

**SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI**

**ANALISIS DAN PERANCANGAN SISTEM INFORMASI JUAL BELI BAHAN BAKU BERBASIS WEB STUDI KASUS PETANI DAN PETERNAK DI KOTA DEPOK**

**SKRIPSI**

**SLAMET SANTOSO**

**0110212030**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**DEPOK**

**JUNI 2018**



**SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI**

**ANALISIS DAN PERANCANGAN SISTEM INFORMASI JUAL BELI BAHAN BAKU BERBASIS WEB STUDI KASUS PETANI DAN PETERNAK DI KOTA DEPOK**

**SKRIPSI**

**SLAMET SANTOSO**

**0110212030**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**DEPOK**

**JUNI 2018**

**HALAMAN PERNYATAAN ORISINALITAS**

**Skripsi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.**

**Nama : SLAMET SANTOSO**

**NIM : 0110212030**

Depok, Juli 2018

Tanda tangan

Materai 6000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# HALAMAN PENGESAHAN TUGAS AKHIR

Skripsi ini diajukan oleh:

Nama : SLAMET SANTOSO

NIM : 0110212030

Program studi : Teknik Informatika

Judul Skripsi : Analisis dan Perancangan Sistem Informasi Jual-Beli Bahan Baku Petani dan Peternak Berbasis Web (Studi Kasus Kota Depok)

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik Komputer pada Program Studi Teknik Informatika Sekoah Tinggi Teknologi Terpadu Nurul Fikri.

**DEWAN PENGUJI**

Pembimbing : Indra Hermawan, M.Kom ( )

Penguji 1 : Sirojul Munir, S.Si, M.Kom ( )

Penguji 2 :  [Ahmad Rio Adriansyah, S.Si. M.Si.](http://nurulfikri.ac.id/index.php/indra-hermawan) ( )

Ditetapkan di : Depok

Tanggal : 30 Juli 2018

**KATA PENGANTAR**

Puji dan syukur ke hadirat Allah swt, karena berkat rahmat-Nya skripsi ini dapat diselesaikan dengan baik. Skripsi ini dibuat dengan tujuan untuk memenuhi syarat mendapatkan gelar Strata-1 (S1) di Sekolah Tinggi Teknologi Terpadu Nurul Fikri. Penulis juga ingin mengucapkan terima kasih sedalam-dalamnya kepada semua pihak yang terlibat dalam pengerjaan penelitian ini, karena atas bantuan dari banyak pihak membuat proses pengerjaan penelitian ini menjadi lancar. Dalam kesempatan ini penulis ingin mengucapkan banyak terima kasih kepada :

1. Indra Hermawan, M.Kom. selaku pembimbing dari tugas akhir ini
2. Ahmad Rio Adriansyah, S.Si. M.Si. selaku kepala program studi teknik informatika
3. Orang tua dan keluarga saya yang telah memberikan bantuan dan dorongan dalam pengerjaan tugas akhir ini, terima kasih yang teramat dalam untuk ibunda penulis yang tiada lelah memberikan motivasi
4. Teman-teman kuliah di STT Nurul Fikri.

Penulis menyadari masih terdapat banyak kekurangan dalam pengerjaan skripsi ini, penulis sangat berterima kasih sekali jika terdapat saran ataupun kritik yang membangun guna membuat penelitian ini menjadi lebih baik. Demikian yang dapat penulis sampaikan, harapan penulis semoga skripsi ini bisa bermanfaat bagi pengembangan ilmu serta menambah wawasan bagi yang membacanya, akhir kata penulis ucapkan terima kasih.

Depok, 30 Juli 2018

Slamet Santoso

# DAFTAR ISI

[HALAMAN PENGESAHAN TUGAS AKHIR ii](#_Toc520713647)

[**KATA PENGANTAR** 3](#_Toc520713648)

[DAFTAR ISI 4](#_Toc520713649)

[DAFTAR GAMBAR 6](#_Toc520713650)

[DAFTAR TABEL 7](#_Toc520713651)

[BAB 1 PENDAHULUAN 8](#_Toc520713652)

[1.1 Latar Belakang 8](#_Toc520713653)

[1.2 Rumusan Masalah 9](#_Toc520713654)

[1.3 Batasan Masalah 9](#_Toc520713655)

[1.4 Tujuan dan Manfaat 10](#_Toc520713656)

[1.4.1 Bagi Pengembangan Ilmu Pengetahuan 10](#_Toc520713657)

[1.4.2 Bagi Institusi 10](#_Toc520713658)

[1.4.3 Bagi Masyarakat 10](#_Toc520713659)

[1.4.4 Bagi Penulis 11](#_Toc520713660)

[1.5 Sistematika Penulisan 11](#_Toc520713661)

[BAB II LANDASAN TEORI 13](#_Toc520713662)

[2.1 Tinjauan Pustaka 13](#_Toc520713663)

[2.1.1 Sistem Informasi 13](#_Toc520713664)

[2.1.2 UML (Unified Modelling Language) 15](#_Toc520713665)

[2.1.3 Framework 20](#_Toc520713666)

[2.1.4 MVC (Model-View-Contreoller) 21](#_Toc520713667)

[2.1.5 Metode Pengembangan SDLC (*System Development Life Cycle*) 21](#_Toc520713668)

[2.1.6 Model Waterfall 25](#_Toc520713669)

[2.1.7 Laravel 26](#_Toc520713670)

[2.2 Penelitian Terkait 29](#_Toc520713671)

[2.3 Aplikasi Terkait 29](#_Toc520713672)

[BAB III METODOLOGI PENELITIAN 31](#_Toc520713673)

[3.1 Alir Tahapan Penelitian 31](#_Toc520713674)

[3.2 Rancangan Penelitian 33](#_Toc520713675)

[3.2.1 Jenis Penelitian 33](#_Toc520713676)

[3.2.2 Metode Pengumpulan Data 34](#_Toc520713677)

[3.2.3 Tempat dan Lokasi Penelitian 34](#_Toc520713678)

[3.2.4 Alat dan Bahan 35](#_Toc520713679)

[3.2.5 Metode Pengerjaan 35](#_Toc520713680)

[BAB IV ANALISIS DAN PERANCANGAN 37](#_Toc520713681)

[4.1 Tinjauan Umum Kondisi Awal 37](#_Toc520713682)

[4.1.1 Diagram Kebutuhan (*Use case diagram*) 38](#_Toc520713683)

[4.2 Analisis Sistem 40](#_Toc520713684)

[4.2.1 Diagram Kebutuhan (*Use case diagram*) 41](#_Toc520713685)

[4.2.2 Deskripsi Kebutuhan (*Use case Scenario*) 42](#_Toc520713686)

[4.3 Perancangan Sistem 46](#_Toc520713687)

[4.3.1 Desain Sistem 46](#_Toc520713688)

[BAB V IMPLEMENTASI DAN EVALUASI 64](#_Toc520713689)

[5.1 Implementasi Penggunaan *Framework* 64](#_Toc520713690)

[5.2 Implementasi Antar Muka 65](#_Toc520713691)

[DAFTAR PUSTAKA 68](#_Toc520713692)

# DAFTAR GAMBAR

[Gambar 3.1- Rancangan Penelitian 33](#_Toc485127831)

# DAFTAR TABEL

[Tabel 2.1 – Use Case Diagram 16](#_Toc520710833)

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

Indonesia merupakan negara agraris dan sebagian besar penduduknya bermata pencaharian dibidang pertanian. Sebenarnya negara ini diuntungkan Karena dikaruniai kondisi alam yang mendukung, hamparan lahan yang luas, keragaman hayati yang melimpah, serta beriklim tropis dimana sinar matahari terjadi sepanjang tahun sehingga bisa menanam sepanjang tahun. Realita sumberdaya alam seperti ini sewajarnya mampu membangkitkan Indonesia menjadi negara yang makmur, tercukupi kebutuhan pangan seluruh warganya. Meskipun demikian belum terpenuhi, pertanian menjadi salah satu sector riil yang memiliki peran sangat nyata dalam membantu menghasilkan devisa negara.

Berdasarkan data yang diperoleh dari BPS (Badan Pusat Statistik) Tahun 2009, jumlah petani 44% dari total angkatan kerja di Indonesia, atau sekitar 46,7 juta jiwa. Sebagai negara agraris, hingga kini mayoritas penduduk Indonesia telah memanfaatkan sumberdaya alam untuk menunjang kebutuhan hidupnya dan salah satunya ialah dengan menggantungkan hidup pada sektor pertanian. Adanya halnya hal tersebut sector pertanian memiliki peran yang sangat pentin, Karena sebagai penghasil pangan bagi penduduk yang jumlah tiap tahunnya selalu bertambah.

Sumber daya alam hewan dapat berupa hewan liar maupun hewan yang sudah dibudidayakan. Pemanfaatannya dapat sebagai pembantu pekerjaan berat manusia, seperti [kerbau dan kuda](http://id.wikipedia.org/w/index.php?title=Kerbau_dan_kuda&action=edit&redlink=1) atau sebagai sumber bahan pangan, seperti [unggas](http://id.wikipedia.org/wiki/Unggas) dan [sapi](http://id.wikipedia.org/wiki/Sapi). Untuk menjaga keberlanjutannya, terutama untuk satwa langka, pelestarian secara in situ dan ex situ terkadang harus dilaksanakan. Pelestarian in situ adalah pelestarian yang dilakukan di habitat asalnya, sedangkan pelestarian ex situ adalah pelestarian dengan memindahkan hewan tersebut dari habitatnya ke tempat lain. Untuk memaksimalkan potensinya, manusia membangun sistem [peternakan](http://id.wikipedia.org/wiki/Peternakan), dan juga [perikanan](http://id.wikipedia.org/wiki/Perikanan), untuk lebih memberdayakan sumber daya hewan.

Memasuki dunia digital saat ini mengajak semua *sector* termaksud pertanian dan perternakan untuk ikut bergabung di dalamnya, agar dapat menjawab tantangan dunia teknologi digital yang berkembang saat ini.

Permasalahan petani dan peternak berkaitan dengan sistem penjualan bahan baku yang tidak dapat dipungkiri terdapat sedikit kendala dimana dalam proses penjualan bahan baku tersebut orang ketiga (tengkulak) bermain harga yang menjadikan harga di pasaran terlampau jauh. dengan adanya Sistem Informasi Berbasis Web untuk Jual Beli Bahan Baku berbasis web menjadi solusi dalam hal transparansi, media punghubung bagi pelaku usaha yang membutuhkan bahan baku langsung dari petani.

Berdasarkan pemaparan diatas akhirnya membuat penulis merasa tergerak untuk melakukan penelitian yang berjudul “Analisis dan Perangcangan Sistem Informasi Jual Beli Hasil Produksi Petani dan Peternak”

## 1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan di atas dapat dirumuskan permasalahan penelitian sebagai berikut :

1. Apakah dengan menggunakan sistem informasi penjualan bahan pokok hasil produksi petani dan peternak berbasis *web* dapat mengetahui harga pangan terkini dari petani dan peternak ?
2. Apakah sistem informasi penjualan bahan pokok hasil produksi petani dan peternak berbasis *web* dapat membantu petani dan peternak memasarkan dan memange hasil panen dan ternak ?
3. Apakah sistem informasi penjualan bahan pokok hasil produksi petani dan peternak berbasis *web* dapat membantu pihak Petani dan Peternak dalam memasukkan data transaksi dan sosialisasi program pada sistem aplikasi ?

## 1.3 Batasan Masalah

Oleh karena besarnya permasalahan dan keterbatasan waktu serta pengetahuan penulis maka agar pembahasan tidak menyimpang dari tujuan dilakukan pembatasan masalah sebagai berikut:

1. Penulis membatasi wilayah yang diteliti studi kasus petani dan peternak di kota depok dengan mengambil sampel ( 3 Petani dan 3 Peternak)
2. Penulis membatasi konsumen 6 dari pelaku usaha yang memanfaatkan bahan baku dari petani atau peternak.
3. Penulis merancang sistem berbasis web menggunakan bahasa PHP dengan framework Laravel 5.6 dan database mysql.
4. Sistem ini akan diuji coba hanya kepada 3 Petani dan 3 Peternak kota Depok, Jawa Barat.
5. Sistem informasi ini nantinya hanya membahas fitur – fitur untuk membantu manajemen petani, peternak, dan pembeli saja dan tidak membahas masalah proses pencairan dana.
6. Dalam penelitian ini penulis tidak membahas secara mendetail terkait dengan keamanan sistem karena sudah diatur oleh *framework PHP* yang digunakan.
7. Dalam penelitian ini penulis tidak membahas secara mendetail terkait dengan *User Interface* dan *User Experience*.

## Tujuan dan Manfaat

Selanjutnya apabila penelitian ini berhasil dengan baik, diharapkan dapat memberikan manfaat sebagai berikut



### Bagi Pengembangan Ilmu Pengetahuan

Menjadi tambahan pustaka terhadap wacana pendidikan di Indonesia dan khazanah keilmuan pendidikan, khususnya tentang pengembangan media informasi yang kreatif dan inovatif.

### Bagi Institusi

Penelitian ini diharapkan berguna sebagai tolak ukur dan referensi institusi pendidikan dalam melakukan penelitian berbasis produk aplikasi yang bermanfaat untuk mensejahterakan petani dan peternak.

### Bagi Masyarakat

Memberikan sumbangan kepada masyarakat luas untuk dapat meningkatkan taraf hidup masyarakat serta mendorong untuk mandiri secara ekonomi.

### ­Bagi Penulis

Menambah dan mengembangkan wawasan keilmuan penulis yang berkaitan dengan pengembangan sistem informasi berbasis *web***.** Serta memotivasi sesama pelajar untuk membuat aplikasi teknologi yang bermanfaat bagi manusia .

## 1.5 Sistematika Penulisan

Penyusunan system informasi jual beli online berbasis web ini mengikuti sistematika berikut ini :

Bab 1 : Pendahuluan

Bab ini berisi latar belakang dari penulisan proposal tugas akhir, perumusan masalah, tujuan dan manfaat penelitian, batasan masalah, dan sistematika dari penuisan proposal tugas akhir ini.

Bab 2 : Landasan Teori

Bab ini berisikan mengenai pembahasan teori pengembangan system dengan metode pendekatan Agile Development SCRUM, teori penelitian terkait dengan perancangan system informasi jual beli bahan baku berbasis web, teori mengenai desain system menggunakan UML, teori mengenai pengembangan system informasi menggunakan framwork, konsep MVC dan pembahasan mengenai laravel.

Bab 3 : Metodologi Penelitian

Bab ini berisikan tahapan yang dilakukan dalam penelitian diantaranya terhadap penegembangan system dengan metode waterfall SDLC. Rancangan penelitian yang menjelaskan mengenai cara menyelesaikan permasalahan, jenis penelitian yang dilakukan, rancangan analisis dan pengumpulan data, metode pengembangan software, metode pengujian serta timeline penelitian dalam perancangan sistem informasi jual beli bahan baku berbasis web.

Bab 4 : Analisis dan Perancangan

Bab ini berisikan tinjauan umum kondisi awal, anaisis sistem, diagram kebutuhan (use case diagram), deskripsi kebutuhan (use case scenario), perancangan sistem berupa desain sistem dan antar muka sistem.

Bab 5 : Implementasi dan Evaluasi

Bab ini berisikan tahapan implementasi aplikasi web yang telah dibagun, implementasi antar muka, evaluasi pengujian melalui UAT (User Acceptene Test) dan User survey.

Bab 6 : Kesimpulan dan Saran

Bab ini berisikan Kesimpulan dan saran dari hasil penelitian.

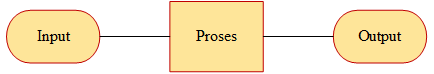
# BAB II LANDASAN TEORI

Bab ini mengenai pembahasan teori pengembangan system dengan metode pendekatan Waterfall SDLC, teori penelitian terkait dengan perancangan sistem informasi jual beli bahan baku berbasis online, teori mengemai desain sistem menggunakan UML, teori mengenai pengembangan sistem informasi menggunakan framework, konsep MVC dan pembahasan mengenai Laravel.

## 2.1 Tinjauan Pustaka

### 2.1.1 Sistem Informasi

Menurut Robert A. Leitch dan K. Roscoe Davis “Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasional, bersifat menajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan” (Jogiyanto, 2003). Tujuan nya adalah untuk menyajikan informasi guna pengambilan keputusan pada perencanaan, pemrakarsaan, pengorganisasian, pengendalian kegiatan operasi subsistem suatu perusahaan dan menyajikan sinergi organisasi pada proses (Murdick, 1993). dengan demikian, sistem informasi berdasarkan konsep (input, proses, output – IPO) dapat dilihat pada gambar berikut :



Gambar 1.1 - Konsep Sistem Informasi (Jogiyanto, 2003)

Sementara Burch dan Grudnistki (1986) berpendapat, sistem informasi yang terdiri dari komponen-komponen diatas disebut dengan istilah blok bagunan (*Building Block*), yaitu blok masukan (input block), blok model (model block). Blok keluaran (output block), blok teknologi (technology block), dan block kendali (control block), sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya (Jogi yanto, 2003)

1. Blok Masukan

Input mewakili data yang masuk ke dalam system informasi. Input di sini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan dapat berupa dokumen-dokumen besar

1. Blok Model

Blok ini terdiri dari kombinasi prosedu, logika dan model matematika yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.

1. Blok Keluaran

Produk dari system informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkat manajemen serta pemakai system.

1. Blok Teknologi

Teknologi merupakan kotak alat (*tool-box*) dalam sistem informasi. Teknologi digunakan untuk meneripa input, menjalankan model, menyimpan dan mengakses data, menghasilkan sekaligus mengirimkan keluaran dan membantu pengemdalian dari system secara keseluruhan.

1. Blok Database

Database merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras *computer* dan digunakan perangkat untuk memanipulasinya.

1. Blok Kendali

Pengedalian perlu dirancang dan diterapkan untuk menyakinkan bahwa hal-hal yang dapat merusak system dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

Sistem informasi selalu menggambarkan, merencang, mengimpelentasi dengan menggunakan proses perkembangan sistematis dan merancang sistem informasi berdasarkan analisis kebutuhan. Jadi, bagian utama dari proses ini adalah mengetahui rancangan dan analisis sistem, seluruh aktivitas utama dilibatkan dalam siklus perkembangan yang lengkap (Syachbana, 2011).

### 2.1.2 UML (Unified Modelling Language)

Unified Modeling Language (UML) adalah tujuan umum, perkembangan, bahasa permodelan di bidang rekayasa perangkat lunak yang dimaksudkan untuk menyediakan cara standar untuk memvisualisasikan desain sistem. UML awalnya termotivasi oleh keinginan untuk membakukan sistem notasi yang berbeda dan pendekatan untuk desain perangkat lunak yang dikembangkan oleh Grady Booch, Ivar Jacobson dan James Rumbaugh di Rational Software di 1994-1995, dengan pengembangan lebih lanjut yang dipimpin oleh mereka malalui tahun 1996.

Menurut Nugroho (2010:6), UML (Unified Modeling Language) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih muda dipelajari dan dipahami. UML sudah menajadi standardized modeling language yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengmbang sistem dan software agar bisa menyelesaikan tugas-tugas seperti (Nugroho, 2009).

1. Spesifikasi
2. Visualisasi
3. Desain Arsitektur
4. Konstruksi
5. Simulasi dan Testing
6. Dokumentasi
7. Skestsa

Di dalam UML terdapat pengelompokan diagram sistem. Diagram ini terbentuk grafik yang menunjukan elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari view tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu (D.Rustandi, 2009).

Jenis-jenis diagram dalam UML (D.Rustandi, 2009).

1. *Use Case Diagram*

Menggambarkan sejumlah *eksternal actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use Case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dari dokumentasi dari Use Case symbol namun juga dapat dilakukan dalam activity diagram. Use case digambarkan hanya yang dilihat dari luar oleh actor (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem.

Tabel 2.1 – Use Case Diagram

|  |  |
| --- | --- |
| Gambar | Keterangan |
| *Use Case* | Fungsionalitas yang disediakan sistem sebagai unit-unit saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *use case* |
| Aktor/ *actor* | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol aktor adalah gambar orang, tapi aktor belum tentu orang; biasanya dinyatakan menggunakan kata benda si awal frase nama actor |
| *association* | Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor |
| *<<extend>>* | Relasi *use case* tambahan ke sebuah *use case* dimana *use* *case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu |
| *Generalization* | Hubungan (umum/khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya |
| *<<include>>* | Relasi *use case* tambahan ke sebuah *use case* dimana *use* *case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya |

(Sumber: Rosa A.S – M. Salahuddin 2011)

1. *Class Diagram*

Menggambarkan struktur statis class di dalam sistem. Class =dapat berhubungan dengan yanglain melalaui berbagai cara: *associated* (terhubung satu sama lain), *depended* (satu class tergantung atau menggunakan class yang lain), *specialed* (satu class merupakan spesialisasi dari class yang lain) atau *packed* (group Bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram.*

Tabel 2. 2. Class Diagram

|  |  |
| --- | --- |
| Gambar | Keterangan |
| Kelas | Kelas pada struktur system |
| *Interface*  **Antar muka**  **Antar muka** | Sama dengan konsep interface dalam pemrograman  berorientasi objek |
| *Association* | Relasi antar kelas dengan makna umum, asosiasi biasanya disertai disertai *multiplicity* |
| *Directed* *asosiation* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai disertai *multiplicity* |
| *Generalization* | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus) |
| *Aggregation* | Relasi antar kelas dengan makna semua-bagian *(whole-part)* |

(Sumber: Rosa A.S – M. Salahuddin 2011)

1. *State Diagram*

Menggunakan semua state (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa *object* lain yang mengirim pesan, state class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

1. *Sequence Diagram*

Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaannya untuk menunjukkan sejumlah pesan yang dikirim antara *object* juga interaksi antar *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

Table 2.3 Sequence Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| NO | GAMBAR | NAMA | KETERANGAN |
| 1 |  | *Lifeline* | Objek *entity* antarmuka yang saling berinteraksi |
| 2 |  | *Actor* | Digunakan untuk menggambarkan user/pengguna |
| 3 |  | *Message* | Spsifikasi dari sebuah komunikasi antar objek yang memuat informasi – informasi tentang aktifias yang terjadi |
| 4 |  | *Boundary* | Digunakan untuk menggambarkan sebuah form |
| 5 |  | *Control Class* | Digunakan untuk menghubungkan sebuah *boundary* dengan form |
| 6 |  | *Entity Class* | Digunakan untuk menggambarkan hubungan kegiatan yang dilakukan |

(Sumber: Rosa A.S – M. Salahuddin 2011)

1. *Collaboration Diagram*

Menggambarkan kolaborasi dinamis seperti *sequence diagram* dalam menunjukkkan pertukaran pesa,, *collaboration diagram* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekanannya pada waktu atau urutan, gunakan sequence diagram tapi jika penekanan nya pada konteks kegunakan *collaboration diagrams.*

1. *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi.

1. *Component Diagram*

Menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa *source code*, komponen biner atau class yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view.*

1. *Deployment Diagram*

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes, executable* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.

### 2.1.3 Framework

Framework secara sederhana dapat diartikan kumpulan dari fungsi-fungsi / prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau class dari awal. Ada beberapa alasan mengapa menggunakan Framework :

1. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
2. Relatif memudahkan dalam proses maintenance Karena sudah ada pola tertentu dalam sebuah Framework (dengan syarat programmer mengikuti pola standar yang ada)
3. Umumnya Framework menyediakan fasilitas-fasilitas yang umum dipakai sehingga tidak perlu membangun dari awal.
4. Lebih bebas dengan pengembangan jika dibandingkan dengan CMS.

Web Framework adalah kerangka kerja pengembangan aplikasi berbasis web. Web Framework digunakan oleh pengembang aplikasi untuk memudahkan pembangunan aplikasi yang dapat berupa sekumpulan library yang berisi fungsi , tools, ataupun class-class dan digunakan sebagai kerangka dalam pembangunan aplikasi.

Di dalam web framework menyediakan solusi untuk akses database, authentication, templating, controls dan fungsi-fungsi lainnya. Pengguna web framework diharapkan membuat pengembangan aplikasi menjadi rapid an bersih, memiliki struktur yang optimal, dan reusable. Struktur aplikasi yang lebih rapid an teratur, biasanya menggunakan structure MVC yang melakukan pemisahan antara business logic dengan presentasion. Pemisahan dilakukan dengan tujuan agar setiap perubahan yang terjadi pada presentasion logic atau business logic tidak memberikan pengaruh satu sama lainnya yang komplek (Yanuar Firdaus AW, 2008).

### 2.1.4 MVC (Model-View-Contreoller)

Konsep MVC (Model-View-Contreoller) merupakan pola yang terbukti lebih efektif dalam melakukan pembangunan suatu proyek perangkat lunak, caranya dengan melakukan pemilihan komponen dalam proyek yang akan dijadikan model, view, dan controller (Taufiq, 2010).

MVC memiliki kepanjangan dari (Model-View-Contreoller) adalah pola desain yang mengijinkan pengembang untuk membagi kode mereka dengan tiga kategori (Myer, 2008)

1. Model untuk mengola data.
2. View untuk menampilkan data dan elemen user interface.
3. Controller menangani user events yang mempenbgaruhi model dan view.

Beberapa manfaat pada pola desain MVC adalah perubahan kode program oleh user dapat dilakukan tanpa mempengaruhi yang lain, seorang desainer dapat bekerja pada antarmuka tanpa khawatir tentang penyimpanan dan pengolahan data, bagi pengembang dapat memprogram penanganan logika tanpa masuk kedalam rincian presentasi. MVC juga memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasai data, user interface, dan bagian yang menjadi control aplikasi (Sabti, 2012).

Arsitektur Web Framework memiliki keterikatan yang kuat dengan konsep MVC, seperti yang tertera pada gambar berikut (Taufiq, 2010)

### 2.1.5 Metode Pengembangan SDLC (*System Development Life Cycle*)

Metode Pengembangan Sistem yang digunakan dalam perancangan sistem

informasi ini adalah menggunakan SDLC (*System Development Life Cycle*) atau yang biasa dikenal dengan siklus hidup pengembangan sistem. Seperti yang berlaku pada kebanyakan proses, pengembangan system informasi juga memiliki siklus hidup. Beberapa definisi tentang siklus hidup pengembangan sistem diantaranya sebagai berikut.

1. Menurut IEEE [24] bahwa *the life cycle of a software system is normaly defined as the period of time that starts when a software product is connceived and ends when the product is no longer available for use.* Dari standard IEEE 1016, ditekan bahwa siklus hidup pengembangan sistem adalah segala sesuatu yang lebih berdasar kepada urutan waktu pengembangan dibandingkan dengan proses yang terjadi.
2. Menurut Keyes [25] bahwa *a system has a life of its own. It starts out as an idea and progresses until this idea germinates and then is born.* Dalam bukunya, Jeniffer Keyes lebih menekankan bahwa sebuah perangkat lunak bisa mengalami sebuah siklus hidup bergantung dari pros pengembangannya mulai dari ide dasar hingga saat lahirnya perangkat lunak itu sendiri.
3. Sedangkan menurut Soetam Rizky [26] pengertian siklus hidup pengembangan sistem adalah urutan hidup sebuah perangkat lunak berdasarkan perkembangan perangkat lunak yang ditentukan oleh pengembang perangkat lunak itu sendiri.

Dari beberapa defini diatas penuis menarik kesimpulan bahwa SDLC (*System Development Life Cycle*) atau siklus hidup pengembangan sistem adalah urutan fase hidup sebuah perangkat lunak berdasarkan perkembangan perangkat lunak yang ditentukan oleh pengembang perangkat lunak itu sendiri sehingga dapat diketahui usia fungsionalitas dari suatu sistem. SDLC ini merupakan metodologi klasik yang digunakan untuk mengembangkan, memelihara, dan menggunakan sistem informasi. Metode Pengembangan Sistem SDLC (*System Development Life Cycle*) atau yang biasa dikenal dengan daur hidup pengembangan sistem. SDLC yang terkenal adalah SDLC model klasik yang biasa disebut dengan model waterfall. Alasan penulis menggunakan waterfall dikarenakan metode ini mempunyai tahapantahapan yang jelas, nyata dan praktis. Setiap tahap harus diselesaikan terlebih dahulu untuk menghindari terjadinya pengulangan dalam tahapan sehingga pengembangan sistem yang dilakukan dapat memperoleh hasil yang diinginkan. Metode ini mencakup sejumlah fase atau tahapan.

Tahapan-tahapan atau fase dalam metode SDLC menurut Abdul Kadir [27] adalah sebagai berikut.

1. Analisis Sistem

Tahapan analisis sistem dimulai karena adanya permintaan terhadap system baru. Permintaan dapat datang dari seorang manajer di luar sistem informasi atau dari pihak eksekutif yang melihat adanya masalah atau menemukan adanya peluang baru. Namun, adakalanya inisiatif pengembangan sistem baru berasal dari bagian yang bertanggungjawab terhadap pengembangan sistem informasi, yang bermaksud mengembangkan sistem yang sudah ada atau mengatasi masalah - masalah yang belum tertangani.

Untuk melaksanakan hal tersebut, dibentuklah proyek baru yang ditangani dalam bentuk tim, yang melibatkan pemakai, analis sistem, dan para spesialis sistem informasi yang lain, atau bahkan juga auditor internal. Tujuan utama analisis sistem adalah untuk menentukkan hal-hal detail tentang sistem yang akan dikerjakan oleh sistem yang diusulkan. Analisis sistem mencakup studi kelayakan dan analisis kebutuhan.

1. Studi kelayakan

Studi kelayakan digunakan untuk menentukan kemungkinan keberhasilan solusi yang diusulkan. Tahapan ini berguna untuk memastikan bahwa solusi yang diusulkan tersebut benar-benar dapat dicapai dengan sumber daya dan dengan memperhatikan kendala yang terdapat pada perusahaan serta dampak terhadap lingkungan sekeliling.

1. Analisis kebutuhan

Analisis kebutuhan dilakukan untuk menghasilkan spesifikasi kebutuhan (disebut juga spesifikasi fungsional). Spesifikasi kebutuhan adalah spesifikasi yang rinci tentang hal-hal yang akan dilakukan sistem ketika diimplementasikan. Spesifikasi ini sekaligus dipakai untuk membuat kesepakatan antara pengembang sistem dengan pemakai yang kelak menggunakan sistem.

1. Desain Sistem

Setelah kebutuhan dikumpulkan secara lengkap informasi mengenai kebutuhan-kebutuhan tersebut diubah ke dalam struktur data dengan menggunakan beberapa alat (tools) seperti DFD (*Data Flow Diagram*) dan ERD (*Entity Relationship Diagram).*

1. Implementasi Sistem

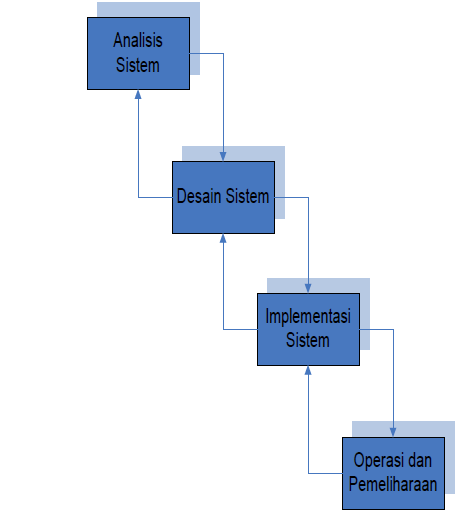
Pada fase ini, desain sistem diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Pemrograman adalah aktivitas pembuatan program atau sederetan instruksi yang digunakan untuk mengatur komputer agar bekerja sesuai dengan maksud masing-maisng instruksi. Setelah itu pemrograman selesai dilakukan, tahap selanjutnya adalah pengujian terhadap unit-unit yang dihasilkan. Pengujian seperti itu disebut dengan pengujian unit.

1. Operasi dan pemeliharaan

Setelah masa sistem berjalan sepenuhnya menggantikan sistem lama, system memasuki pada tahapan operasi dan pemeliharaan. Selama sistem beroperasi, pemeliharaan sistem tetap diperlukan karena beberapa alasan. Pertama, mungkin sistem masih menyisakan masalah-masalah yang tidak terdeteksi selama masa pengujian sistem. Kedua, pemeliharaan diperlukan karena perubahan bisnis atau lingkungan atau adanya permintaan kebutuhan baru (misalnya berupa laporan). Ketiga, pemeliharaan juga bisa dipicu karena kinerja sistem yang menjadi menurun sehingga ada perubahan-perubahan dalam penulisan

Tahapan-tahapan pengembangan sistem dengan SDLC dapat digambarkan sebagai berikut.

Gambar 2. 1.. Tahapan dalam SDLC



(Sumber : Kadir, 2003)

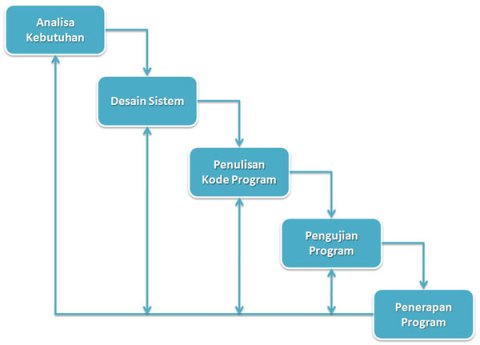
### 2.1.6 Model Waterfall

Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian. Berikut adalah gambar pengembangan perangkat lunak berurutan/ linear. [28]

Dalam pengembangannya metode waterfall memiliki beberapa tahapan yang runtut: requirement, design, implementation, verification dan maintenance. Tahap requirement atau spesifikasi kebutuhan sistem adalah analisa kebutuhan sistem yang dibuat dalam bentuk yang dapat dimengerti oleh klien dan staf pengembang. Dalam tahap ini klien atau pengguna menjelaskan segala kendala dan tujuan serta mendefinisikan apa yang diinginkan dari sistem. Setelah dokumen spesifikasi disetujui maka dokumen tersebut menjadi kontrak kerja antara klien dan pihak pengembang. Tahap selanjutnya adalah desain, dalam tahap ini pengembang akan menghasilkan sebuah arsitektur sistem secara keseluruhan, dalam tahap ini menentukan alur perangkat lunak hingga pada tahap algoritma yang detil.

Selanjutnya tahap implementasi, yaitu tahapan dimana keseluruhan desain diubah menjadi kode-kode program. kode program yang dihasilkan masih berupa modul-modul yang selanjutnya akan di integrasikan menjadi sistem yang lengkap untuk meyakinkan bahwa persyaratan perangkat lunak telah dipenuhi. Tahap selanjutnya adalah verifikasi oleh klien, klien menguji apakah sistem tersebut telah sesuai dengan kontrak yang telah disetujui. Tahap akhir adalah pemeliharaan yang termasuk diantaranya instalasi dan proses perbaikan sistem sesuai kontrak. [29]

Gambar 2. 3 Tahapan Model Waterfall



(Sumber : <http://rhs13.blog.ugm.ac.id/files/2011/02/waterfall.jpg> diakses pada 13 Juni 2017)

### 2.1.7 Laravel

Laravel merupakan framework PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya. Laravel dirilis dibawah lisensi MIT dengan sumber kode yang disediakan di Github. Sama seperti framework PHP lainnya, Laravel dibagun dengan basis MVC (Model-View-Controller). Laravel dilengkapi command line tool yang bernama “Artisan” yang bisa digunakan untuk packaging bundle dan instalasi bundle. Framework Laravel dibuat oleh Taylor Otwell, proyek larevel dimulai pada april 2011. Awal mula proyek ini dibuat Karena Otwell sendiri tidak menemukan framework yang up-to-date dengan versi PHP. Mengembangkan framework yang sudah ada juga bukan merupakan ide yang bagus Karena keterbatasan sumber daya. Dikarnakan beberapa keterbatasan tersebut, Otwell membuat sendiri framework dengan nama Laravel. Oleh Karena itu Laravel mengisyaratkan PHP versi 5.3 keatas (Putih, 2014).

Laravel merupakan framework aplikasi web yang ekspresif dan sintaks yang elegan. Kami membangun framewok ini dengan menyenangkan dan dari pengelaman pengalaman kreatif agar menjadi produk yang memuaskan. Laravel berusaha mencoba menjadi framework yang mudah digunakan dengan mengurangi tugas-tugas umum yang sering digunakan dalam sebagian besar proyek-proyek web seperti otentikasi, routing, session, dan caching. Larevel membuat proses development yang menyenangkan bagi pengembang tanpa mengurangi fungsionalitas apliasi, dengan harapan, pengembang dapat membuat rangkaian kode-kode terbaik. Larevel berusaha untuk menggabungkan yang terbaik dari apa yang ada di dalam framework lain, termasuk framework yang menggunakan Bahasa lain, seperti Ruby on Rails, ASP.NET MVC, dan Sinatra. Laravel framework yang mudah diakses, powerful dan menyediakan tools yang diperlukan untuk skala aplikasi besar. Larevel juga merupakan sebuah aplikasi luar biasa dari sebuah kumpulan program control, sistem migrasi yang eksresif dan dukungan tools yang anda butuhkan dalam menguji aplikasi anda yang terintegrasi dengan beberapa apliaksi lain (Putih, 2014).

Beberapa fitur yang dimiliki framework Laravel (Putih, 2014):

1. Bundles yaitu sebuah fitur dengan system pengemasan modular dan berbagai bundle telah tersedia untuk digunakan dalam aplikasi anda
2. Eloquent ORM merupakan penerapan PHP lanjutan dari pola “active record”, menyediakan metode internal untuk mengatasi kendala hubungan antara objek database. Pembangunan query Laravel;s fluent ini didukung oleh Eloquent.
3. Aplicaton logic merupakan bagian dari aplikasi yang dikembangkan baik menggunakan Controllers atau sebagai bagian dari deklarasi route. Sintak yang digunakan untuk mendefinisikannya mirip dengan yang digunakan oleh Framework Sinatra
4. Reverse routing, mendefinisikan hubungan antara link dan Route, sehingga jika suatu saat ada perubahan pada route secara otomatis akan tersambung dengan link yang relevan. Ketika link yang dibuat dengan menggunakan nama-nama route yang ada, secara otomatis Laravel akan membuat URI yang sesuai.
5. Restful Controllers, memberikan sebuah option (pilihan) untuk memisahkan logika dalam melayani HTTP GET dan permintaan POST
6. Class auto loading, menyediakan otomatisasi loading untuk class-class PHP, tanpa membutuhkan pemeriksaan manual terhadap jalur masuknya. Fitur ini mencegah loading yang tidak perlu.
7. View composer adalah kode untuk logical yang dapat dieksekusi ketia sebuah View diload.
8. IoC Container memungkinkan untuk objek baru yang dihasilkan dengan mengikuti prinsip control pembalik dengan pilihan contoh dan referensi dari objek baru sebagai singletons.
9. Migrations menyediakan versi sistem control untuk skema database, sehingga memungkinkan untuk menghubungkan perubahan dalam basis dalam kode aplikasi dan keperluan yang dibutuhkan dalam merubah tata letak database, mempermudah dalam penempatan dan memperbarui aplikasi.
10. Unit Testing mempunyai peran penting dalam framework Laravel, dimana unit testing ini mempunyai banyak tes untuk mendeteksi dan mencegah regresi. Unit testing ini dapat dijalankan melalaui utilitas “artisan command-line”.
11. Automatic pagination menyederhanakan tugas dari penerapan halaman, menggantikan penerapan yang manual dengan metode otomatis yang terintegrasi ke Laravel.

## 2.2 Penelitian Terkait

Berdasarkan studi pustaka yang telah dikerjakan, telah ditemukan beberapa penelitian yang dijadikan acuan dalam pengembangan penelitian yang akan di buat, diantaranya adalah sebagai berikut :

Tabel 2. 1. Hasil Penelitian Terdahulu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO | JUDUL PENELITIAN | MASALAH PENELITIAN | HASIL | PERBEDAAN | PERSAMAAN |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## 2.3 Aplikasi Terkait

Berdasarkan studi pustaka yang telah dikerjakan, telah ditemukan beberapa penelitian yang dijadikan acuan dalam pengembangan penelitian yang akan di buat, diantaranya adalah sebagai berikut :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Penelitian Terkait | Penerapan | Fitur | Ket |
| 1. | Limakilo.id | Web |  |  |
| 2. | TaniBox | Web & Mobile | Blockchain & IOT |  |
| 3. | Igrow | Web & Mobile |  |  |
| 4. | Tanihub | Web & Mobile |  |  |
| 5. | PaktaniDigital | Web & Mobile |  |  |

Tabel 2. 2. Posisi Penelitian

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sistem informasi untuk Agriculture | Rancangan Desain Sistem | Web | Mobile | Referal Sistem | MVC |
|  |  |  |  |  |  |

# BAB III METODOLOGI PENELITIAN

Bab ini berisi tahapan yang dilakukan dalam penelitian diantaranya terhadap pengembangan sistem dengan metode SDLC (*System Development Life Cycle*) dengan model klasik yang biasa disebut dengan model *waterfall*. Rancangan penelitian yang menjelaskan mengenai cara menyelesaikan permasalahan, jenis penelitian yang dilakukan, rancangan analisis, tahapan pengumpulan data, tempat lokasi serta waktu dilaksanakannya penelitian..

## 3.1 Alir Tahapan Penelitian

Alir tahapan penelitian digunakan untuk memudahkan penulis untuk merancang aplikasi yang dibuat.

Metode pengembangan sistem yang penulis gunakan dalam penelitian ini adalah SDLC *(System Development Life Cycle).* Proses-proses pengembangan sistem ini dikenal dengan daur hidup pengembangan sistem yang memiliki beberapa tahapan. SDLC yang terkenal adalah SDLC model klasik yang biasa disebut dengan model *waterfall.* Alasan penulis menggunakan *waterfall* dikarenakan metode ini mempunyai tahapantahapan yang jelas, nyata dan praktis. Setiap tahap harus diselesaikan terlebih dahulu untuk menghindari terjadinya pengulangan dalam tahapan sehingga pengembangan sistem yang dilakukan dapat memperoleh hasil yang diinginkanAdapun tahapan-tahapan dalam pengembangan sistem menggunakan *waterfall* adalah sebagai berikut.

1. Perencanaan

Agar sistem yang akan dibuat sesuai dengan yang diinginkan, maka dibutuhkan perencanaan yang baik. Untuk itu, perencanaan merupakan langkah awal sebelum suatu sistem dibuat atau dikembangkan. Dalam tahap ini, penulis melakukan perencanaan dengan mengumpulkan data-data yang akurat, *valid* dan reliabel. Untuk pengumpulan datanya, penulis menggunakan beberapa metode yaitu, observasi, wawancara dan studi literatur.

1. Analisis Sistem

Setelah mengumpulkan beberapa data, penulis menganalisis sistem apa yang paling dibutuhkan oleh pihak petani, peternak dan pembeli dalam waktu dekat ini. Untuk itu penulis menganalisis sistem yang berjalan dan menganalisis sistem usulan yang dapat memberikan solusi dalam manajemen transaksi penjualan. Penjelasan analisis sistem berjalan dan sistem usulan penulis bahas di Bab IV.

1. Desain Sistem

Setelah kebutuhan dikumpulkan secara lengkap, penulis mengubah kebutuhan-kebutuhan tersebut dalam struktur data dengan menggunakan beberapa *tools* seperti UML (*Unified Modelling Language*) dan ERD (*Entity Relationship Diagram*)*.* Dalam tahap ini, penulis membuat 2 desain yaitu desain database dan desain interface. Untuk lebih jelasnya, dijelaskan di bab IV.

1. Implementasi Sistem (Kode Program)

Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Jika desain dilakukan secara lengkap, pembuatan kode dapat diselesaikan secara mekanis. Tahap kode dapat dilakukan setelah perencanaan, analisis, dan desain. Dalam penelitian ini, tahap disini dibuat dengan menggunakan bahasa pemograman PHP dalam bentuk *framework* guna menerapkan konsep MVC dan dan menggunakan *database* MYSQL.

1. Pengujian

Setelah perangkat lunak dibangun, maka dilakukan pengujian untuk menguji tingkat keberhasilan sistem yang telah dibangun. Hal ini dilakukan untuk memastikan kehandalan *software*.

1. Operasi Sistem atau Penerapan

Setelah sistem atau perangkat lunak diuji, dilakukan penerapan atau pengoperasian terhadap sistem yang telah dibuat atau dikembangkan. Selain pengoperasian, tahap ini juga dilakukan untuk mengontrol dan memperbaiki sistem yang telah dibangun.

1. Survei Pengguna

Tahapan ini bukan merupakan tahapan dari proses SDLC (*System Development Life Cycle*) dengan model *waterfall*. Namun tahapan ini menjadi sangat penting ketika penulis ingin mendapatkan data untuk membantu menjawab rumusan masalah penelitian. Survei ini menghasilkan data yang dapat dijadikan kesimpulan tentang kebermanfaatan sistem informasi dalam menyelesaikan kendala yang dialami Petani dan Peternak dalam melakukan proses penjualan dan hal terkait kebutuhan lainnya.

Gambar 3.1- Rancangan Penelitian



## 3.2 Rancangan Penelitian

### 3.2.1 Jenis Penelitian

Penelitian penulisan merupakan penelitian dengan pendekatan studi kasus maka dari itu penulis menggunakan jenis penelitian kualitatif-deskriptif. Studi kasus termasuk ke dalam penelitian analisis deskriptif, yaitu penelitian yang dilakukan proses analisis secara cermat dan tuntas. Penelitian ini tidak menggunakan data angka sehingga menggunakan pendekatan kualitatif. Metode pengembangan sistem penulis menggunakan pendekatan metode SDLC (*System Development Life Cycle*).

### Metode Pengumpulan Data

Sumber data dalam penelitian yang dilakukan oleh penulis dalam tugas akhir ini adalah data primer. Data primer adalah data yang didapatkan secara langsung oleh penulis, proses untuk mendapatkan data primer ini melalui observasi (pengamatan) dan melakukan proses wawancara terhadap user yang terlibat langsung dengan penggunaan aplikasi. Data-data ini sangat berguna bagi penulis dalam melakukan proses penelitian tugas akhir ini. Berikut ini adalah tahapan penulis dalam melakukan proses pengumpulan data

1. Observasi

Pada tahap ini dilakukan proses pengamatan terhadap proses bisnis yang dilakukan petani, peternak, pengusaha dan pembeli saat ini dan mengamati bagaimana permasalahan dalam manajemen dan proses transaksi guna memenuhi kebutuhan petani, peternak dan pembeli.

1. Wawancara

Pada tahap ini penulis melakukan proses wawancara atau tanya jawab dengan pihak petani, peternak dan pembeli guna mengetahui apa saja kendala yang dihadapi oleh mereka dalam proses kegiatan jual beli, masukkan dari para pihak tersebut akan sangat membantu dalam proses penggalian kebutuhan yang nantinya akan digunakan oleh penulis dalam penelitian.

### Tempat dan Lokasi Penelitian

Lokasi dari penelitian ini dilakukan di beberapa Peternakan dan Pertanian yang berada di Kota Depok dengan fokus pada studi kasus analisa dan perancangan prototipe sistem informasi jual beli bahan baku petani dan peternak. Waktu penelitian ini berlangsung sejak Mei 2017.

### Alat dan Bahan

Konfigurasi Sistem :

Sistem yang digunakan berbasis web dengan menggunakan *framework* (kerangka kerja), aplikasi nantinya akan bisa diakses via web browser.

Adapun perangkat keras yang digunakan untuk membangun sistem informasi manajemen Jual Beli Bahan Baku Petani dan Peternak ini antara lain :

1. Komputer dengan prosesor AMD A8
2. Kapasitas minimal RAM 2 GB
3. Kapasitas Hardisk minimal 500 GB
4. Sistem Operasi Windows 10 / Linux

Software yang digunakan :

1. *Apache Web Server* 2.x
2. *Minimal PHP 5.6*
3. *MySQL 5.5.x : PhpMyAdmin*
4. *Framework*
5. *IDE : Sublime Text 3*
6. *PHP Composer*
7. *Laravel Library :: Github Repository*

Alat desain yang digunakan :

1. *Enterprise Architect ( UML 2.0 )*
2. *Execute Query :: Query Designer*

### Metode Pengerjaan

Metode pengujian ini dilakukan untuk menguji dan menjawab perumusan masalah diatas dengan melakuan pengusian sebagai berikut :

1. Pengujian Menggunakan Aplikasi

Metode pengujian ini dengan melakukan wawancara feedback dari pengguna aplikasi.

1. Pengujian Fungsionalitas

Pada pengujian fungsionalitas penguji hanya berkepentingan dengan menggunakan Blackbox Testing.

# BAB IV ANALISIS DAN PERANCANGAN

Pada bab ini berisi mengenai proses dari bagian kegitan ini yang dilakukan pada tahapan analisis sistem dan design. Adapun kegiatan yang dilakukan adalah sebagai berikut.

1. Tahapan analisis sistem diantaranya analisa dari kondisi awal yang sedang berjalan saat ini dan penjelasan mengenai proses pendefinisian kebutuhan untuk sistem yang akan dibuat dengan menggunakan *use case diagram dan use case scenario*.
2. Tahapan design sistem diantaranya dilakukan proses perancangan sistem dengan menggukaan pendekatan UML antara lain : *Domain Model, Sequence diagram, Class Diagram,* juga dilakukan proses perancangan sistem *database* dengan menggunakan ERD(*Entity Relationship Diagram*) . Dan terakhir dilakukan perancangan antar muka sistem dari mulai struktur menu dan *mock-up* aplikasi.

### Tinjauan Umum Kondisi Awal

Dalam media digital dan media cetak kompas (07/17) menyatakan beberapa harga bahan pokok meningkat sedangkan petani pedesaan tak kunjung sejahtera karena ada beberapa pihak yang mendapat keuntungan dari hal tersebut.

"Program kementerian berfokus untuk memperpendek rantai pasok tata niaga pangan, membangun Toko Tani Indonesia, membentuk satgas pangan dan lainnya," kata Lutful Hakim.

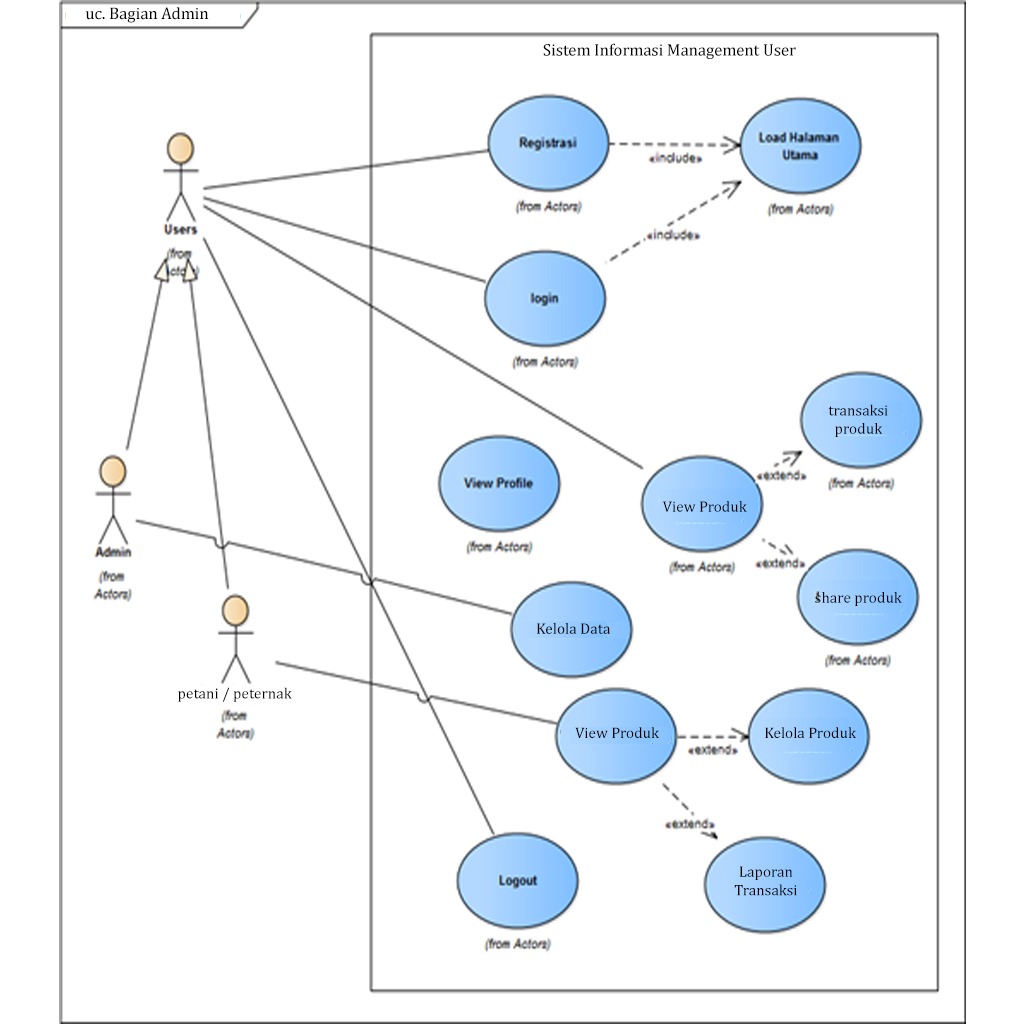
Permasalahan timbul saat ini dimana harga bahan pokok tidak mudah didapat, petani mengeluhkan penghasilan dari hasil panen menurut, rakyat mendapatkan dengan harga tinggi namun tengkulak mendapatkan keuntungan yang lebih tidak sebanding dengan petani untuk itu kami membuat sistem digital untuk memperpendek ratai distribusi yang terlalu panjang.

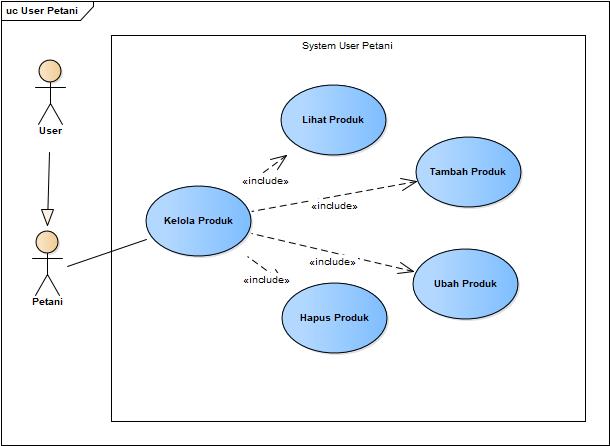
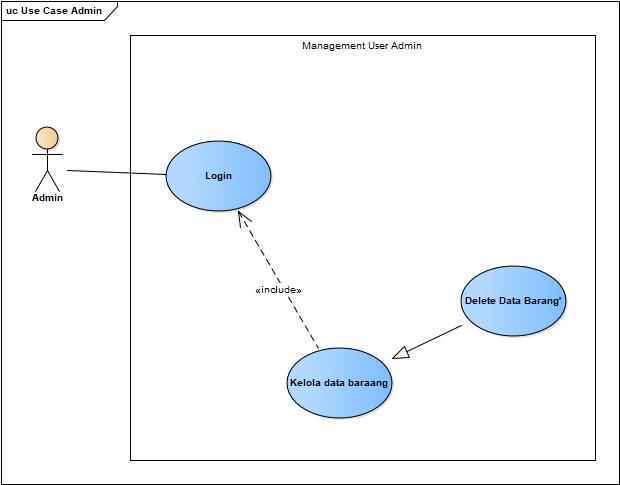
secara umum ada tiga pihak yang berperan dalam menjalankan program swasembada aplikasi ini yaitu petani, pelaku industri dan konsumen.

1. Petani : dapat memasukkan hasil panen beserta detail stok, jenis dan harga perkilo, melakukan transaksi dengan pelaku usaha maupun konsumen.
2. Pelaku Industri / Usaha : Menjadi mitra petani dimana pelaku usaha ini mendapatkan harga khusus dikarnakan melakukan pembeian hasil petani secara berkelanjutan, dalam hal ini pelaku industri / usaha dapat melakukan transaksi pembelian produk petani.
3. Konsumen : melakukan pengecekan harga pasar, petani di daerah sekitar dan daerah lain dan melakukan transaksi untuk pembelian produk hasil dari petani.

### Diagram Kebutuhan (*Use case diagram*)

Pada Gambar 4.1 dan Gambar 4.2 merupakan *use case diagram* proses manajemen jual beli petani dan peternak yang berhasil dikumpulkan berdasarkan keterangan *user* *requrment*.





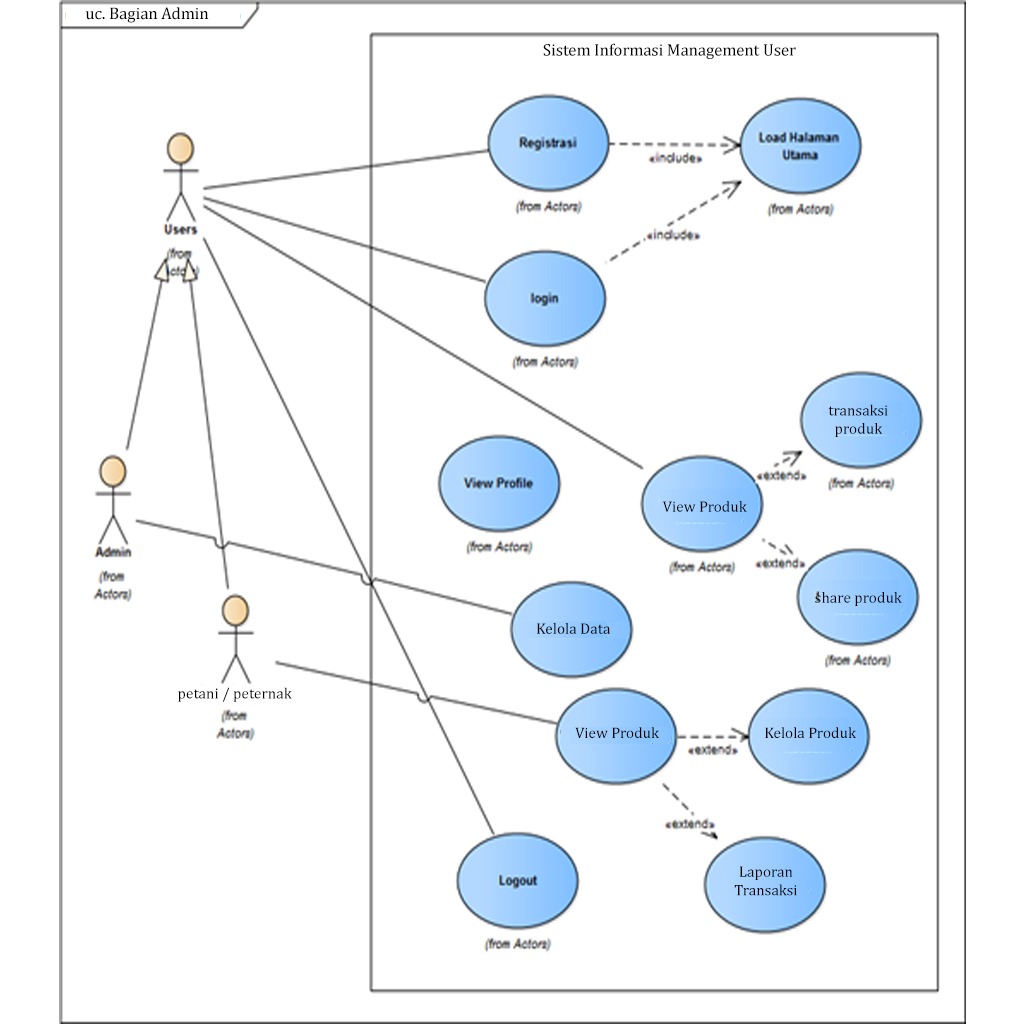
### Analisis Sistem

pada penelitian ini dilakukan proses pengumpulan requirment user dengan melakukan proses wawancara dengan beberapa orang yang berperan selaku petani, perkebunan, peternak, pelaku industri usaha dan konsumen untuk mengetahui kendala apa saja yang terjadi ketika proses pertanian di kota depok. diantara nya koordinator selsurh pertanian di kota depok

proses yang telah disebutkan diatas masuk dalam fase perancakangan dengan workflow utamananya pengumpulan data kebutuhan (requrment gethering) karena melihat penting nya program ini kami mencaoba mengidentifikasi permasalahan dengan peneyelenggaraan, idnetifkasi pada pertanian, perkebungan di sebagian kota Depok, antara lain sebagai berikut.

### Diagram Kebutuhan (*Use case diagram*)

Pada Gambar 4.1 dan Gambar 4.2 merupakan *use case diagram* proses manajemen jual beli bahan baku petani dan peternak kepada konsumen

.

### Deskripsi Kebutuhan (*Use case Scenario*)

Pada Tabel 4.1 merupakan *requirement user* untuk bisnis utamanya saja pada sistem menajemen Jual Beli Bahan Baku Petani dan Peternak, *requirement* ini diambil berdasarkandari kedua *use case diagram* yang telah dibuat sebelumnya yaitu *Use Case Diagram* Jual Beli Bahan Baku Petani dan Peternak pada Gambar 4.1 dan *Use Case Diagram* Nasabah pada Gambar 4.2, lalu ditambahkan kode pada tiap *requirement* yang berguna untuk pembuatan *use case scenario*. *Requirement* ini nantinya akan dijabarkan bagaimana *use case* skenarionya. *Requirement* ini diharapkan dapat membantu menjawab rumusan masalah pada penelitian ini.

Tabel 4. 1. Requirement Sistem Informasi Manajemen Jual Beli Bahan Baku Petani dan Peternak

|  |  |
| --- | --- |
| **Kode** | **Deskripsi** |
| K.1 | Registrasi |
| K.2 | Transaksi Barang |
| K.3 | Laporan Transaksi |
| K.4 | Forum Diskusi |
| K.5 | Mitra Usaha |

Berdasarkan pada Tabel 4.1 maka dibuatlah *use case scenario* dari masing-masing *requirement* di antaranya sebagai berikut :

1. *Use case scenario* Registrasi dijelaskan pada Tabel 4.2

Tabel 4. 2 *Use Case Scenario* Registrasi

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Use case* | Registrasi | | |
| *Requirement* Terkait | K.1 | | |
| Tujuan | Menyimpan data input dari aktor kedalam sistem sehingga aktor terdaftar sebagai konsumen, pengusaha atau petani sesuai *role* yang dipilih. | | |
| Kondisi Awal | Aktorsudah mengakses halaman utama dari sistem. | | |
| Kondisi Akhir – Sukses | Aktor berhasil terdaftar pada sistem dan terlihat halaman utama dengan foto *profile* serta nama terdaftar pada *navbar* | | |
| Kondisi Akhir – Gagal | Aktor gagal terdafftar pada sistem dan tetap pada halaman registrasi bisa juga terlihat terlihat *eror validation message* | | |
| *Primary actor* | Konsumen, Pengusaha dan Petani | | |
| *Secondary actor* | Admin | | |
| *Trigger* | Aktor menekan tombol register untuk memulai proses penyimpanan data *input* | | |
| *Main flow* | *Actor* *Action* | | *System* *Response* |
| 1. Aktor masuk ke dalam menu utama lalu menekan menu daftar sebagai konsumen, pengusaha ataupun petani pada *navbar* aplikasi *web.* | | 2. Sistem menampilkan halaman registrasi disertai *form input user* baik untuk data diri maupun lokasi aktor bekerja sebagai konsumen, pengusaha ataupun petani. |
| 3. Aktor mengisi *field* pada *form* tersebut antara lain nama, no.telepon, alamat e-mail, alamat, foto, password, konfirmasi password, mengisi *field* bank sampah jika registrasi sebagai bank sampah, kemudian menekan *select button* pada propinsi | | 4. Sistem akan menarik data propinsi dari *database*, kemudian menampilkan data pada *select button*. |
| 5. Aktor memilih data propinsi *select button* pada *field* propinsi | | 6. Sistem menampilkan data kabupaten pada *select button* pada *field* kecamatan |
| 7. Aktor memilih data propinsi *select button* pada *field* kabupaten*.* | | 8. Sistem menampilkan data bank sampah pada *select button* pada *field* kelurahan jika mendafar sebagai nasabah |
| 9. Aktor mengisi *field* RT dan RW ,memilih data kelurahan (jika daftar sebagai pengusaha dan petani). | | 10.Sistem menampilkan data kabupaten pada *select button* pada *field* kelurahan jika mendafar sebagai pengusaha ataupun petani. |
| 11 Aktor memilih data produk petani (jika daftar sebagai konsumen). Kemudian aktor menekan tombol register. | | 12. Sistem akan menyimpan data dan akan mengalihkan aktor ke halaman utama |
| *Extension* | *Step* | *Action* | |
| 3 | Jika aktor salah dalam menginput data form,maka akan muncul *eror validation message* pada tiap *inputan,* mengisi email untuk *field* alamat email tidak diwajibkan namun jika diisi harus sesuai format email yang benar. | |
| 12 | Jika pada saat proses *save* data terjadi kesalahan diluar *form validation*,maka akan terlihat *message eror* pada halaman registrasi. | |

1. *Use case scenario share* kegiatan dijelaskan pada Tabel 4.3

Tabel 4. 3. Use Case Scenario Share Kegiatan

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Use case* | Share Kegiatan | | |
| *Requirement* Terkait | K.2 | | |
| Tujuan | Untuk *sharing* informasi mengenai kegitan bank sampah ke sosial media | | |
| Kondisi Awal | Aktor sudah mengakses halaman kegiatan pada aplikasi *web*. | | |
| Kondisi Akhir – Sukses | Aktor berhasil *sharing link* yang merujuk pada halaman sosial media | | |
| Kondisi Akhir – Gagal | Aktor gagal *sharing*  data *link* kegiatan pertanian pada halaman sosial media | | |
| *Primary actor* | Konsumen, Pengusaha dan Petani | | |
| *Secondary actor* | Admin | | |
| *Trigger* | Aktor menekan tombol sismbol *icon* sosial media kemudian sisem akan mengarahkan ke halaman sosial media dengan link url adri aplikasi *web.* | | |
| *Main flow* | *Actor* *Action* | | *System* *Response* |
| 1. Aktor menekan *icon* sosial media pada halaman kegiatan yang diinginkan. | | 2. Sistem mengarahkan pengguna ke halaman sosial media yang *icon*nya ditekan aktor. |
| 3. Kemudian sistem juga akan mengirim *url link* dari halaman kegiatan di aplikasi *web* pada halaman sosial media. |
| *Extension* | *Step* | *Action* | |
|  |  | |

## Perancangan Sistem

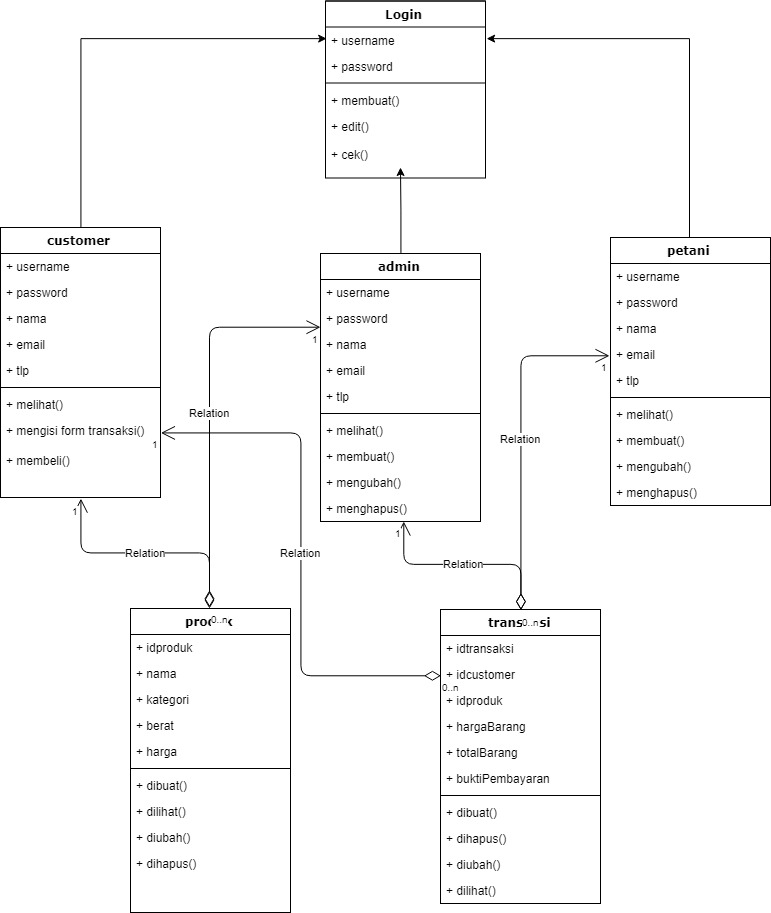


### Desain Sistem

Desain sistem yang dibuat pada penelitian ini antara lain *domain model, sequence diagram dan class diagram.* Jika pada proses pembuatan *use case diagram* dan *use case scenario* dibuat berdasarkan pandangan *user*, sekarang masuk pada tahap pengembangan desain lebih lanjut lagi dengan sudut pandang sistem. Desain sistem yang telah dibuat adalah sebagai berikut :

1. *Domain model*

*Domain model* memiliki kegunaan untuk menunjukan hubungan yang mungkin ada di antara objek-objek yang sangat kompleks, secara umum objek diagram berisi dengan objek dan *link* [39]. Pada Gambar 4.3 menggambarkan keterhubungan antar objek dari sistem yang akan buat, keterhubungan antar objek tersebut dihubungkan dengan *link* berupa garis disertai informasi *multiplicity* pada masing-masing objek.

****

Gambar 4. 1. Domain Model Sistem Informasi Jual Beli Petani dan Peternak

1. *Sequence diagram*

*Sequence diagram* digunakan untuk mengambarkan prilaku pada skenario yang telah dibuat. Pada *sequence diagram* ini memiliki komponen utama yang terdiri atas objek yang digambarkan dengan kotak persegi yang bernama, *messege* (pesan) yang digambarkan dengan garis tanda panah, serta waktu yang ditunjukan dengan garis terputus vertikal [39].

*Sequence diagram* bersama *collaboration diagram* termasuk kedalam jenis diagram interaksi yang menggambarkan interaksi antara objek – objek dalam sistem berdasarkan *use case. Sequence diagram* diurutkan berdasarkan waktu sedangkan *collaboration diagram* lebih berkonsentrasi pada hubungan antara objek – objek pada sistem. Meskipun begitu baik *sequence diagram* maupun *collaboration diagram* menampilkan informasi yang sama. [40]

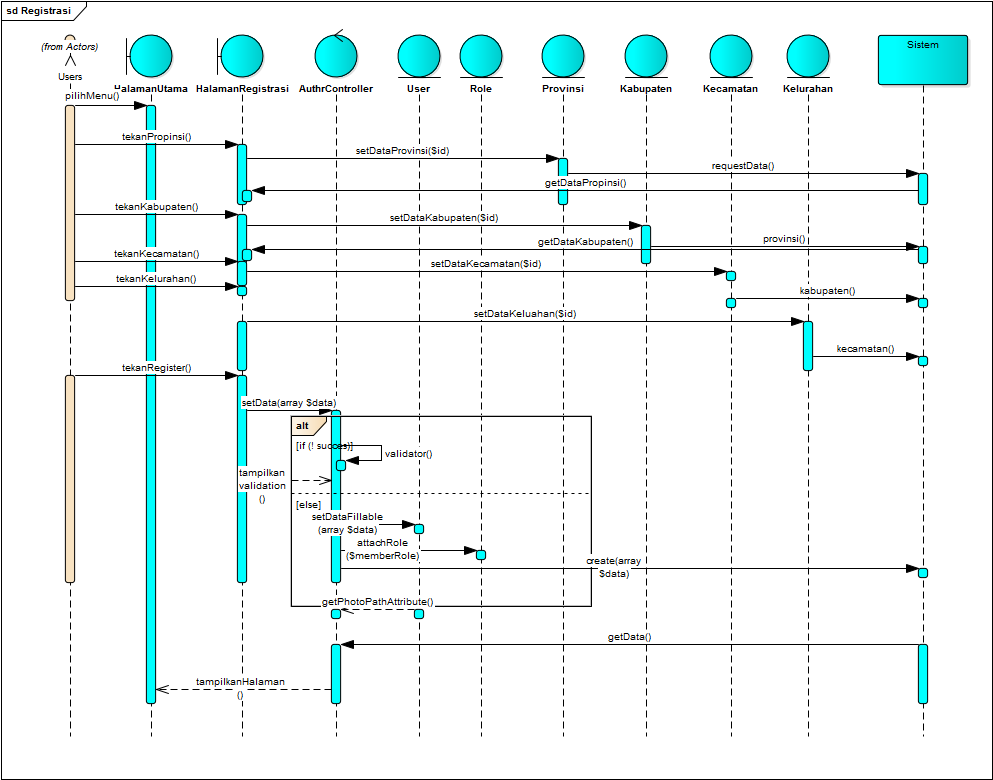
Skenario yang sebelumnya telah dibuat pada *use case scenario*, lalu dikembangkan lagi ke dalam bentuk *sequence diagram*. Jika pada *use case scenario* tidak dideskripsikan objek-objek apa saja yang terlibat dalam sistem, maka pada *sequence diagram* bisa terlihat objek-objek apa saja yang terlibat dalam skenario yang telah dibuat. Berikut ini gambaran *sequence diagram* yang telah dibuat :

1. Pada Gambar 4.4 merupakan *sequence diagram* dari proses buat registrasi*,* dengan terdapat 6 *class entity* : User, *Role*, Provinsi ,Kabupaten, Kecamatan, dan Kelurahan. Serta terdapat 2 *boundary* : HalamanUtama dan HalamanRegistrasi Terlihat banyak pesan yang dilewatkan oleh tiap *participants,* dan objek aktif (aktor) yang menginisiasi proses regisrasi untuk semua *user*. Terdapat pula satu buah *controller class* yaitu *AuthController* yang digunakan untuk proses registrasi dan autentifikasi.

Pada Gambar 4.4 terdapat beberapa pesan yang tidak hanya sebagai pesan interaksi tapi juga menunjukkan method yang akan digunakan dalam proses implementasi selanjutnya diantaranya.

* kabupaten(),kecamatan(), kelurahan() menunjukkan method relasi.
* *attachRole()* menunjukka nmethod untuk *setting* hak akses.
* *create()* method untuk *insert* data.

Terdapat juga *alternate* jika *input* data pada form sesuai dengan *validasi* maka sistem memproses penyimpanan datadan sebalikanya jika gagal, maka sistemakan menampilkan pesan eror.

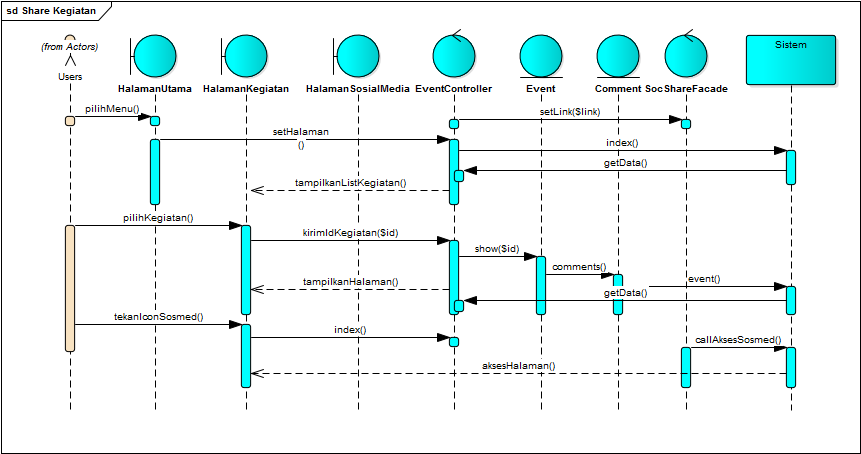


Gambar 4. 2 Sequence diagram Dari Proses Registrasi

1. Pada Gambar 4.5 merupakan *sequence diagram* dari proses *share* kegiatan*,* serta terdapat 3  *boundary* : HalamanUtama, HalamanKegiatan dan juga HalamanSosialMedia. Terdapat pula dua *entity class* yaitu *Event* dan *Comment.* Selain itu terdapat satu *controller class* yaitu *EventController* dan satu *library class* yaitu *SocShareFacade.*

Pada Gambar 4.5 terdapat beberapa pesan yang tidak hanya sebagai pesan interaksi tapi juga menunjukkan method yang akan digunakan dalam proses implementasi selanjutnya diantaranya.

* *event()* menunjukkan method relasi antara *Event* dan *Comment class* .
* *show()* dan *index()* method untuk  *view* data.



Gambar 4. 3. Sequence diagram Dari Proses Share Kegiatan

1. Pada Gambar 4.6 merupakan *sequence diagram* dari proses menimbang*,* serta terdapat 4 *boundary* : HalamanUtama, HalamanDataNasabah, HalamanFormTimbangan dan juga HalamanPdf. Terdapat pula tiga *entity class* yaitu *User,* Tabungandan Sampah*.* Selain itu terdapat dua *controller class* yaitu NasabahSaldo*Controller* dan Laporan*Controller* sertasatu *library class* yaitu *LibraryPdf.*

Pada Gambar 4.6 terdapat beberapa pesan yang tidak hanya sebagai pesan interaksi tapi juga menunjukkan method yang akan digunakan dalam proses implementasi selanjutnya diantaranya.

* *event()* dan *saveData()* menunjukkan method menyimpan data saldo.
* *show()* method untuk  *view* data.

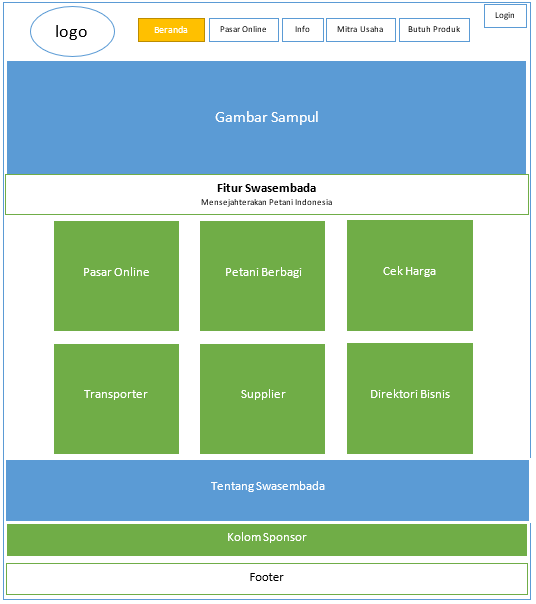
1. *Mockup* Aplikasi­­

Rancangan *mockup* aplikasi dibuat untuk bisa memberi gambaran terhadap aplikasi yang nantinya akan dibangun. Berikut ini rancangan *user interface* yang telah dibuat :

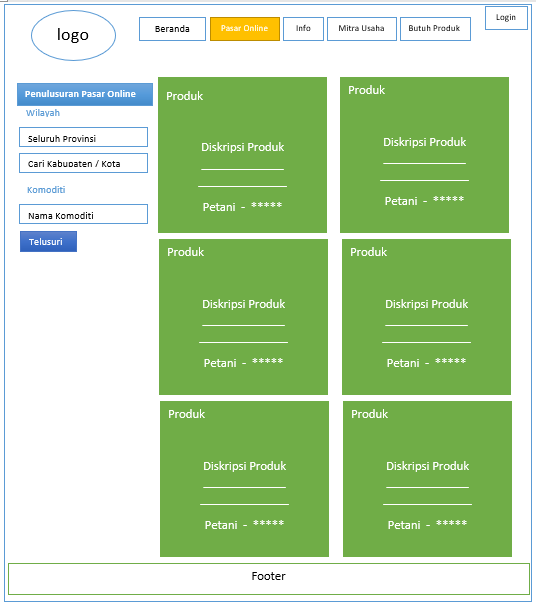
Pada Gambar 4.14 merupakan halaman *Home* dimana ketika *user* mengakses *website* ini halaman inilah yang pertama kali muncul. Pada halaman ini juga dilengkapi dengan *form login*  bagi *user* yang akan masuk ke dalam modul keuangan sesuai dengan role masing-masing, *form* tersebut terletak di sebelah kanan yaitu pada menu masuk.

**Mockup Website Swasembada**

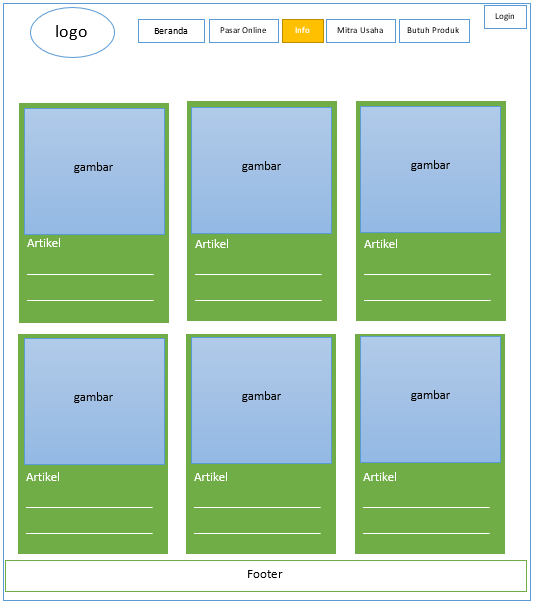
Halaman homepage website swasembada



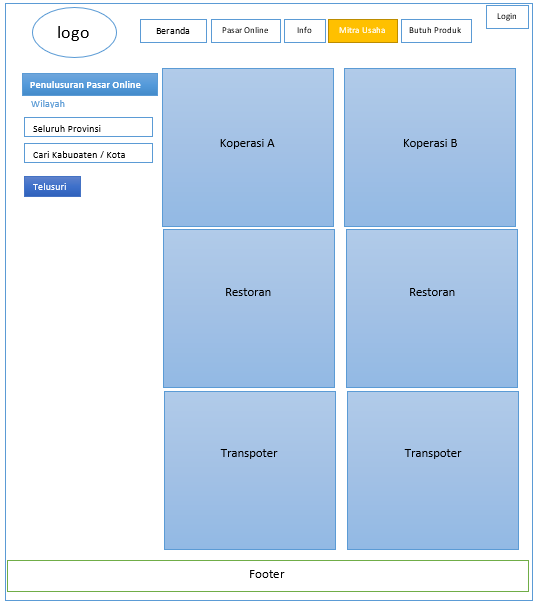
Tampilan menu pasar online



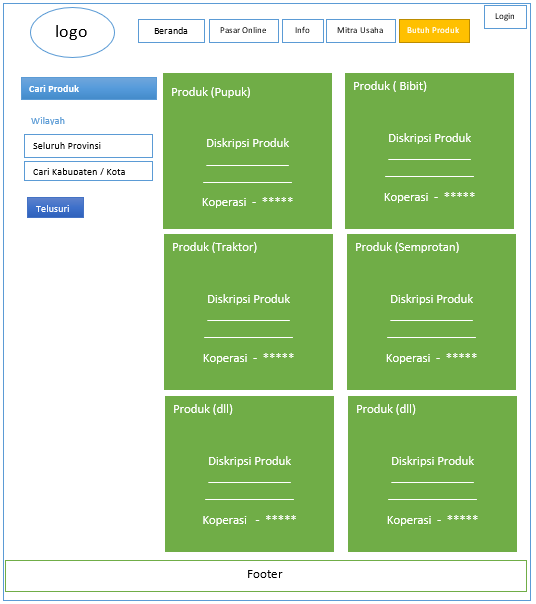
Tampilan untuk menu info / artikel yang ada di website swasembada



Tampilan menu mitra usaha



Tampilan menu butuh produk

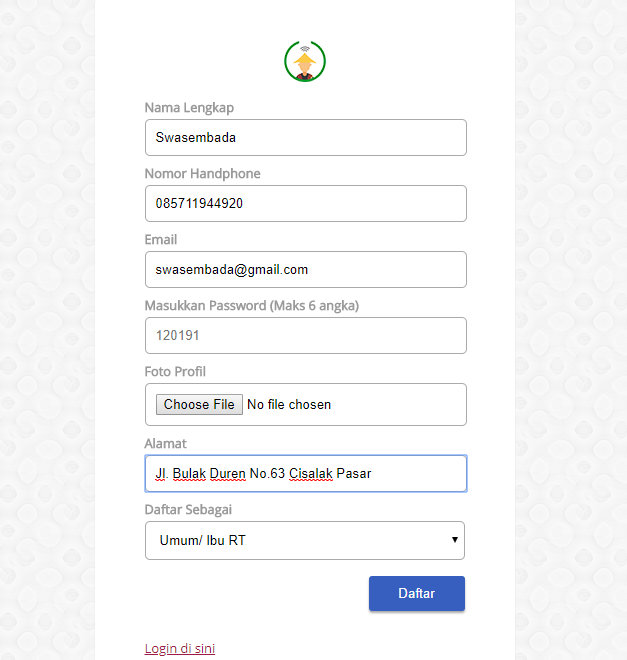


Tampilan registrasi

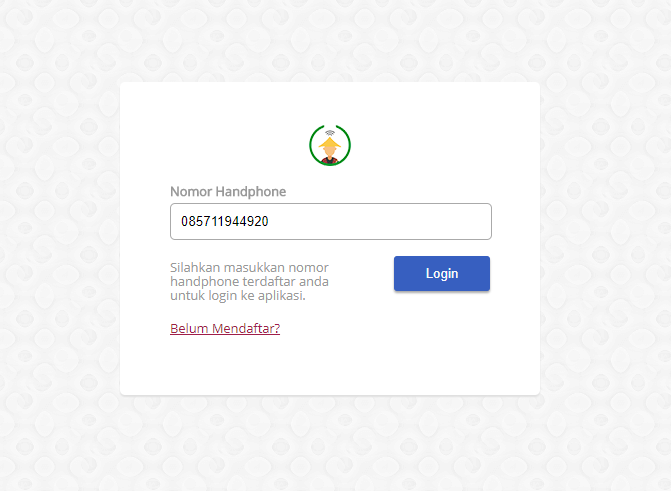


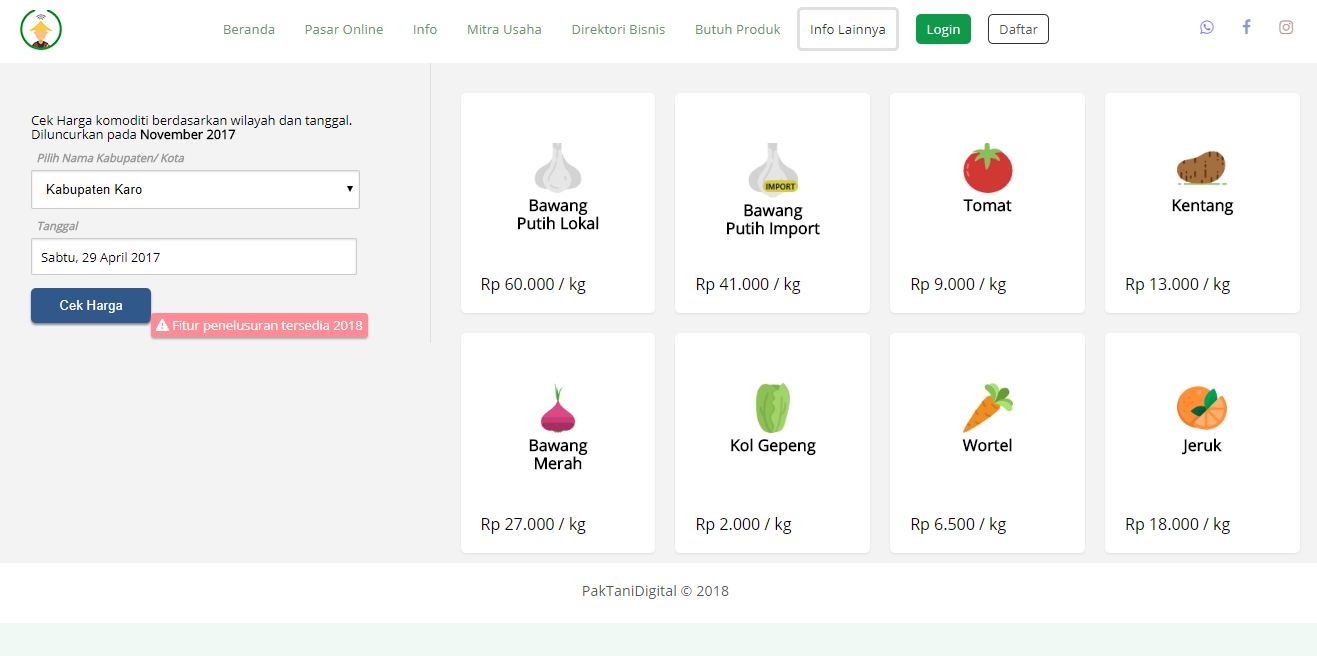
Tampilan login

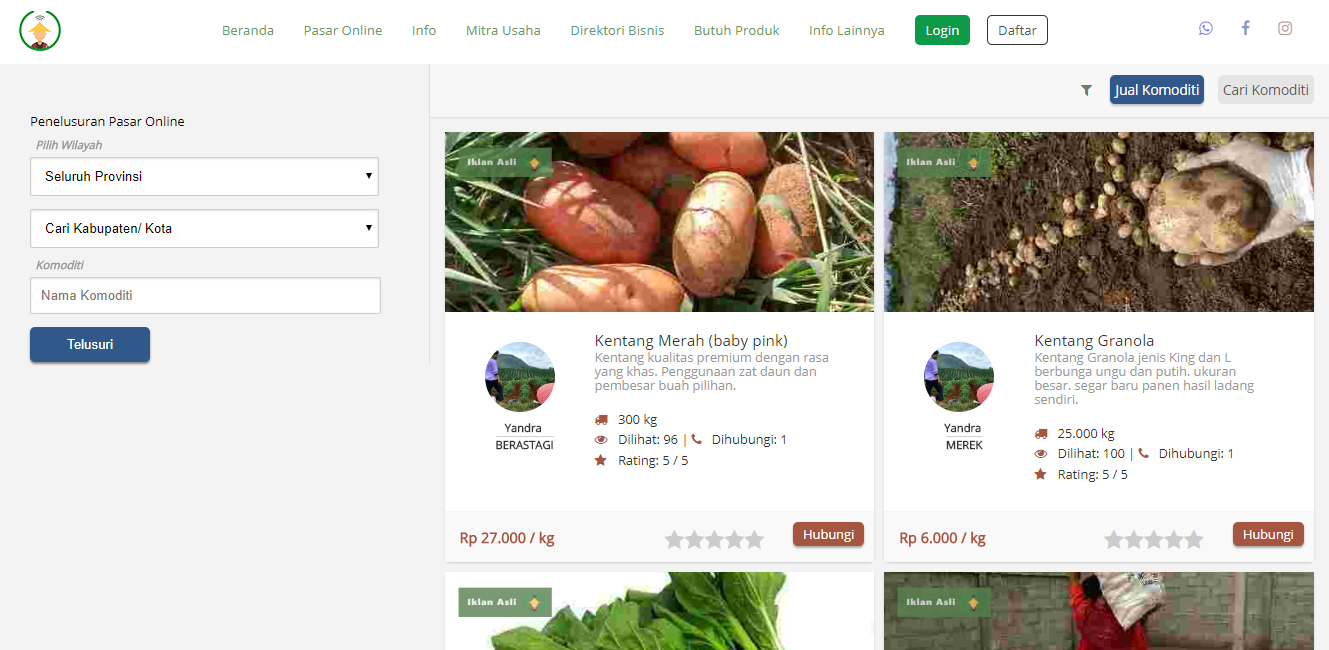


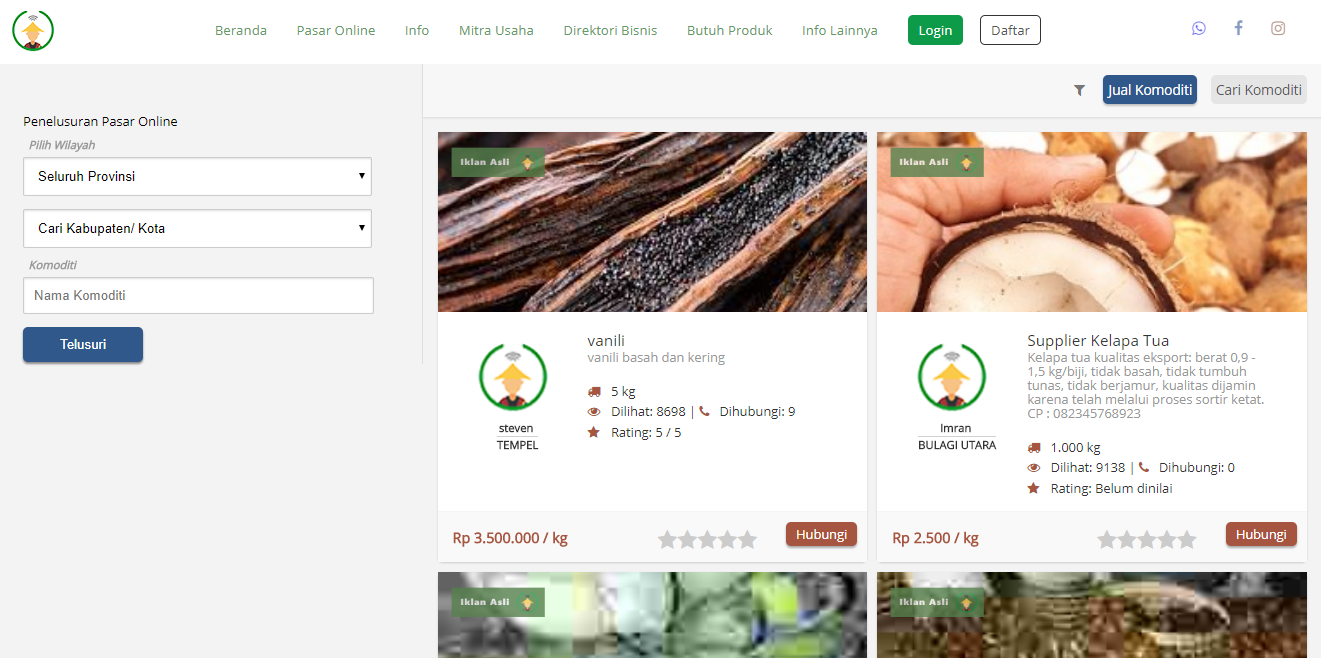


Login website swasembada

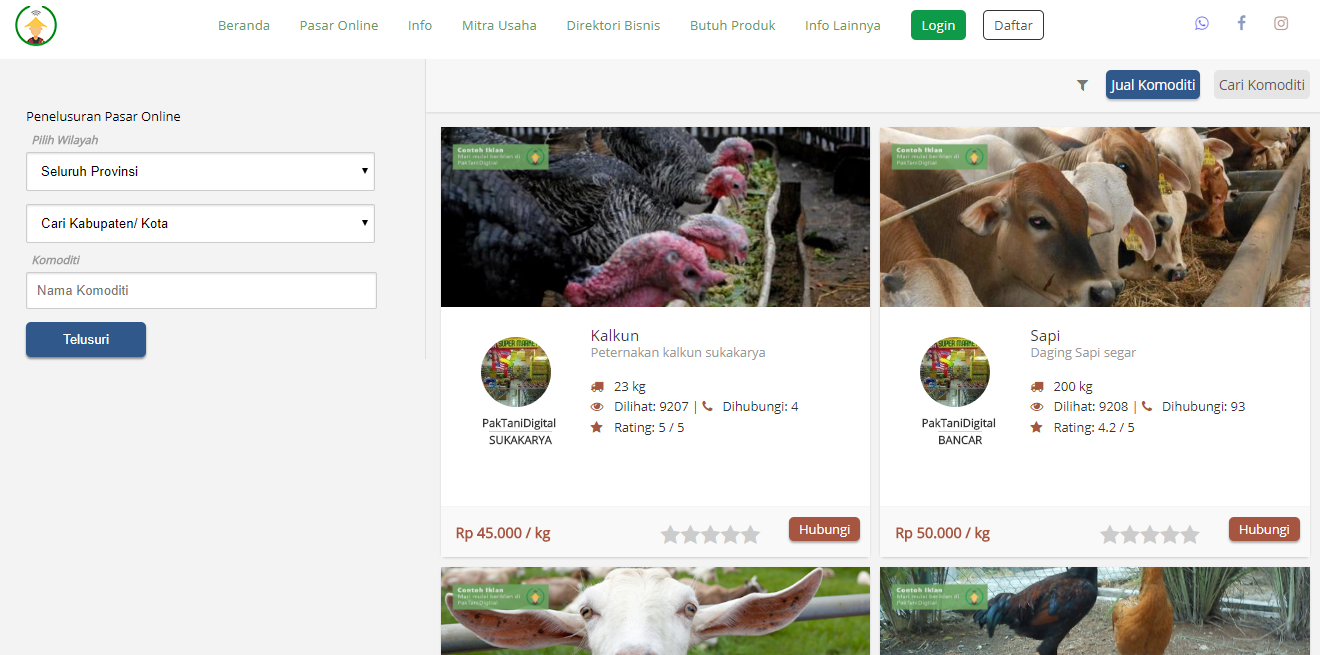


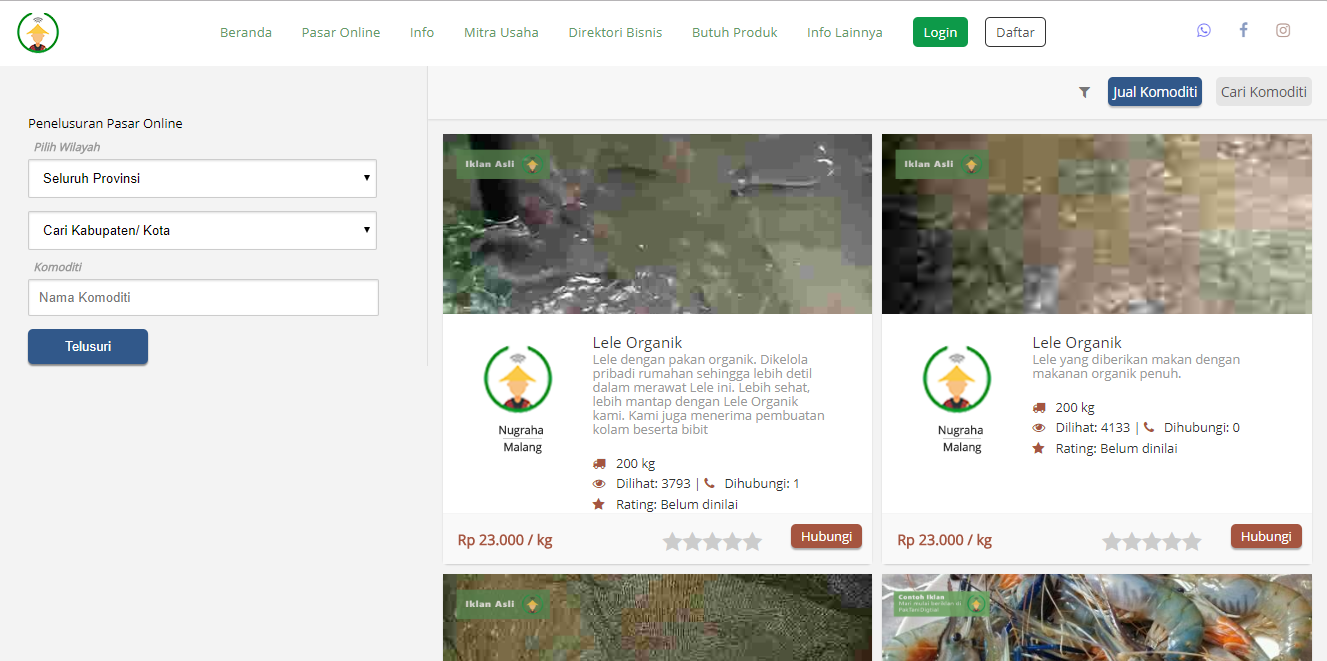




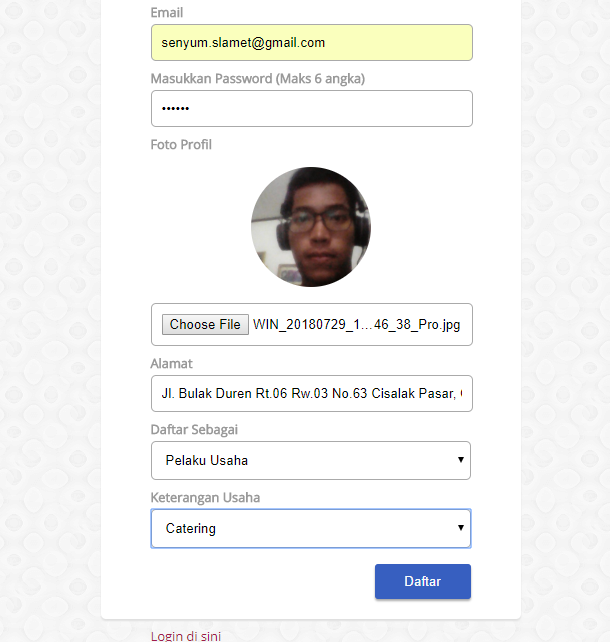


Perkebungan





Perikanan



# BAB V IMPLEMENTASI DAN EVALUASI

Pada bab ini berisi pembahasan mengenai implementasi dari sistem yang dibuat serta dilakukan proses evaluasi di dalamnya. Proses implementasi yang dibahas antara lain mengenai alur kerja dari system. Sedangkan proses evaluasi yang terdiri dari proses *testing* dengan menggunakan *User Acceptance Test* (UAT) testingdan survei yang akan membantu menjawab perumusan masalah pada bab sebelumnya.

## Implementasi Penggunaan *Framework*

Pada tahap implementasi ini merupakan bagian dari fase *coding* dengan *workflow* utamanya adalah pemilihan *framework* yang akan digunakan dalam pengembangan aplikasi web manajemen bank sampah. Salah satu pertimbangan penulis dalam pemilihan *framework* adalah bahasa pemrograman. Bahasa yang digunakan *framework* adalah bahasa pemrograman php. Pertimbangan menggunakan *framework* dengan bahasa pemrogaman php adalah sebagai berikut.

1. Mempercepat dalam pengembangan web.
2. Memberikan kode yang terorganisir dengan baik, kode dapat digunakan dan dipelihara kembali.
3. Memungkinkan penulis untuk tumbuh dari waktu ke waktu sebagai aplikasi web yang berjalan pada kerangka kerja yang terukur.
4. Keamanan sistem sudah terhandel oleh *framework* php.
5. Code dan file teroganisir dengan baik.
6. Sebagai praktek pengembangan web modern seperti pemrograman berorientasi obyek.
7. Menggunakan pola MVC (*Model-View-Controller*) yang menjamin pemisahan antara presentasi dan logika.
8. Terdapat banyak komunitas pengembang.

Untuk *framework* php yang digunakan penulis dalam pengembangan aplikasi ini yaitu Laravel. Laravel menjadi primadona sebagai *framework* terpopuler tahun ini sebagaimana yang ada pada beberapa website diantaranya <http://noeticforce.com/best-php-frameworks-for-modern-web-development> dan <http://www.hongkiat.com/blog/best-php-frameworks/>, serta masih banyak lagi website yang memberikan informasi yang sama. Beberapa kelebihan laravel adalah sebagai berikut. [41]

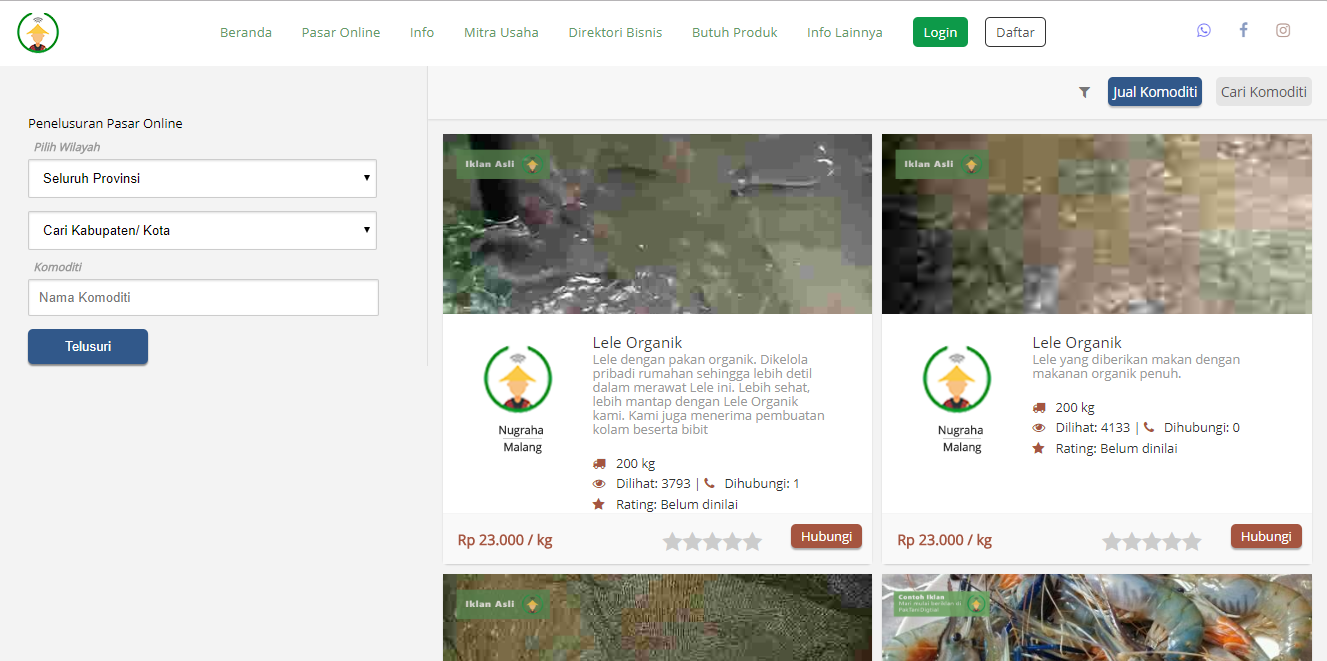
1. Menggunakan PHP Versi terbaru : Versi PHP 5.3 keatas memiliki cukup banyak fitur baru yang membuat PHP lebih power full dan modern.
2. Syntax yang *Cool & Expressive* : Laravel dirancang untuk memudahkan pengembang, bahkan untuk tahap pemula pun yang notaben belum paham tentang pemrograman dapat dengan mudah memahami alur dan pengkodean laravel.
3. Adanya *composer* yang merupakan sebuah *dependency* manager untuk bahasa pemrograman PHP. Dengan fasilitas ini kita dapat dengan mudah membuat *library* secara otomatis.
4. *Official website* yang bagus dan memudahkan kita dalam mencari informasi.
5. Dokumentasi lengkap
6. Komunitas yang banyak dan sedang populer tahun ini.
7. Dilengkapi dengan Aplikasi *Eloquent ORM.*
8. *Blade Templating*.

Dan masih banyak lagi kelebihan – kelebihan lain dari laravel yang sobat bisa dapatkan datanya dari internet dan sumber lainnya. Laravel sekarang sudah memasuki versi 5. Laravel 5.1 menjadi versi *Long Term Support* (LTS) untuk 3 tahun mendatang sehingga akan aman dalam menggunakan arena mendapatkan dukungan perbaikan bugs dari tim laravel. Hingga kini laravel semakin handal pada versi terbarunya 5.4. Laravel versi 5.4 lebih handal dengan berbagai fitur baru lainnya. Pada penngembangan aplikasi ini penulis menggunakan Laravel versi 5.2.

## Implementasi Antar Muka

Pada tahap implementasi ini, dilakukannya proses pembuatan kode program untuk mewujudkan desain yang telah dibuat sebelumnya pada fase desain sistem untuk menu utama saja diantaranya sebagai berikut.

Pada Gambar 5.1 merupakan tampilan halaman *home*



Gambar 5. 1. Tampilan Halaman Home

Deskripsi :

Halaman ini merupakan halaman *home* dari aplikasi manajemen bank sampah, ketika *user* mengakses url aplikasi maka halaman inilah yang akan pertama kali muncul

# BAB VI KESIMPULAN DAN SARAN

Pada bab ini berisi mengenai kesimpulan yang ditarik dari keseluruhan proses yang dilakukan dalam penelitian ini serta terdapat saran yang diberikan untuk menjadi masukan bagi pengembangan lebih lanjut.

## Kesimpulan

Pada penelitian yang telah dilakukan, dapat ditarik kesimpulan bahwa pengembangan prototipe sistem informasi berbasis web menggunakan *framework* dengan pola *model - view – controller* untuk manajemen bank sampah dapat menjawab rumusan masalah yang telah didefinisikan sebelumnya yaitu :

1. Dengan menggunakan sistem informasi pengelolaan bank sampah berbasis web dapat meningkatkan minat masyarakat dalam keikutsertaan program bank sampah. Hal tersebut terlihat dari data persentasi yang tinggi pada minat untuk ikut serta dari non nasabah yang mayoritasnya adalah masyarakat sekitar tempat bank sampah dan belum menjadi nasabah.
2. Sistem informasi bank sampah berbasis web juga terbukti dapat membantu memudahkan pihak nasabah mengetahui saldo tabungan dan memudahkan dalam pengembaliannya. Kesimpulan ini diambil dari hasil presentasi yang cukup tinggi dari respon positif *user* nasabah dalam penggunaan fitur tarik dana dan *profile* pada sistem.
3. Dengan sistem informasi pengolahan bank sampah berbasis web juga membantu pihak bank sampah dalam memasukkan data transaksi dan sosialisasi program bank sampah. Hal ini dapat disimpulkan dari hasil persentasi yang tinggi dari kuesioner *user* bank sampah dalam penggunaan fitur input timbangan dan *share* kegiatan.

.

## Saran

Pada penelitian ini telah dilakukan proses analisis dan perancangan prototipe sistem informasi berbasis web menggunakan framework dengan pola *model - view – controller* untuk manajemen bank sampah, analisis telah dilakukan secara menyeluruh terhadap seluruh kebutuhan amanejmen bank sampah walaupun proses tersebut tidak didefinisikan pada batasan masalah terkait fitur yang akan diimplementasikan, diharapkan dokumentasi terkait analisis dan desain yang telah dirancang dapat menjadi acuan untuk pengembangan selanjutnya. Untuk fitur penarikan sebaiknya bisa dikembangkan lagi agar bisa mengalihkan penarikan dana untuk keperluan lain seperti pembelian pulsa, token listrik atau sembako. Aplikasi ini sebaiknya bisa dibuatkan aplikasi *mobile* agar pengguna bisa memperoleh notifikasi terkait informasi bank sampah secara *real-time* dan lebih mudal menggunakannya.

# DAFTAR PUSTAKA

Andi. Kristanto. (2004). Model Waterfall.

D.Rustandi. (2009). Perancangan Sistem Informasi Akademik Berbasis Web di Akademik Keperawatan Rumah Sakit Hasan Sadikin Bandung. JBPTUNIKOMPP, vol. I, no.1, p.8.

Farhan, M. (n.d.). SISTEMINFORMASILEMBAGAPENELITIANDANPENGABDIAN KEPADAMASYARAKAT(LPPM)STMIKPALCOMTECH PALEMBANGBERBASISWEB. STMIK PalCom Tech Palembang.

H. Akik Hidayata, A. W. (2017). ABDIMAS LPPM STMIK DCI . JURNAL MANAJEMEN INFORMATIKA Vol 06 No 01 .

Jogiyanto. (2003). Analisis dan Sistem Informasi. Yogyakarta:

Murdick, R. (1993). Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML. Yogyakarta: Andi.

Myer, T. (2008). Professional Codeigneter (Wrox professional guide). wrox.

Nugroho, A. (2009). Rekayasa Perangkat Lunak Menggunakan UML dan Java. Yogyakarta: Andi Offset.

Nurdiansyah, D. (2016). SDLC : Pengertian Waterfall Model | Keuntungan - Kelemahan.