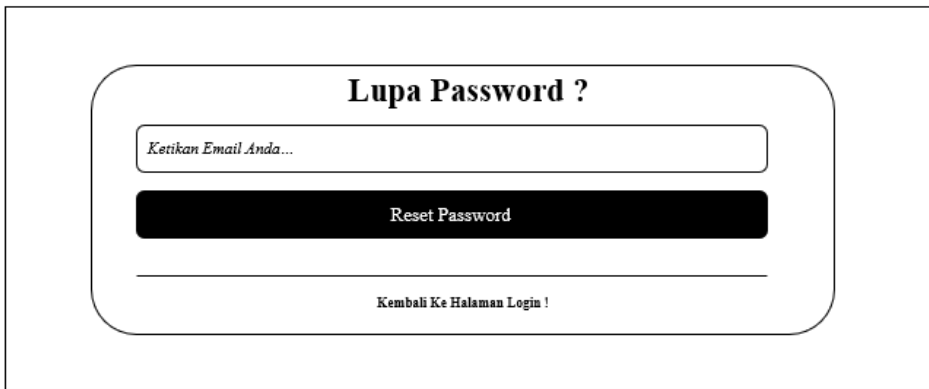
3). *User Interface* Daftar Akun

The image shows a user interface for account registration. It is titled "Daftarkan Akun Anda!". Below the title, there are four input fields: "Nama Lengkap ..." (Full Name), "Email Anda ..." (Your Email), "Password Anda ..." (Your Password), and "Ulangi Password Anda ..." (Repeat Your Password). Below these fields is a large black button labeled "Daftar Akun". At the bottom of the form, there are two links: "Lupa Password ? Klik Disini !" (Forgot Password? Click Here!) and "Sudah Punya Akun ? Silahkan Login !" (Already have an account? Please login!).

Gambar 5.42 UI Daftar Akun

Pada *UI* daftar akun tersebut dimana hanya terdapat satu *layer* yang didalamnya terdiri dari 4 *textbox* yang diberikan fungsi *placeholder*, terdapat juga 1 *button* untuk melakukan *action registrasion*. Di bagian bawah *layer* tersebut terdapat 2 *link* yang yang terdiri dari *Forgot Password* dan *Already have as account ? Login !* yang dimana akan menuju ke halaman lain sesuai *link* tersebut apabila di *click*.

4). *User Interface Lupa Password*

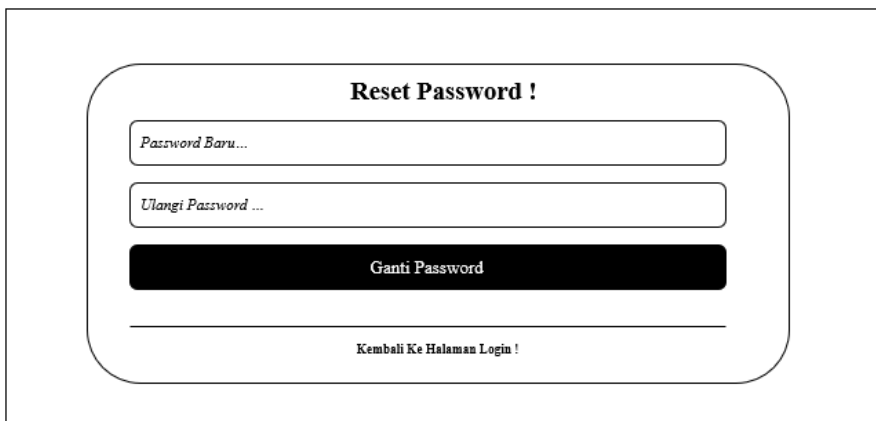


The image shows a user interface for a 'Forgot Password' feature. It is contained within a rounded rectangle. At the top, the title 'Lupa Password ?' is centered. Below the title is a text input field with the placeholder text 'Ketikan Email Anda...'. Underneath the input field is a large, solid black button with the text 'Reset Password' in white. At the bottom of the interface, there is a horizontal line followed by the text 'Kembali Ke Halaman Login !'.

Gambar 5.43 UI Forgot Password

Pada *UI Forgot Password* dimana hanya terdiri dari satu *layer* utama yang di dalamnya terdapat 1 *textbox* dengan menggunakan fungsi *placeholder*, terdapat juga 1 *button* untuk melakukan *action reset password*. Pada bagian bawah *layer* terdapat satu *link Back to login* yang dimana akan berpindah ke halaman *login* apabila di *click*.

5). *User Interface Reset Password*

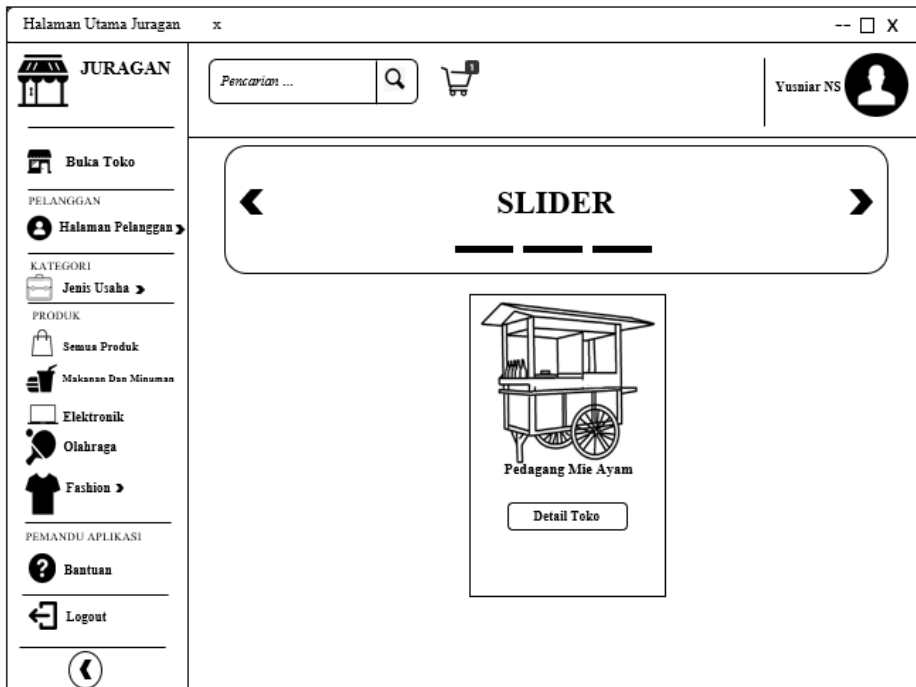


The image shows a user interface for a 'Reset Password' feature. It is contained within a rounded rectangle. At the top, the title 'Reset Password !' is centered. Below the title are two text input fields. The first input field has the placeholder text 'Password Baru...'. The second input field has the placeholder text 'Ulangi Password ...'. Underneath the input fields is a large, solid black button with the text 'Ganti Password' in white. At the bottom of the interface, there is a horizontal line followed by the text 'Kembali Ke Halaman Login !'.

Gambar 5.44 UI Reset Password

Pada *UI Reset Password* tersebut hanya terdapat satu *layer* utama yang dimana di dalamnya terdapat 2 *textbox* dengan fungsi *placeholder*, terdapat juga 1 *button* untuk melakukan *action changed password*. Di bagian bawah *layer* terdapat 1 *link* untuk menuju ke halaman *login* apabila di *click*.

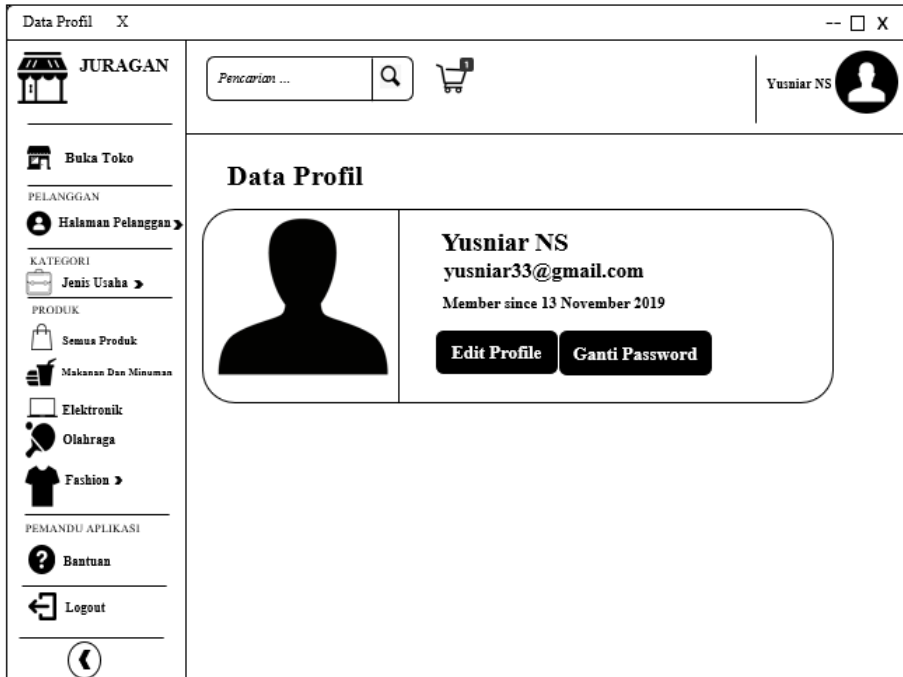
6). *User Interface* Halaman Utama Sesudah Login



Gambar 5.45 UI Halaman Utama Sesudah Login

Pada gambar 4.45 merupakan UI pada halaman utama sistem JURAGAN dengan kondisi sesudah *login*, dimana bedanya dengan gambar 5.40 UI pada gambar 4.45 ini memberikan akses yang lebih luas yaitu dengan adanya *icon-icon* yang bertambah seperti buka toko dan halaman pelanggan pada *sidebar* dan juga nama *user* beserta foto profil dan *icon cart* pada *topbar*.

7). User Interface Data Profil



Gambar 5.46 UI Data Profil

Pada gambar 5.46 merupakan UI dari data profil dimana untuk bagian *sidebar* dan *topbar* akan terlihat sama dengan gambar 5.45. Pada bagian index nya terdapatsebuah *card* yang didalamnya diberikan informasi – informasi data profil *user*.

8). User Interface Edit Profil

The screenshot shows a web application interface for editing a user profile. The sidebar on the left lists various navigation options under the 'JURAGAN' header. The top navigation bar includes a search bar, a shopping cart icon, and the user's name 'Yusniar NS' with a profile picture. The main content area is titled 'Edit Profile' and contains three form fields: 'Email' (yusniar33@gmail.com), 'Nama Lengkap' (Yusniar NS), and 'Foto Profil' (a silhouette placeholder). Below the photo field are two buttons: 'Edit Profile' and 'Kembali'.

Gambar 5.47 UI Edit Profil

Pada gambar 5.47 merupakan UI dari edit profil pelanggan dimana pada bagian *sidebar* dan *topbar* terlihat sama dengan gambar 5.45. Pada bagian *index* dimana terdapat sebuah *form* yang pada setiap *textbox* nya di ambil dari *database*. Pada bagian *email* dimana bersifat *readonly*. Terdapat dua buah *button* sebagai memulai aksi yang diberikan.

9). User Interface Ganti Password

The screenshot displays the 'Ganti Password' (Change Password) interface. It features a sidebar with navigation options like 'Buka Toko', 'PELANGGAN', 'KATEGORI', 'PRODUK', and 'PEMANDU APLIKASI'. The topbar includes a search bar, a shopping cart icon, and a user profile icon. The main content area contains three text input fields for 'Password Lama', 'Password Baru', and 'Ulangi Password', followed by 'Ganti Password' and 'Kembali' buttons.

Gambar 5.48 UI Ganti Password

Pada gambar 5.48 tersebut dimana merupakan UI dari ganti password bagi *user*. Pada bagian *sidebar* dan *topbar* akan terlihat sama dengan gambar 5.45. Pada bagian *index* terdapat sebuah *form* yang diberikan 3 *textbox* dan 2 *button* untuk memulai aksi ganti *password*.

10). *User Interface Profil Toko*

Halaman Profil Toko X

JURAGAN

Pencarian ...

Yusniar NS

Profil Toko

Email

Nama Toko

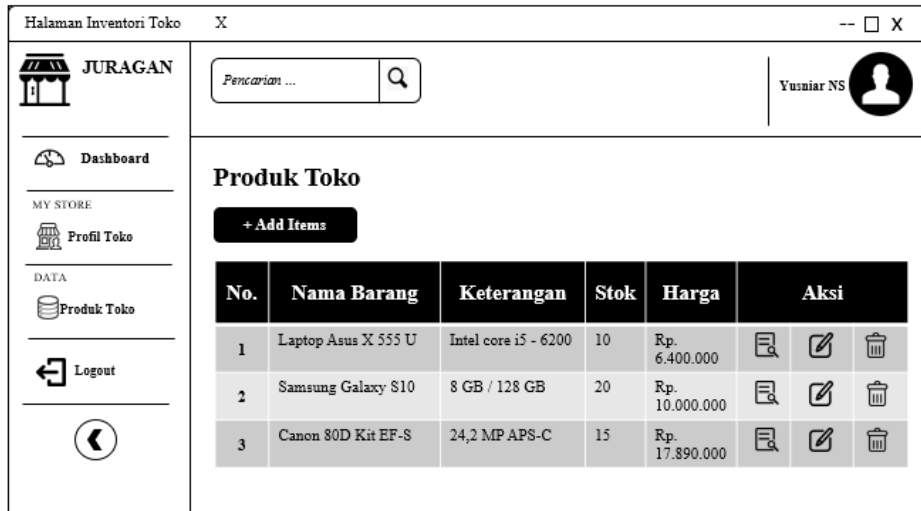
Alamat Toko

Gambar Toko

Gambar 5.49 UI Profil Toko

Pada gambar 5.49 merupakan UI dari profil toko bagi *user* penjual. Pada bagian *sidebar* terdapat beberapa *icon* yang nampak lebih sedikit dari pada gambar 5.45. Hal ini dikarenakan bedanya akses yang diberikan. Pada bagian *topbar* sekilas terlihat sama dengan gambar 5.45 akan tetapi yang membedakan adalah tidak adanya *icon cart* yang dimana *icon cart* merupakan sesuatu *fitur* yang hanya diberikan oleh *user* dengan level pelanggan. Pada bagian *index* nya terdapat satu *form* yang diberikan beberapa *textbox* serta *button edit* profil toko yang dimana akan melakukan pendaftaran pada toko tersebut.

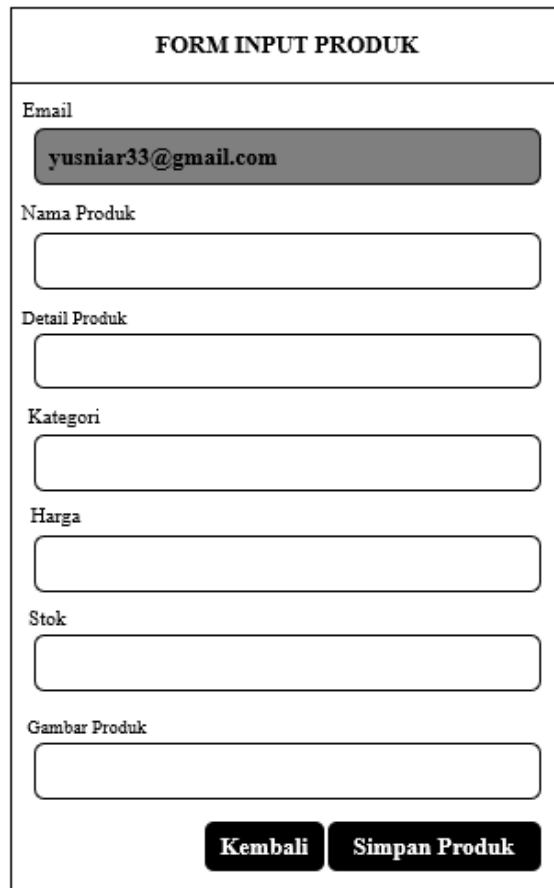
11). User Interface Produk Toko



Gambar 5.50 UI Produk Toko

Pada gambar 5.50 merupakan UI dari produk toko yang dimana akses tersebut diberikan kepada *user* dengan level pelanggan. Pada bagian *sidebar* dan *topbar* akan terlihat sama dengan gambar 5.49. Pada bagian *index* nya dimana terdapat tabel untuk melakukan *view* dari *database* berdasarkan *session* yang diberikan. Terdapat juga satu *button* yang berfungsi untuk mengakses halaman *input* produk.

12). *User Interface Form Input Produk*



FORM INPUT PRODUK

Email
yusniar33@gmail.com

Nama Produk

Detail Produk

Kategori

Harga

Stok

Gambar Produk

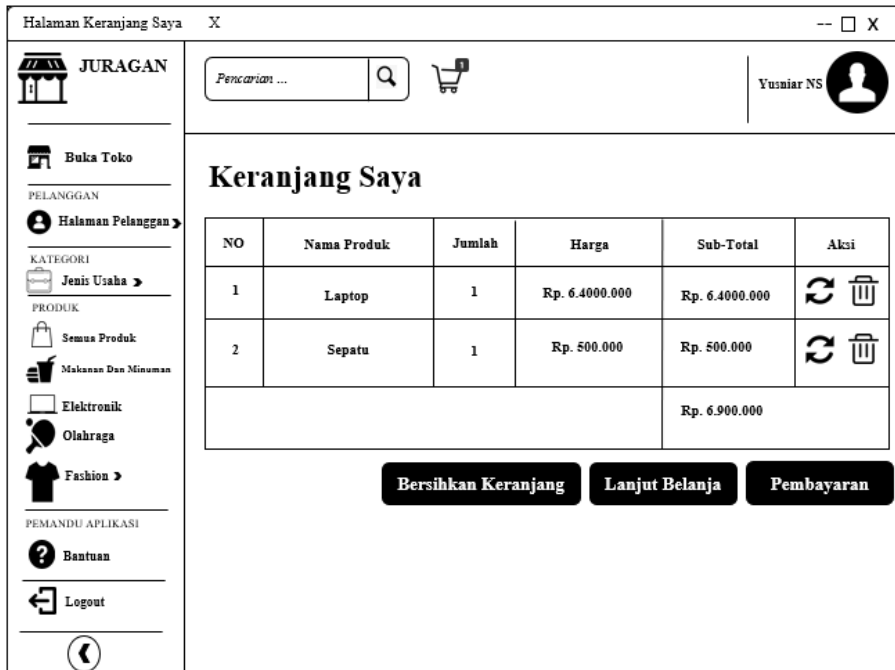
Kembali **Simpan Produk**

Gambar 5.51 UI Form Input Produk

Pada gambar 5.51 merupakan UI dari *form input* produk. Dimana hanya *user* dengan level pelanggan yang dapat mengakses UI tersebut. UI tersebut dapat dibuka dengan cara mengklik pada *button* yang berada dalam gambar 5.50. *form* tersebut berfungsi untuk menambahkan suatu produk baru yang akan di jual oleh pelanggan. Pelanggan diperlukan mengisi informasi – informasi yang dibutuhkan oleh sistem. Informasi tersebut akan di tampilkan dengan *textbox*.

Terdapat dua *button* yang dimana berfungsi untuk memulai aksi *input* produk.

13). *User Interface* Halaman Keranjang



Gambar 5.52 *UI* Halaman Keranjang

Pada gambar 5.52 merupakan *UI* dari halaman keranjang yang dimana hanya dapat di akses oleh *user* dengan level pelanggan. Pada bagian *sidebar* dan *topbar* terlihat sama dengan gambar 5.45. Pada bagian *index* nya terdapat tabel yang isinya di peroleh dari hasil menambahkan produk kedalam keranjang melalui *button*. Terdapat 2 *icon* di dalam tabel tersebut yaitu *refresh* dan *delete* yang sudah di fungsikan. Di luar tabel terdapat 3 *button* yang dimana untuk memulai aksi yang diberikan.

4.12 Perancangan Arsitektur *Software* Dan *Hardware*

Pada perancangan perangkat lunak dan perangkat keras tersebut merupakan kebutuhan – kebutuhan untuk mengakses aplikasi JURAGAN tersebut yang dimana telah dicantumkan pada tabel 5.23 – 5.24.

A. Aplikasi Kebutuhan Perangkat Lunak

Perangkat lunak (*software*) yang dibutuhkan dalam aplikasi JURAGAN tersebut adalah sebagai berikut :

Tabel 5.23 Software Requirement

No.	<i>Tool/Software</i>	<i>Description</i>
1	<i>Windows 7/8/10</i>	<i>Operation System</i>
2	<i>Xampp, VS Code, CodeIgniter, Microsoft Visio, Bizagi Modeler, Start UML, Microsoft Word</i>	<i>Development Tools</i>
3	<i>Browser</i>	<i>Web Browser Application</i>
4	<i>MariaDB</i>	<i>Database</i>

B. Aplikasi Kebutuhan Perangkat Keras

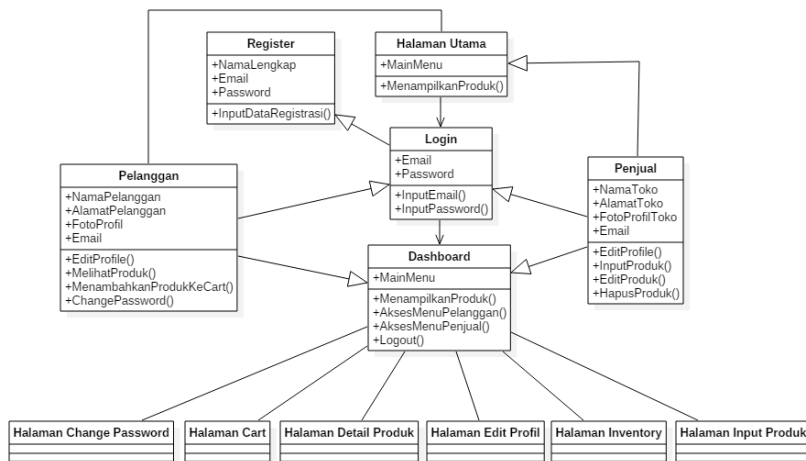
Perangkat keras (*hardware*) yang dibutuhkan dalam aplikasi JURAGAN tersebut adalah sebagai berikut :

Tabel 5.24 Hardware Requirement

No.	Device Name	Specification
1	Processor	Core Inside (min)
2	Memory	DDR3 2 GB (min)
3	Harddisk	500 GB HD (min)
4	VGA	512 MB (min)

4.13 Pemetaan Struktur Diagram User

Adapun pemetaan struktur diagram *user/aktor* pada sistem aplikasi JURAGAN tersebut adalah sebagai berikut :



Gambar 5.53 Pemetaan Struktur Diagram User/Aktor Pada Sistem JURAGAN

BAB VI

EVALUASI

6.1 Implementasi Dan Pengujian

Implementasi merupakan sistem/aplikasi yang dibuat dengan merinci komponen – komponen pendukung berupa program, lingkungan implementasi, tampilan antarmuka, dan petunjuk penggunaan.

6.1.1 Lingkungan Implementasi

Perancang aplikasi ini dapat dilaksanakan dengan baik karena didukung oleh perangkat pendukung yaitu perangkat lunak dan perangkat keras.

A. Kebutuhan Perangkat Lunak

Pada pembuatan aplikasi JURAGAN ini dimana perangkat lunak pendukung yang digunakan adalah sebagai berikut :

Tabel 6.1 Kebutuhan Perangkat Lunak

No.	<i>Tool/Software</i>	Fungsi
1	<i>Microsoft Windows 10</i>	<i>Operation System</i>
2	<i>MariaDB</i>	<i>Database</i>
3	<i>CodeIgniter</i>	<i>Framework</i>
4	<i>XAMPP For Windows 5.6.35</i>	<i>Web Server</i>
5	<i>PHP</i>	Bahasa Pemrograman
6	<i>Google Chrome</i>	<i>Web Browser</i>

7	<i>Bootstrap</i>	<i>Framework CSS</i>
---	------------------	----------------------

B. Kebutuhan Perangkat Keras

Pada pembuatan aplikasi JURAGAN ini dimana perangkat keras pendukung yang digunakan adalah sebagai berikut :

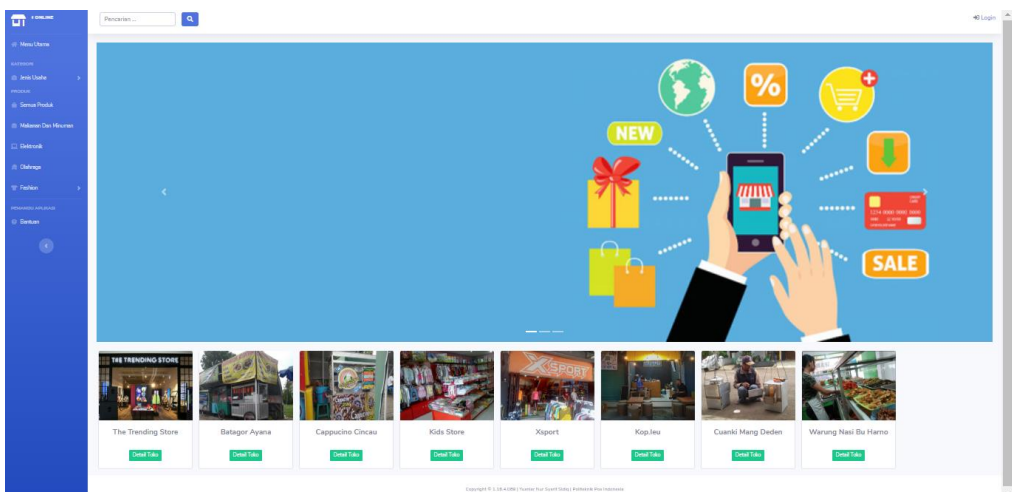
Tabel 6.2 Kebutuhan Perangkat Keras

No.	Nama Perangkat	Spesifikasi	Keterangan
1	<i>Processor</i>	<i>Intel Core i5-62000 2,8 GHz</i>	Media untuk menyimpan data aplikasi yang dibuat.
2	<i>Memory</i>	<i>8 GB</i>	<i>Memeory system</i> yang digunakan.
3	<i>Hardisk</i>	<i>500 GB</i>	Untuk kecepatan <i>transfer</i> data dari sistem yang sangat bergantung pada kecepatan prosesor dan sebagai media penyimpanan data.
4	<i>Mouse</i> dan <i>Keyboard</i>	<i>Standart</i>	Alat pendukung.
5	<i>Monitor</i>	<i>All Device</i>	Menampilkan <i>user interface</i> .
6	Infrastruktur Jaringan		Merupakan media penghubung jaringan komputer.

6.2 Pembahasan Hasil Implementasi

Berdasarkan peranangan yang telah dibuat, didapat hasil dari implementasi yang menjadi tujuan pembuatan perangkat lunak ini yaitu dapat mengelola seluruh data Rancang Bangun Aplikasi JURAGAN (Jualan Rakyat Gabungan Online) Dengan *Platform E-commerce* Berbasis Komunitas.

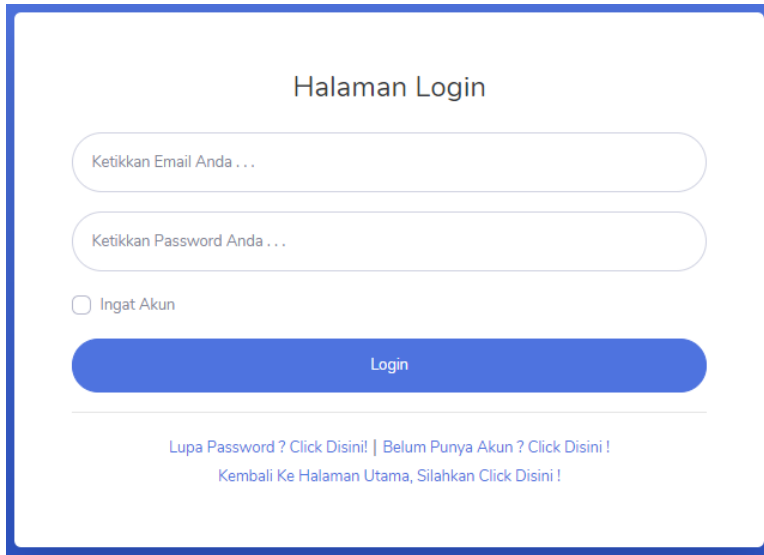
A. Halaman Utama JURAGAN



Gambar 6.1 Halaman Utama JURAGAN

Pada halaman utama tersebut merupakan tampilan awal saat pertama kali *user* mengakses sistem JURAGAN. Pada halaman utama tersebut dimana *user* dapat melakukan *explore* terlebih dahulu terhadap sistem akan tetapi tidak bisa menambahkan produk ke dalam keranjang.

B. Halaman *Form Login*



Halaman Login

Ketikkan Email Anda ...

Ketikkan Password Anda ...

☐ Ingat Akun

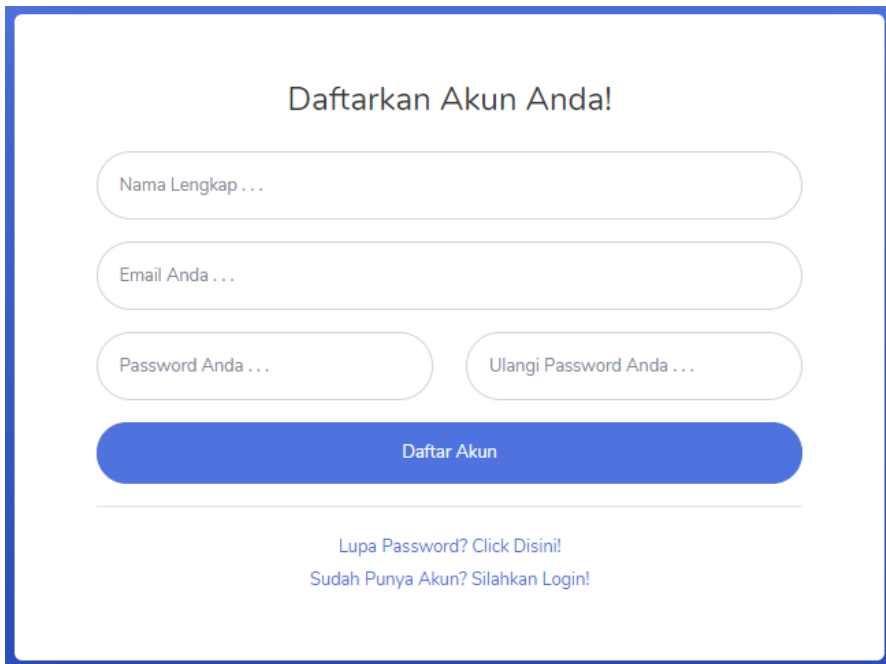
Login

[Lupa Password ? Click Disini!](#) | [Belum Punya Akun ? Click Disini !](#)
[Kembali Ke Halaman Utama, Silahkan Click Disini !](#)

Gambar 6.2 Halaman Form Login

Pada gambar 6.2 merupakan halaman *form login* yang dimiliki oleh aplikasi JURAGAN tersebut. Prosedur yang diperlukan dalam melakukan *login* dimana *user* harus melakukan *input email* dan *password* yang telah terdaftar dan aktif di aplikasi JURAGAN.

C. Halaman Daftar Akun



Daftarkan Akun Anda!

Nama Lengkap ...

Email Anda ...

Password Anda ...

Ulangi Password Anda ...

Daftar Akun

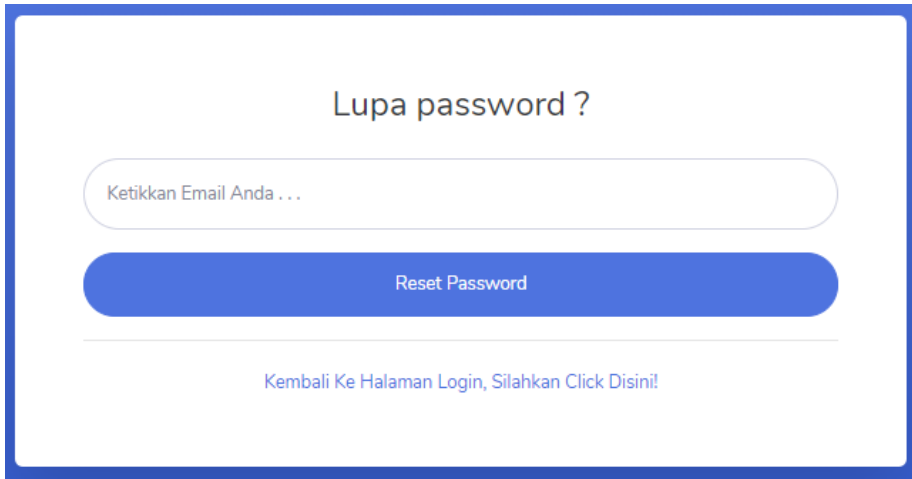
[Lupa Password? Click Disini!](#)

[Sudah Punya Akun? Silahkan Login!](#)

Gambar 6.3 Halaman Daftar Akun

Pada gambar 6.3 tersebut merupakan halaman daftar akun yang dimiliki oleh aplikasi JURAGAN. Dimana prosedur yang perlu dilakukan adalah *user* akan melakukan *input* data sesuai dengan keterangan *form* tersebut. Jika sudah *user* akan menekan *button* daftar akun untuk mengirim data tersebut. Data akan terdaftar namun belum aktif, untuk mengaktifkannya dimana *user* perlu melakukan aktivasi melalui *email*.

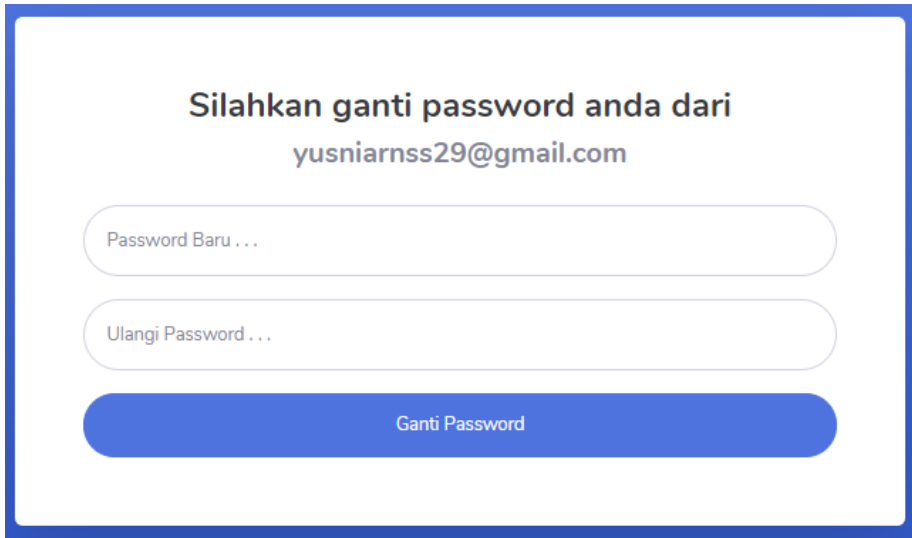
D. Halaman Lupa Password



Gambar 6.4 Halaman Lupa Password

Pada gambar 6.4 tersebut merupakan halaman lupa *password* pada aplikasi JURAGAN. Dimana prosedur dalam melakukannya *user* akan meng *input email* yang sudah terdaftar pada aplikasi JURAGAN. Sistem akan mengirimkan *email* kepada *user* dan apabila di *click* maka *password* akan di hapus, sistem akan membawa *user* ke halaman *reset password* pada gambar 6.5 untuk melakukan *reset password*.

E. Halaman Ganti *Password*



Silahkan ganti password anda dari
yusniarnss29@gmail.com

Password Baru ...

Ulangi Password ...

Ganti Password

Gambar 6.5 Halaman Ganti Password

Pada gambar 6.5 tersebut merupakan halaman ganti *password* pada aplikasi JURAGAN. Dimana *user* akan melakukan *input password* dan memberikan *action* terhadap *button* ganti *password*, maka *password* akan diganti dengan yang baru.