**IMPLEMENTASI SISTEM INFORMASI KEUANGAN DENGAN METODE PERPETUAL DAN PERIODIK   
PADA CV PANEN REJO SEMARANG**

# **TUGAS AKHIR**



**DISUSUN:**

**MUHAMMAD AGUS NADHIF**

**G.211.13.0019**

**PROGRAM STUDI SI – TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI**

**UNIVERSITAS SEMARANG**

**2017**

# **PERNYATAAN PENULIS TUGAS AKHIR**

**DENGAN JUDUL**

**IMPLEMENTASI SISTEM INFORMASI KEUANGAN DENGAN METODE PERPETUAL DAN PERIODIK**

**PADA CV PANEN REJO SEMARANG**

Dengan ini saya :

NAMA : MUHAMMAD AGUS NADHIF

NIM : G.211.13.0019

PROGRAM STUDI : TEKNIK INFORMATIKA

“Saya menyatakan dan bertanggungjawab dengan sebenarnya bahwa Tugas Akhir (TA) ini adalah hasil karya saya sendiri kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya. Jika pada waktu selanjutnya ada pihak lain yang mengklaim bahwa Tugas Akhir (TA) ini sebagai karyanya, yang disertai dengan bukti-bukti yang cukup, maka saya bersedia untuk dibatalkan gelar Sarjana Komputer saya beserta segala hak dan kewajiban yang melekat pada gelar tersebut”.

Semarang, Agustus 2017

Penulis

Muhammad Agus Nadhif

NIM.G.211.13.0019

# **PERSETUJUAN TUGAS AKHIR**

**DENGAN JUDUL**

**IMPLEMENTASI SISTEM INFORMASI KEUANGAN DENGAN METODE PERPETUAL DAN PERIODIK**

**PADA CV PANEN REJO SEMARANG**

OLEH

NAMA : MUHAMMAD AGUS NADHIF

NIM : G.211.13.0019

DISUSUN DALAM RANGKA MEMENUHI SYARAT GUNA

MEMPEROLEH GELAR SARJANA KOMPUTER

PROGRAM STUDI S1 – TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

FAKULTAS TEKNOLOGI INGORMASI DAN KOMUNIKASI

UNIVERSITAS SEMARANG

TELAH DIPERIKSA DAN DISETUJUI

SEMARANG,

KETUA PROGRAM STUDI PEMBIMBING TUGAS AKHIR   
 S1-TEKNIK INFORMATIKA

April Firman Daru, S.Kom, M.Kom Whisnumurti Adhiwibowo,S.T.,M.Kom   
 NIS. 06557003102133 NIS. 06557003102137

# **PENGESAHAN TUGAS AKHIR**

**DENGAN JUDUL**

**IMPLEMENTASI SISTEM INFORMASI KEUANGAN DENGAN METODE PERPETUAL DAN PERIODIK**

**PADA CV PANEN REJO SEMARANG**

OLEH

NAMA : MUHAMMAD AGUS NADHIF

NIM : G.211.13.0019

Telah diujikan dan dipertahankan dihadapan Dewan Penguji pada Sidang Tugas Akhir (TA) Hari ………… tanggal …………………… pandangan kami, Tugas Akhir ini memadai dari segi kualitas maupun kuantitas untuk tujuan penganugerahan gelar Sarjana Komputer (S.Kom)

Semarang, ……………………

Dewan penguji,

KETUA TIM PENGUJI

……………………………………

NIS.

PENGUJI PENDAMPING I PENGUJI PENDAMPING II

……………………………….. ………………………………

NIS. NIS.

# **ABSTRACT**

# **ABSTRAK**

# **KATA PENGANTAR**

Puji syukur penulis panjatkan kepadaTuhan Yang Maha Esa Allah SWT yang telah melimpahkan Rahmat, Taufik serta Hidayah-Nya, sehingga penulis dapat melaksanakan Kerja Praktek serta dapat menyelesaikan laporannya tepat waktu dan tanpa adanya halangan apapun. Dalam pelaksanaan kerja praktek dan penyusunan laporan, penulis mendapat banyak bantuan, dukungan dan bimbingan dari berbagai pihak. Oleh Karena itu dalam kesempatan ini penulis ingin menyampaikan ucapan terimakasih kepada:

1. Bapak Prof. Dr. H. Pahlawansjah Harapan, SE, ME selaku Rektor Universitas Semarang.
2. Bapak Susanto, S.Kom, M.Kom. selaku Dekan Fakultas Teknologi Informasi Dan Komunikasi Universitas Semarang.
3. Ibu Nur Wakhidah, S.Kom., M.Cs. selaku Ketua Prgram Studi S1 – Teknik Informatika Fakultas Teknologi Informasi Dan Komunikasi Universitas Semarang.
4. Ibu Nurtriana Hidayati, S.Kom., M.Kom. selaku dosen pembimbing yang telah membantu dan memberikan bimbingan kepada penulis sehingga penulis dapat menyesesaikan laporan ini.
5. Keluarga, Teman dan pihak CV. Panen Rejo yang selalu memberikan dukungan, arahan dan bimbingan.

Akhir kata penulis menyadari bahwa pelaksanaan Tugas Akhir dan penyusunan laporan ini masih belum sempurna. Oleh karena itu, kritik dan saran sangat diharapkan, mohon maaf atas kesalahan penulisan kata dan kalimat. Semoga penyusunan laporan ini bermanfaat.

Semarang,

Penulis

# **DAFTAR ISI**

Halaman Judul ………………………………………………………………. i

Halaman Pengesahan ………………………………………………………... ii

Kata Pengantar ………………………………………………………………. iii

Daftar Isi …………………………………………………………………….. iv

Daftar Gambar ………………………………………………………………. vi

Daftar Tabel …………………………………………………………………. viii

BAB I PENDAHULUAN ……………………………………………… 1

1. Latar Belakang …………………………………………………….. 1
2. Batasan Masalah …………………………………………………... 1
3. Tujuan Kerja Praktek ……………………………………………… 2
4. Metodologi Pelaksanaan …………………………………………... 2
5. Sistematika Penulisan ……………………………………………... 3

BAB II TINJAUAN UMUM …………………………………………… 5

1. Sejarah CV Tunas Agri Farm ……………………………………... 5
2. Visi, Misi, dan Value CV Tunas Agri Farm ………………………. 5
3. Tentang CV Tunas Agri Farm …………………………………….. 6
4. Struktur Organisasi CV Tunas Agri Farm ………………………… 9
5. Tugas dan Wewenang Masing-Masing Jabatan …………………... 10

BAB III LANDASAN TEORI ………………………………………….. 12

1. Pengertian Aplikasi ……………………………………………….. 12
2. Inventory atau Persediaan ………………………………………… 12
3. Penjualan ………………………………………………………….. 13
4. United Modeling Language (UML) ………………………………. 15
5. Waterfall …………………………………………………………... 23

BAB IV ANALISA DAN PERANCANGAN …………………………... 25

1. Analisa Sistem yang Berjalan ……………………………………... 25
2. Analisa Kebutuhan ………………………………………………... 25
3. Analisa Desain …………………………………………………….. 26
4. Perancangan Database …………………………………………….. 39
5. Perancangan Antar Muka …………………………………………. 41

BAB V PENUTUP ……………………………………………………... 52

1. Kesimpulan ………………………………………………………... 52
2. Saran ………………………………………………………………. 52

DAFTAR PUSTAKA ………………………………………………………. 53

# **DAFTAR GAMBAR**

Gambar 2.1 Peta Lokasi CV Tunas Agri Farm ……………………………… 6

Gambar 2.2 Gerbang Toko CV Tunas Agri Farm …………………………... 7

Gambar 2.3 Halaman Toko CV Tunas Agri Farm ………………………….. 7

Gambar 2.4 Toko CV Tunas Agri Farm …………………………………….. 8

Gambar 2.5 Sarana Tanaman Perkebunan CV Tunas Agri Farm …………... 8

Gambar 2.6 Prasarana Pertanian dan Perkebunan CV Tunas Agri Farm …... 9

Gambar 2.7 Struktur Oraganisasi CV Tunas Agri Farm ……………………. 9

Gambar 3.1 Metode *Waterfall* ………………………………………………. 23

Gambar 4.1 *Use Case Diagram* ……………………………………………... 27

Gambar 4.2 *Activity diagram* mengelola penjualan ………………………… 30

Gambar 4.3 *Activity diagram* mengelola data produk ………………………. 31

Gambar 4.4 *Activity diagram* mengelola laporan …………………………… 32

Gambar 4.5 *Activity diagram* cetak laporan ………………………………… 33

Gambar 4.6 Activity diagram analisa pendapatan ………………………….. 33

Gambar 4.7 *Activity diagram* mengelola pengguna ………………………… 34

Gambar 4.8 *Activity diagram* *edit password* ………………………………... 35

Gambar 4.9 *sequence diagram* penjualan ………………………………….. 36

Gambar 4.10 *sequence diagram* produk …………………………………….. 36

Gambar 4.11 *sequence diagram* laporan ……………………………………. 37

Gambar 4.12 *sequence diagram* analisa pendapatan ……………………….. 37

Gambar 4.13 *sequence diagram* pengguna …………………………………. 38

Gambar 4.14 *class diagram* ………………………………………………… 38

Gambar 4.15 Gambar Rancangan Masuk Aplikasi …………………………. 42

Gambar 4.16 Gambar Rancangan Dashbard ………………………………... 42

Gambar 4.17 Gambar Rancangan Data Kategori …………………………… 43

Gambar 4.18 Gambar Rancangan Tambah Data Kategori ………………….. 43

Gambar 4.19 Gambar Rancangan Detail Data Kategori ……………………. 44

Gambar 4.20 Gambar Rancangan Edit Data Kategori ……………………… 44

Gambar 4.21 Gambar Rancangan Data Produk …………………………….. 45

Gambar 4.22 Gambar Rancangan Tambah Data Produk …………………… 45

Gambar 4.23 Gambar Rancangan Detail Data Produk ……………………... 46

Gambar 4.24 Gambar Rancangan Edit Data Produk ……………………….. 46

Gambar 4.25 Gambar Rancangan Data *Inventory* ………………………….. 47

Gambar 4.26 Gambar Rancangan Update Data *Inventory* ………………….. 47

Gambar 4.27 Gambar Rancangan Data Laporan …………………………… 48

Gambar 4.28 Gambar Rancangan Analisa dan Laporan ……………………. 48

Gambar 4.29 Gambar Rancangan Cetak Data Laporan …………………….. 49

Gambar 4.30 Gambar Rancangan Data Pengguna ………………………….. 49

Gambar 4.31 Gambar Rancangan Tambah Data Pengguna ………………… 50

Gambar 4.32 Gambar Rancangan Detail Data Pengguna …………………... 50

Gambar 4.33 Gambar Rancangan Edit Data Pengguna …………………….. 51

# **DAFTAR TABEL**

Tabel 3.1 Notasi-Notasi *Use Case Diagram* ………………………………... 16

Tabel 3.2 Notasi-Notasi *Class Diagram* ……………………………………. 17

Tabel 3.3 Notasi-notasi *Activity Diagram* …………………………………... 19

Tabel 3.4 Notasi-notasi *Sequence Diagram* ………………………………… 21

Tabel 4.1 Keterangan aktor dalam perancangan aplikasi …………………...

*inventory* dan penjualan ………………………………………..... 27

Tabel 4.2 Sekenario *Use Case* Mengelola Penjualan ……………………….. 27

Tabel 4.3 Sekenario *Use Case* Mengelola Data Produk ……………………. 28

Tabel 4.4 Sekenario *Use Case* Mengelola Laporan ………………………… 28

Tabel 4.5 Sekenario *Use Case* Analisa Perkembangan ……………………... 30

Tabel 4.6 Sekenario *Use Case* Mengelola Pengguna ……………………….. 30

Tabel 4.7 Perancangan Tabel Produk ……………………………………… 39

Tabel 4.8 Perancangan Tabel Kategori ……………………………………. 39

Tabel 4.11 Perancangan Tabel Penjualan …………………………………… 40

Tabel 4.12 Perancangan Tabel Pembayaran ………………………………… 40

Tabel 4.13 Perancangan Tabel Laporan …………………………………….. 41

Tabel 4.14 Perancangan Tabel Pengguna …………………………………… 41

# **BAB I**

# **PENDAHULUAN**

1. **Latar Belakang**

Semakin berkembangnya teknologi informasi pada era globalisasi saat ini membawa dampak pada semakin ketatnya persaingan dalam dunia bisnis. Pada akhirnya banyak perusahaan berlomba-lomba untuk berinovasi dalam menerapkan strategi pemasaran yang lebih baik dalam pengelolaan data yang lebih efisien untuk membantu menganalisa perkemabangan perusahaan.

Salah satu cara untuk meningkatkan strategi perkembangan perusahan adalah dengan menggunakan fasilitas aplikasi untuk membantu menganalisa perkembangan pelaporan dan pengelolaan data perusahaan.

CV. Panen Rejo adalah toko retail yang bertujuan untuk memenuhi kebutuhan pertanian dan peternakan, serta berbagai obat – obatan dan hasil olahan, dimana semua laporan penjualan dan laporan stok barang yang dilakukan masih bersifat manual dicatat di buku-buku, serta yang masih membebankan kepala toko untuk melakukan analisa perkembangan penjualan barang dan pencatatan laporan penjualan,

Dengan permasalahan yang didapat, untuk memenuhi kebutuhan yaitu dengan menggunakan aplikasi pencatatan keuangan, berdasarkan uraian tersebut penulis mengambil pembahasan mengenai “**Implementasi Sistem Informasi Keuangan dengan Metode Sistem Perpetual dan Periodik pada CV Panen Rejo Semarang**” yang di harapkan menjadi solusi untuk dapat membantu CV Panen Rejo dalam pengelolaan data keuangan.

1. **Batasan Masalah**

Adapun Batasan masalah dari Perancangan Aplikasi Inventory dan Penjualan di CV Tunas Agri Farm ini meliputi :

1. Laporan ini hanya sebatas Perancangan Aplikasi Inventory dan Penjualan.
2. Perancangan aplikasi ini menggunakan model pengembangan *system waterfall*
3. Perancangan aplikasi ini meliputi penginputan data *inventory*, pengelolaan *inventory*, dan penjualan barang,
4. **Tujuan Kerja Praktek**

Tujuan dibuatnya laporan kerja praktek ini adalah untuk merancang Aplikasi yang mampu membantu mempermudah pekerjaan pengelolaan data penjualan dan *inventory* yang ada pada CV Tunas Agri Farm, Suwakul, Bandarjo, Ungaran Barat.

1. **Metodologi Pelaksanaan**
   1. Pengumpulan Data

Data adalah sumber atau bahan mentah yang sangat berharga bagi suatu proses yang akan menghasilkan informasi, oleh karena itu dalam pengumpulan sumber data perlu penanganan secara cermat, sehingga menghasilkan data yang akurat dan berkualitas. Beberapa metode dalam pengumpulan sumber data yang digunakan :

* + 1. Wawancara

Metode pengumpulan sumber data ini diperoleh secara langsung dengan melakukan wawancara langsung dengan Pemilik CV. Tunas Agri Farm dan kasir. Dengan mengajukan pertanyaan tentang proses pengelolaan data dan kendala yang terjadi.

* + 1. Observasi

Metode pengumpulan sumber data ini diperoleh dengan pengamatan secara langsung proses barang masuk dan keluar, dan proses penjualan agar didapat data yang sesuai kebutuhan.

* + 1. Studi pustaka

Studi pustaka dilakukan dengan cara mempelajai teori-teori *literature* dan buku-buku yang berhubungan dengan aplikasi *inventory* dan sistem informasi sebagai dasar dalam pelakasaan penelitian.

Dengan sumber data yang di peroleh meliputi data primer dan data sekunder sebagai berikut:

1. Data primer

Merupakan data uang didapat secara langsung dari narasumber yang bersangkutan dengan penelitian melalui wawancara dengan kepala toko dan dari pengamatan penulis.

1. Data sekunder

Merupakan data yang diperoleh secara tidak langsung dari berbagai *literature* yang berhubungan dengan perancangan aplikasi *inventory*, data ini berupa bukti, catatan dalam pembuatan laporan kerja praktek.

* 1. Analisa Data

Mengidentifikasi masalah, menganalisa kebutuhan informasi proses penjualan dan pemasukan barang, Dan memahami perancangan aplikasi sesuai data yang ada dan mengimplementasikan model yang diinginkan oleh pemakai, serta penentuan relasi tabel *database* yang mempermudah dalam proses pembuatan aplikasi.

* 1. Pengembangan Aplikasi

Adapun metodologi pengembangan sistem dimulai dengan perancangan pada permodelan UML (*United Modeling Language*) dan penerapan pengembangan menggunakan model *waterfall*.

1. **Sistematika Penulisan**

Penyusunan laporan untuk kerja praktek memaparkan pembahasan tentang perancangan yang dilakukan penulis, dalam penyusunan laporan dengan sistematika sebagai berikut:

BAB I : PENDAHULUAN

Memaparkan latar belakang masalah, perumusan masalah yang akan dihadapi, Batasan masalah yang diambil, tujuan kerja praktik, metode pengumpulan data yang digunakan, dan sistematia penulisan laporan perancangan.

BAB II : TINJAUAN UMUM PERUSAHAAN

Menguraikan penjelasan tentang sejarah perusahaan, struktur organisasi, pengertian, tugas dan fungsi masing – masing struktur di CV Tunas Agri Farm.

BAB III : LANDASAN TEORI

Memuat tentang landasan teori yang relevan terhadap perancangan sebagai landasan dalam pembuatan aplikasi dan penjelasan yang berhubungan dengan permasalahan yang dibahas.

BAB IV : ANALISA DAN PEMBAHASAN

Menguraikan tentang perancangan aplikasi inventori yang di dasarkan pada analisa perancangan masukan, perancangan proses, dan perancangan keluaran.

BAB V : PENUTUP

Memuat mengenai penguraian terhadap kesimpulan hasil perancangan aplikasi dan saran yang perlu diambil oleh perusahaan yang bersangkutan dengan tujuan melancarkan dan mempermudah dalam pengelolaan barang masuk dan keluar.

DAFTAR PUSTAKA

LAMPIRAN

# **BAB II**

# **TINJAUAN UMUM CV PANEN REJO**

1. **Sejarah**

CV Panen Rejo berdiri pada tanggal 1 Juli 2013, dengan bidang usaha toko ritail pertanian yang terbagi menjadi beberapa kelompok. kelompok saprotan, obat – obatan, dan pupuk. CV Panen Rejo hanya menjual produk - produk pertanian dan alat – alat penunjang pertanian.

1. **Visi dan Misi**
2. **Visi**

Menjadi toko ritail pertanian terlengkap yang tersebar diseluruh kota besar di indonesia

1. **Misi**
2. Menyediakan sarana pertanian, tanaman, media informasi, produksi olahan herbal dan jasa - jasa bidang pertanian kepada masyarakat umum
3. Medidik masyarakat perkotaan akan pentingnya menanam pohon di sekitar rumah
4. Memberikan informasi atau pengetahuan mengenai pertanian kepada masyarakat pada umumnya
5. Membangun jaringan agribisnis
6. **Tentang CV Panen Rejo**
7. **Alamat**

CV Panen Rejo beralamat di Jalan Tamrin.



Gambar 2.1 Peta Lokasi

1. **Kontak** **Informasi :**

Telp : 024-6922976, 0812-2504-8383

Email : admin@tunasagri.com

Website : tunasagri.com

Media Sosial : https://facebook.com/tunasagrifarm

https://instagram.com/tunas\_agri\_farm



Gambar 2.2 Gerbang Toko



Gambar 2.3 Halaman Toko



Gambar 2.4 Ruangan Toko

1. **Sarana dan Prasarana**
   1. Sarana

CV Panen Rejo adalah tempat pembelian sarana pertanian, sarana perkebunan, dan sarana obat tanaman.



Gambar 2.5 Sarana Tanaman Perkebunan

* 1. Prasarana

Prasarana penunjang pertubuhan dan pembuahan tanaman



Gambar 2.6 Prasarana Pertanian dan Perkebunan

1. **Struktur Organisasi**

C:\Users\Nandhief\AppData\Local\Microsoft\Windows\INetCache\Content.Word\TUNAS STRUKTUR ORGANISASI.PNG

Gambar 2.7 Struktur Oraganisasi

1. **Tugas dan Wewenang Masing – Masing Jabatan**
2. **Pemilik**

Pemilik sebagai wewenang yang bertugas dalam hal perancangan, pengelolaan dan pengambilan keputusan terhadap perusahaan dan membina tenaga kerja.

1. **Kasir dan Keuangan**

Kasir & Keuangan bertugas untuk mengatur dan mengelola keuangan perusahaan. Proses keluar dan masuknya uang perusahaan semua harus diketahui dan disetujui oleh pihak keuangan.

# **BAB III**

# **LANDASAN TEORI**

1. **Pengertian Sistem**
2. **Persediaan**

Dalam perusahaan dagang maupun manufaktur, persediaan sangat dan termasuk bagian aktivasi lancer yang aktif.

Suatu aktifa yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam periode usaha tertentu atau persediaan barang – barang yang masih dalam pengerjaan atau proses produksi ataupun persediaan bahan baku yang menunggu penggunaannya dalam proses produksi (Benny Alexandri 2009)

Sedangkan menurut Ristono (2009) persediaan merupakan teknik untuk manajemen material yang berkaitan dengan perusahaan.

1. **Penjualan**

Didalam sebuah perusahaan khususnya dalam perusahaan yang bergerak dibidang usaha perdagangan, penghasilan utama yang dihasilkannya ialah berasal dari penjualan.

1. Pengertian penjualan menurut beberapa ahli
   1. Menurut Mulyadi (2013:455) menyatakan bahwa “Penjualan adalah penjualan yang dilaksanakan oleh perusahaan dengan cara mewajibkan pembeli melakukan pembayaran harga barang terlebih dahulu sebelum barang diserahkan oleh perusahaan kepada pembeli”.
   2. Menurut Basu Swastha (2010: 8) menyatakan bahwa “Menjual adalah ilmu dan seni mempengaruhi pribadi yang dilakukan oleh penjual untuk mengajak orang lain agar bersedia membeli barang atau jasa yang ditawarkan”.
   3. Menurut Leny Sulistiyowati (2010:270) manyatakan bahwa "Penjualan adalah pendapatan yang berasal dari penjualan produk perusahaan, disajikan setelah dikurangi potongan penjualan dan retur penjualan".
   4. Berdasarkan pengertian diatas mengenai penjualan, maka disimpulkan bahwa penjualan adalah salah satu prosedur yang digunakan untuk memperjualbelikan barang dan jasanya kepada konsumen, yang tujuannya untuk memperoleh keuntungan atau laba.
2. Jenis – jenis penjualan
   1. Menutut Basu Swastha (2005:11) ada lima jenis penjualan, yaitu:
      1. *Trade Selling*, dapat terjadi bila produsen dan pedagang besar mempersilahkan pengecer untuk berusahah memperbaiki distribusi produk-produk mereka.
      2. *Missionary Selling*, adalah usaha untuk meningkatkan penjualan dengan memborong pembeli untuk memberi barang-barang dari penyalur perusahaan.
      3. *Technical Selling*, usaha untuk meningkatkan penjualan dengan memberikan saran dan nasehat kepada pembeli akhir dari barang dan jasa yang dijual.
      4. *New Business Selling*, usaha untuk membuka transaksi baru dengan mengubah calon pembeli. Jenis penjualan ini sering digunakan oleh perusahaan asuransi.
      5. *Responsive Selling*, setiap tenaga penjualan diharapkan dapat memberikan reaksi terhadap permintaan pembeli.
   2. Sedangkan menurut Azhar Susanto (2013:176) berdasarkan dari jenis transaksi membagi kedalam:
      1. Penjualan secara tunai, yaitu penjualan yang bersifat cash and carry, pada umumnya terjadi secara spontan. Pembayaran dalam jangka waktu satu bulan yang masih sebagai penjualan tunai.
      2. Penjualan secara kredit, penjualan yang periode pembayarannya rata-rata diatas satu bulan.
      3. Penjualan secara tender, yaitu penjualan yang dilaksanakan melaui proses tertentu untuk memenuhi permintaan pihak pembeli yang membuka tender tersebut, untuk memenangkan tender selain harus memenuhi berbagai prosedur yaitu pemenuhan dokumen tender yang berupa jaminan tender, jugaharus dapat bersaing dengan pihak lain.
      4. Penjualan *export*, yaitu penjualan yang dilakukan dengan pembelian luar negeri yang mengimport barang.
      5. Penjualan secara konsinyasi yaitu, menjual barang-barang secara titipan kepada pembeli atau penjual. Apabila barang tersebut tidak laku dijual, maka barang tersebut akan dikembalikan kepada si penjual.
      6. Penjual melaui grosir, yaitu penjualan yang tidak langsung kepada konsumen, tetapi melalui pedagang pertama (grosir). Grosir berfungsi sebagai perantara antara pabrik atau import dengan pelanggan atau toko eceran.
3. **Pengertian Pencatatan**

Pada suatu perusahaan tentunya diperlukan untuk adanya pencatatan persediaan, karena akan membantu kegiatan operasional perusahaan, pencatatan persediaan sangat membantu dalam mengontrol serta mengelola masuk keluarnya persediaan, dan pencatatan adalah proses analisis atas suatu transaksi atau peristiwa keuangan yang terjadi dalam entitas dengan cara menempatkan transaksi di sisi debet dan sisi kredit menurut Rahman Pura (2013:26).

1. **Sistem Pencatatan Periodik**

Menurut Imam Santoso (2010:241) sistem pencatatan periodi adalah suatu sistem pengelolaan persediaan dimana dalam penentuan persediaan dilakukan melakukan melalui perhitungan secara fisik (physical counting) yang lazim dilakukan pada setiap akhir periode akuntansi dalam rangka penyiapan laporan keuangan. Melaui perhitungan fisik ini, jumlah kuantitas porsediaan (inventory quantity) akan diketahui ( misalnya dalam berat, meter, kilogram dan sebagainya) sehingga nilai persediaan (inventory value) dapat dihitung dengan mengalikan jumlah kuantitas persediaan dengan suatu harga”.

Sedangkan menurut Dwi Martani (2012:250) sistem pencatatan periodik adalah sistem periodik merupakan sistem pencatatan persediaan dimana kuantitas persediaan ditentukan secara periodik yaitu hanya pada saat perhitungan fisik yang biasanya dilakukan secara stock opname.

1. **Sistem Pencatatan Perpetual**

Menurut Raja Adri Satriawan Surya (2012:114) Sistem persediaan periodik memiliki karakteristik sebagai berikut: a. Pembelian persediaan di debet ke dalam akun pembelian (purchases). b. Asuransi dan biaya pengangkutan masuk, retur dan pengurangan pembelian dicatat ke dalam akunnya masing-masing. c. Akun persediaan ditentukan secara periodic dengan menutup nilai persediaan awal dan persediaan akhir ke dalam ikhtisar laba-rugi. d. Biaya persdiaan dan harga pokok penjualan ditentukan secara periodik

1. ***United Modeling Language* (*UML*)**

Menurut Adi Nugroho (2010), UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan (*modeling*) sesungguhnya digunakan untuk menyederhanakan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipahami dan dipelajari. *UML* juga menyediakan standar notasi ataupun diagram yang digunakan untuk pemodelan sistem,

1. ***Use* *Case* *Diagram***

Menurut Adi Nugroho (2010), *Use Case Diagram* digunakan untuk memodelkan fungsionalitas - fungsionalitas sistem/perangkat lunak dilihat dari pengguna yang ada di luar sistem (yang sering dinamakan sebagai aktor). *Use Case* pada dasarnya merupakan unit fungsionalitas koheren yang diekspresikan sebagai transaksi-transaksi yang terjadi antara aktor dan sistem. Kegunaan dari *use case diagram* adalah untuk mendaftarkan aktor-aktor dan *use case – use case* dan memperlihatkan aktor-aktor mana yang berpartisipasi dalam masing-masing *use case*. Berikut notasi-notasi yang digunakan dalam *use case diagram*:

Tabel 3.1 Notasi-Notasi *Use Case Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NOTASI** | **KEGUNAAN** | | **SIMBOL** |
| *Actor* | Menggambarkan semua objek diluar sistem (bukan hanya pengguna sistem/ perangkat lunak) yang berinteraksi dengan sistem yang dikembangkan. | |  |
| *Use Case* | Menggambarkan fungsionalitas yang dimiliki sistem. | |  |
| *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). | |  |
| *Assosiation* | Lintasan komunikasi antara *actor* dengan *use case.* | |  |
| *Include* | Penambahan perilaku ke suatu *use case* dasar yang secara eksplisit mendeskripsikan penambahan tersebut. | | <<include>> |
| *Extend* | Penambahan perilaku ke suatu *use case* dasar. | | <<extend>> |
| *Generalization* | Relasi antara pengklasifikasi yang memiliki deskripsi yang bersifat labih umum dengan berbagai pengklasifikasi yang lebih spesifik, digunakan dalam struktur pewarisan. | |  |
| *Dependency* | | Relasi antar dua elemen model. |  |

1. ***Class* *Diagram***

Menurut Adi Nugroho (2010), dalam notasi *UML*, himpunan kelas-kelas beserta hubungan / relasi / asosiasi antar kelas biasanya digambarkan menggunakan sebuah *diagram UML* yang dinamakan diagram kelas (*class diagram*). Jika kita perhatikan lebih jauh, sesungguhnya diagram kelas memiliki dua kegunaan / fungsi yang sangat penting, yaitu:

1. Mempresentasikan keadaan statis kelas-kelas yang terlibat dalam sistem. Kelas-kelas ini bisa saja merupakan kelas-kelas dalam bahasa pemrograman dan kelas-kelas persisten yang hadir dalam bentuk tabel-tabel yang ada di sistem basis data relasional.
2. Hubungan antar kelas dalam sistem atau perangkat lunak yang sedang kita kembangkan dapat terlihat dengan mudah.

Berikut notasi-notasi yang digunakan dalam *class diagram UML*:

Tabel 3.2 Notasi-Notasi *Class Diagram*

|  |  |  |
| --- | --- | --- |
| **NOTASI** | **KETERANGAN** | **SIMBOL** |
| *Class* | *Class* adalah balok-balok pembangun pada pemrograman berorientasi objek. Sebuah *class* digambarkan sebagai sebuah kotak yang terbagi menjadi 3 bagian. Bagian atas adalah bagian nama dari *class*. Bagian tengah mendefinisikan atribut *class*. Bagian bawah mendefinisikan *method* dari sebuah *class*. |  |

Tabel 3.2 Notasi-Notasi *Class Diagram* (Lanjutan)

|  |  |  |
| --- | --- | --- |
| **NOTASI** | **KETERANGAN** | **SIMBOL** |
| *Assosiation* | Sebuah asosiasi merupakan sebuah *relationship* paling umum antara 2 *class*, dan dilambangkan oleh sebuah garis yang menghubungkan antar 2 *class*. Garis ini dapat melambangkan tipe - tipe *relationship* dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah *relationship*. | 1..n  owned  by 1 |
| *Composition* | Jika sebuah *class* tidak bisa berdiri sendiri dan harus menjadi bagian dari *class* yang lain, maka *class* tersebut memiliki relasi *composition* terhadap *class* tempatnya bergantung tersebut. |  |
| *Dependency* | Kadang kala class menggunakan *class* yang lain. Hal ini disebut *dependecy*. Umumnya *dependency* digunakan untuk menunjukkan operasi pada suatu *class* yang menggunakan *class* yang lain. |  |
| *Aggregation* | *Aggregation* mengindikasikan keseluruhan bagian *relationship* dan biasanya disebut relasi “bagian dari”. Sebuah *aggregation* digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi / tidak solid. |  |
| *Generalization* | Sebuah *generalization* dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke arah “*parent*”-nya / induknya. |  |

1. ***Activity* *Diagram***

Menurut Adi Nugroho (2010), diagram aktivitas (*activity diagram*) sesungguhnya merupakan bentuk khusus dari state machine yang bertujuan untuk memodelkan komputasi-komputasi dan aliran - aliran kerja yang terjadi dalam sistem / perangkat lunak yang sedang dikembangkan. State pada diagram aktivitas merepresentasikan state dari komputasi yang dieksekusi, bukan state dari suatu objek biasa.

Biasanya, suatu *diagram* aktivitas mengasumsikan komputasi - komputasi dilaksanakan tanpa adanya interupsi-interupsi eksternal berbasis *event* terjadi padanya.

Suatu *diagram* aktivitas memuat di dalamnya *activity state* dimana suatu *activity state* merepresentasikan eksekusi pernyataan dalam suatu prosedur atau kinerja suatu aktivitas dalam suatu aliran kerja. Alih-alih menunggu selesainya atau event seperti yang terjadi pada state tunggu, *activity state* menunggu selesainya komputasi. Saat suatu aktivitas selesai maka akan berlanjut ke *activity state* berikutnya yang terlihat pada diagram aktivitas. Penyelesaian transisi dalam suatu diagram aktivitas biasanya akan terpicu saat aktivitas sebelumnya selesai.

*Activity state* biasanya tidak memiliki transisi-transisi yang diakibatkan oleh *event-event* eksplisit, tapi mungkin dibatalkan oleh transisi-transisi pada state yang melingkupinya. Berikut adalah notasi - notasi / simbol-simbol yang digunakan pada *activity diagram* :

Tabel 3.3 Notasi-notasi *Activity Diagram*

|  |  |  |
| --- | --- | --- |
| **NOTASI** | **KETERANGAN** | **SIMBOL** |
| *Initial* | Titik awal untuk memulai suatu aktivitas. |  |

Tabel 3.3 Notasi-notasi *Activity Diagram* (Lanjutan)

|  |  |  |
| --- | --- | --- |
| **NOTASI** | **KETERANGAN** | **SIMBOL** |
| *Final* | Titik akhir untuk mengakhiri aktivitas. |  |
| *Activity* | Menandakan sebuah aktivitas. |  |
| *Decision* | Pilihan untuk menggambil keputusan. |  |
| *Fork* | Menunjukkan kegiatan yang dilakukan secara paralel. |  |
| *Join* | Untuk menggabungkan beberapa kegiatan secara paralel menjadi satu. |  |

1. ***Sequence* *Diagram***

Menurut Adi Nugroho (2010), *Sequence* *Diagram* memperlihatkan interaksi sebagai *diagram* dua matra (dimensi). Matra vertikal adalah sumbu waktu, waktu bertambah dari atas ke bawah. Matra horizontal memperlihatkan peran pengklasifikasian yang merepresentasikan objek - objek mandiri yang terlibat dalam kolaborasi. Masing - masing pengklasifikasian direpresentasikan sebagai kolom-kolom vertikal dalam *sequence diagram* yang sering disebut sebagai garis waktu (*life line*). Selama objek ada, peran digambarkan menggunakan garis tegas. Selama aktivitas prosedur pada objek aktif, garis waktu digambarkan sebagai garis ganda. Pesan-pesan digambarkan sebagai suatu tanda panah dari garis waktu suatu objek ke garis waktu objek lainnya.

Panah-panah menggambarkan aliran pesan antar peran pengklasifikasian digambarkan dalam urutan waktu kejadiannya dari atas ke bawah. Berikut selengkapnya notasi-notasi yang digunakan dalam *sequence* *diagram*:

Tabel 3.4 Notasi-notasi *Sequence Diagram*

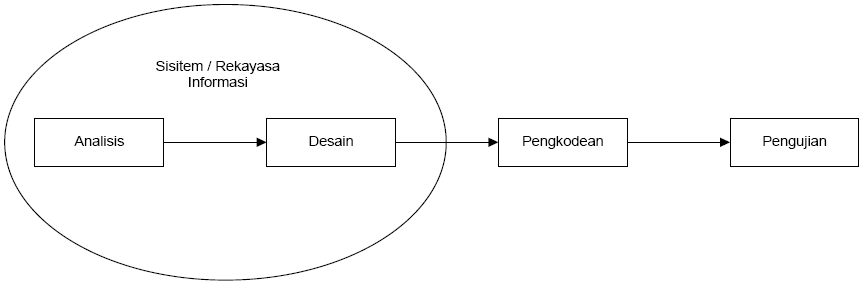
|  |  |  |
| --- | --- | --- |
| **NOTASI** | **KETERANGAN** | **SIMBOL** |
| *Object*  (Partisipan) | Objek atau biasa disebut partisipan merupakan *intance* dari sebuah *class* dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah *class* (kotak) dengan nama objek di dalamnya yang diawali dengan titik koma. |  |
| *Actor* | *Actor* juga dapat bekomunikasi dengan objek, maka *actor* juga dapat diurutkan sebagai kolom. |  |
| *Life Line* | *Life line* mengindikasikan keberadaan sebuah *object* dalam baris waktu. Notasi untuk *life line* adalah garis putus-putus vertikal yang ditarik dari sebuah *object*. |  |

Tabel 3.4 Notasi-notasi *Sequence Diagram* (Lanjutan)

|  |  |  |
| --- | --- | --- |
| **NOTASI** | **KETERANGAN** | **SIMBOL** |
| *Activation* | Activation dinotasikan sebagai sebuah kotak persegi empat yang digambarkan pada sebuah *life line*. *Activation* mengindikasikan sebuah *object* yang akan melakukan sebuah aktivasi. |  |
| *Boundary* | *Boundary* terletak antara sebuah sistem dengan sekelilingnya. Semua *form*, laporan-laporan, antar muka diperangkat keras seperti *printer* dan *scanner* dan antar muka ke sistem lainnya adalah termasuk ke dalam kategori. |  |
| *Control* | *Control* berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi, atau penanganan kesalahan. |  |
| *Entity* | *Entity* digunakan untuk menangani informasi yang mungkin akan disimpan secara permanen. *Entity* bisa juga merupakan sebuah tabel pada struktur basis data. |  |
| *Message* | *Message*, digambarkan dengan anak panah horizontal antar *activation*. *Message* mengindikasikan komunikasi antara objek-objek. |  |
| *Self Message* | *Self message* mengindikasikan komunikasi kembali ke dalam sebuah objek itu sendiri. |  |

1. ***Waterfall***

Menurut Rosa dab Shalahudin, (2013) Model Air Terjun (*Waterfall*) disebut juga model Sekuensial Linier (*Sequential Linear*) atau alur hidup klasik (*Clasic Life Cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung.



Gambar 3.1 Metode *Waterfall*

(Rosa A. S dan M. Shalahuddin, 2013)

1. Analisa Sistem

Analisa sistem bertujuan untuk menentukan ruang lingkup, unit organisasi kegiatan dan skala sumber daya yang diperlukan untuk mensepesifikasi kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh pengguna. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

1. Desain Sistem

Pada tahap desain sistem ini adalah proses multi langkah yang focus pada desain pembuatan program untuk menentukan proses, struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

1. Pengkodean program (*Coding*)

*Coding* merupakan unsur yang sangat penting dalam pembuatan aplikasi. Desain harus diterjemahkan kedalam program perangkat lunak untuk menjadikan fungsi-fungsi atau fitur yang sesuai dengan desain yang telah dibuat pada tahap desain.

1. Pengujian (*Testing system*)

Tahap pengujian sistem focus pada perangkat lunak secara segi *logic* dan fungsional, dan memastikan bahwa semua bagian sudah diuji, hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

1. Pemeliharaan (*Maintanance*)

*Maintenance*, pada tahap ini perangkat lunak sudah terbentuk dengan baik, namun tidak menutup kemungkinan untuk perbaikan, perubahan ketika sudah dikirim ke pengguna, Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari tahap analisis spesifikasi untuk perubahan perangkat lunak baru.

# **BAB IV**

# **ANALISA DAN PERANCANGAN**

1. **Analisa Sistem yang Berjalan**

Saat ini sistem pencatatan inventori dan penjualan barang masih menggunakan prosedur manual yang di lakukan oleh kasir maupun pemilik toko yaitu mencatat di buku dan menghitung kembali setiap ada pemasukan barang dan pengeluaran barang.

1. **Analisa Kebutuhan**

Dalam mengimplementasikan aplikasi ini di butuhkan empat buah komponen penting yang akan dianalisa kebutuhannya:

1. Perangkat yang disarankan untuk implementasi program. Spesifikasi perangkat keras yang dibutuhkan meliputi:
   1. Komputer dengan *Processor* minimal Pentium IV
   2. *Monitor* VGA
   3. RAM *minimal* 512 Mb
   4. Hardisk berkapasitas *minimal* 40 Gb
2. Perangkat lunak yang disarankan untuk implementasi program yaitu:
   1. Web Browser Google Chrome

Perangkat lunak ini di tujukan untuk memjalankan aplikasi pencatatan keuangan dengan berbasis web

* 1. XAMPP (Apache, MYSQL *Database*, PHP *Script* *Language*)

Perangkat lunak ini berfungsi sebagai server untuk mengelola pemprosesan dari aplikasi pencatatan keuangan.

1. Data dan Informasi

Dalam pembuatan aplikasi ini, daftar kebutuhan data dan informasi yang dikehendaki adalah:

* 1. Data Area
  2. Data Barang
  3. Laporan Pencatatan

Dalam menyusun sistem baru, software dan pemprograman yang di gunakan adalah berbasis web meliputi perpaduan antara PHP, HTML, dan MYSQL.

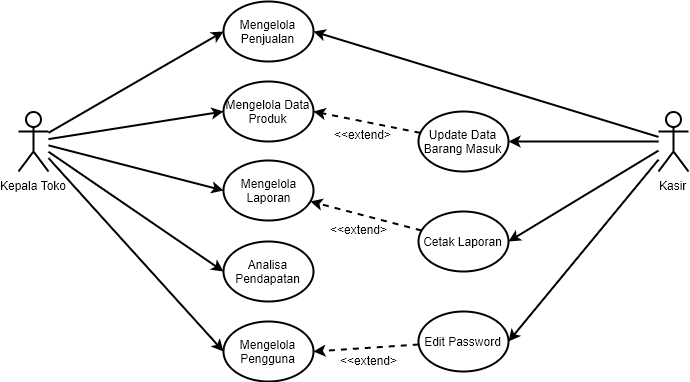
1. Pengguna (*Brainware*)

Brainware diperlukan Karena aplikasi ini tidak dapat beroperasi dengan sendirinya tanpa adanya intruksi dari pengguna. Dalam tahapan analisis ini pengguna dapat dikategorikan sebagai berikut:

* 1. Administrator yaitu seorang yang memiliki kewenangan dalam mengatur data. Mengatur disini antara lain mengelola data barang inventori, mulai dari penambah, mengedit, dan menghapus barang, serta menangani apabila terjadi error pada aplikasi.
  2. User yaitu seorang yang menggunakan website untuk melakukan proses penggajian, melakukan proses pengeluaran barang, cetak laporan penjualan.

1. **Analisa Desain**
2. *Unified Modeling Language* (UML)
   1. *Use Case Diagram*

*Use case diagram* merupakan diagram yang menampilkan aktor, *use case*, dan relasi dalam sebuah sistem, sebuah *use case* akan mempresentasikan sebuah interaksi antar pelaku dengan sistem.



Gambar 4.1 *Use Case Diagram*

Tabel 4.1 Keterangan aktor dalam perancangan aplikasi *inventory* dan penjualan

|  |  |  |
| --- | --- | --- |
| NO | Nama Aktor | Keterangan |
| 1 | Pemilik Toko atau Kasir | * + - 1. Mengelola penjualan, dan mengontrol data produk,       2. Mengelola laporan untuk di analisa perkembangan perusahaan |

1. Sekenario *Use Case* Mengelola Penjualan

Nama *Use Case* : Mengelola Penjualan

Aktor : Kepala Toko / Kasir

Tipe : *Primary*

Prekondisi : Memasukkan *username* dan *password*

Tujuan : Mengelola Data Penjualan

Tabel 4.2 Sekenario *Use Case* Mengelola Penjualan

|  |  |
| --- | --- |
| Aktor | Sistem Aplikasi |
| 1. Pemilik Toko / Kasir masuk ke halaman utama web | 1. Menampilkan Halaman Utama dengan daftar penjualan barang |
| 1. Pemilik Toko mengakses menu Penjualan untuk melakukan aksi tambah, dan hapus data penjualan | 1. Sistem melakukan perubahan data di-*database* dan menyimpannya |

1. Sekenario *Use Case* Mengelola Data Produk

Nama *Use Case* : Mengelola Data Produk

Aktor : Kepala Toko

Tipe : *Primary*

Prekondisi : Memasukkan *username* dan *password*

Tujuan : Mengelola Data Produk

Tabel 4.3 Sekenario *Use Case* Mengelola Data Produk

|  |  |
| --- | --- |
| Aktor | Sistem Aplikasi |
| 1. Pemilik Toko masuk ke halaman utama web | 1. Menampilkan Halaman Utama dengan daftar data produk |
| 1. Pemilik Toko mengakses menu data produk untuk melakukan aksi edit, dan hapus data produk | 1. Sistem melakukan perubahan data di-*database* dan menyimpannya |

1. Sekenario Use Case Mengelola Laporan

Nama Use Case : Mengelola Laporan

Aktor : Kepala Toko

Tipe : *Primary*

Prekondisi : Memasukkan *username* dan *password*

Tujuan : Mengelola Laporan

Tabel 4.4 Sekenario *Use Case* Mengelola Laporan

|  |  |
| --- | --- |
| Aktor | Sistem Aplikasi |
| 1. Pemilik Toko masuk ke halaman utama web | 1. Menampilkan Halaman Utama dengan daftar menu laporan |
| 1. Pemilik Toko mengakses menu laporan untuk melakukan aksi cetak, edit, dan hapus data laporan | 1. Sistem melakukan perubahan data di-*database* dan menyimpannya |

1. Sekenario *Use Case* Analisa Perkembangan

Nama Use Case : Melihat Analisa Perkembangan

Aktor : Kepala Toko

Tipe : *Primary*

Prekondisi : Memasukkan *username* dan *password*

Tujuan : Melihat Perkembangan Perusahaan

Tabel 4.5 Sekenario *Use Case* Analisa Perkembangan

|  |  |
| --- | --- |
| Aktor | Sistem Aplikasi |
| 1. Pemilik Toko masuk ke halaman utama web | 1. Menampilkan Halaman Utama dengan daftar Analisa Perkembangan |
| 1. Pemilik Toko mengakses menu untuk melihat perkembangan | 1. Sistem melakukan perhitungan perkembangan |

1. Sekenari *Use Case* Mengelola Pengguna

Nama Use Case : Mengelola Pengguna

Aktor : Kepala Toko

Tipe : *Primary*

Prekondisi : Memasukkan *username* dan *password*

Tujuan : Mengelola data pengguna aplikasi

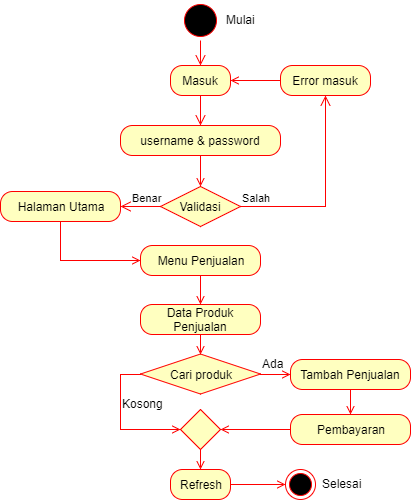
Tabel 4.6 Sekenario *Use Case* Mengelola Pengguna

|  |  |
| --- | --- |
| Aktor | Sistem Aplikasi |
| 1. Pemilik Toko masuk ke halaman utama web | 1. Menampilkan Halaman Utama dengan daftar Hak Akses |
| 1. Pemilik Toko mengakses menu untuk mengelola edit, tambah, hapus hak akses pengguna | 1. Sistem melakukan perubahan data di-*database* dan disimpan |

* 1. *Activity* *Diagram*

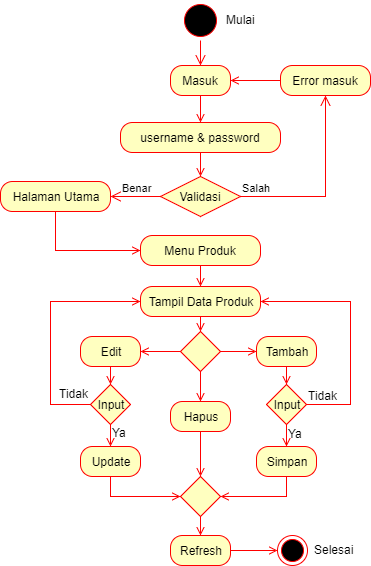
*Activity diagram* memberikan suatu gambaran ilustrasi alur dari setiap fungsi yang ada pada sistem.

1. *Activity diagram* mengelola penjualan



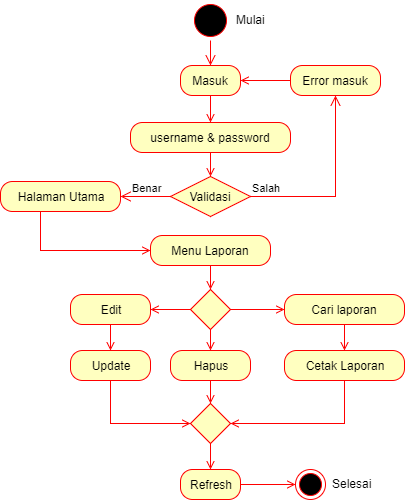
Gambar 4.2 *Activity diagram* mengelola penjualan

1. *Activity diagram* mengelola data produk



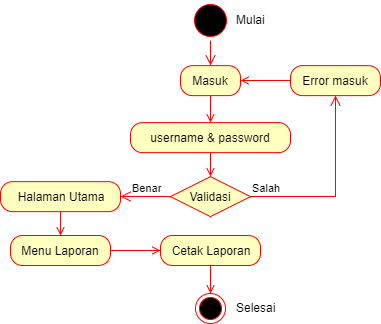
Gambar 4.3 *Activity diagram* mengelola data produk

1. *Activity diagram* mengelola laporan



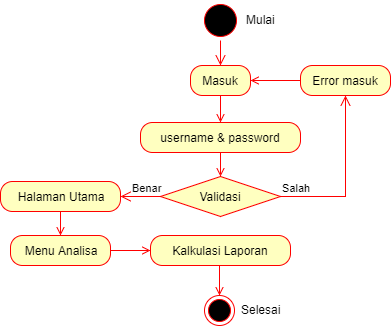
Gambar 4.4 *Activity diagram* mengelola laporan

1. *Activity diagram* cetak laporan



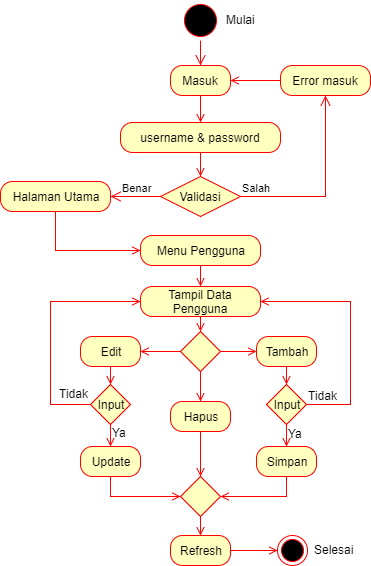
Gambar 4.5 *Activity diagram* cetak laporan

1. *Activity diagram* analisa pendapatan



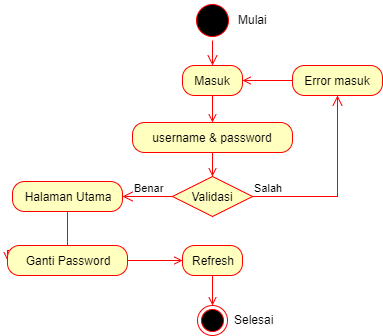
Gambar 4.6 *Activity diagram* analisa pendapatan

1. *Activity diagram* mengelola pengguna



Gambar 4.7 *Activity diagram* mengelola pengguna

1. *Activity diagram* *edit password*

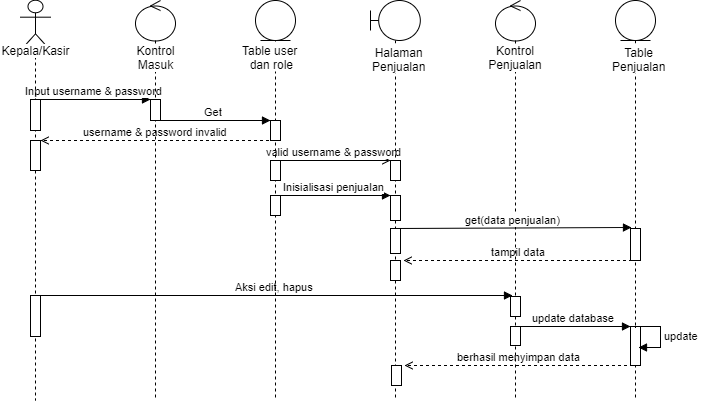


Gambar 4.8 *Activity diagram* *edit password*

* 1. *Sequence Diagram*

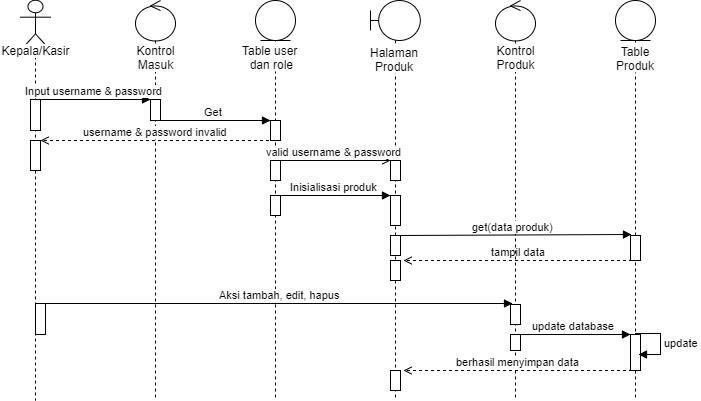
*Sequence diagram* menjelaskan interaksi antar objek-objek yang disusun dalam urutan waktu. Dalam sistem ini terdapat *sequence diagram* yang merupakan penjabaran dari masing-masing *usecase* yang ada pada *usecase diagram*.

1. *sequence diagram* penjualan



Gambar 4.9 *sequence diagram* penjualan

1. *sequence diagram* produk



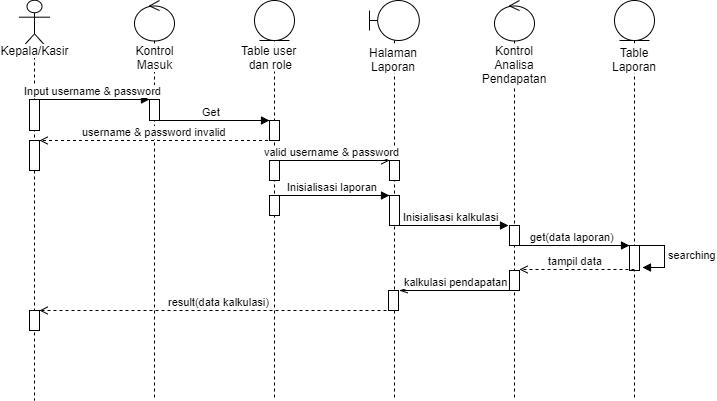
Gambar 4.10 *sequence diagram* produk

1. *sequence diagram* laporan



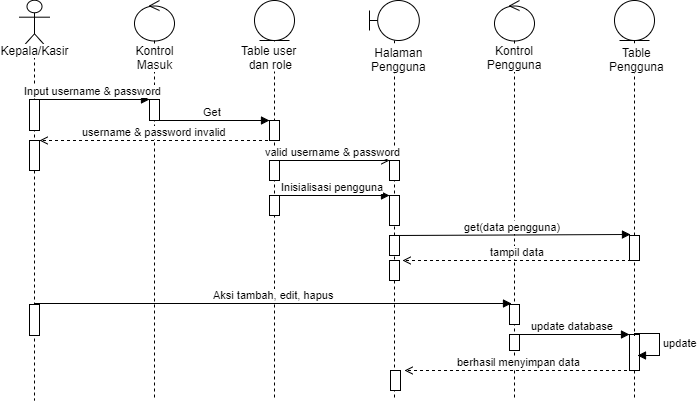
Gambar 4.11 *sequence diagram* laporan

1. *sequence diagram* analisa pendapatan



Gambar 4.12 *sequence diagram* analisa pendapatan

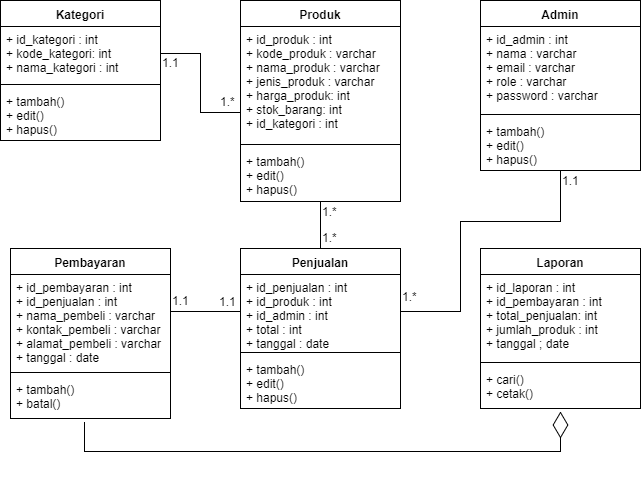
1. *sequence diagram* pengguna



Gambar 4.13 *sequence diagram* pengguna

* 1. *Class Diagram*

*Class Diagram* memperlihatkan hubungan antar kelas dan menggambarkan keadaan (atribut/properti) dari suatu sistem.



Gambar 4.14 *class diagram*

1. **Perancangan *Database***

Perancangan database dimaksud untuk membantu pembentukan database ketika mengimplikasikan system dalam bentuk aplikasi. Perancangan database ini juga menggambarkan tingkat relasi antar tabel satu dengan yang lainnya untuk membentuk table baru.

1. Rancangan Tabel Produk

Rancangan tabel produk akan memberikan standar pembentukan tabel untuk produk, baik itu *field* maupun atributnya.

Tabel 4.7 Perancangan Tabel Produk

|  |  |  |
| --- | --- | --- |
| **Field** | **Tipe Data** | **Keterangan** |
| id\_produk | Int | Primary key |
| kode\_produk | Varchar |  |
| nama\_produk | Varchar |  |
| jenis\_produk | Varchar |  |
| harga\_produk | Int |  |
| stok\_produk | Int |  |
| id\_kategory | Int | Foreign key |

1. Rancangan Tabel Kategori

Rancangan tabel kategori akan memberikan standar pembentukan tabel untuk kategori, baik itu *field* maupun atributnya.

Tabel 4.8 Perancangan Tabel Kategori

|  |  |  |
| --- | --- | --- |
| **Field** | **Tipe Data** | **Keterangan** |
| id\_kategori | Int | Primary key |
| kode\_kategori | Varchar |  |
| nama\_kategori | Varchar |  |

1. Rancangan Tabel Penjualan

Rancangan tabel kategori akan memberikan standar pembentukan tabel untuk kategori, baik itu *field* maupun atributnya.

Tabel 4.9 Perancangan Tabel Penjualan

|  |  |  |
| --- | --- | --- |
| **Field** | **Tipe Data** | **Keterangan** |
| id\_penjualan | Int | Primary key |
| id\_produk | Int | Foreign key |
| id\_admin | Int | Foreign key |
| total | Int |  |
| tanggal | Date |  |

1. Rancangan Tabel Pembayaran

Rancangan tabel penjualan akan memberikan standar pembentukan tabel untuk penjualan, baik itu *field* maupun atributnya.

Tabel 4.10 Perancangan Tabel Pembayaran

|  |  |  |
| --- | --- | --- |
| **Field** | **Tipe Data** | **Keterangan** |
| id\_pembayaran | Int | Primary key |
| id\_penjualan | Int | Foreign key |
| nama\_pembeli | Varchar |  |
| kontak\_pembeli | Varchar |  |
| alamat\_pembeli | Varchar |  |
| tanggal | Date |  |

1. Rancangan Tabel Laporan

Rancangan tabel laporan akan memberikan standar pembentukan tabel untuk laporan, baik itu *field* maupun atributnya.

Tabel 4.11 Perancangan Tabel Laporan

|  |  |  |
| --- | --- | --- |
| **Field** | **Tipe Data** | **Keterangan** |
| id\_laporan | Int | Primary key |
| id\_pembayaran | Int | Foreign key |
| total\_penjualan | Int |  |
| jumlah\_produk | Int |  |
| tanggal | Date |  |

1. Rancangan Tabel Pengguna

Rancangan tabel admin akan memberikan standar pembentukan tabel untuk admin, baik itu *field* maupun atributnya.

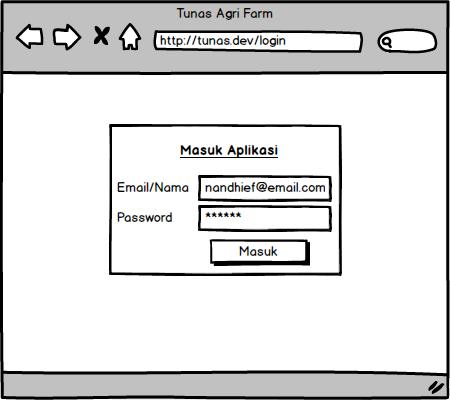
Tabel 4.12 Perancangan Tabel Pengguna

|  |  |  |
| --- | --- | --- |
| **Field** | **Tipe Data** | **Keterangan** |
| id\_admin | Int | Primary key |
| email\_admin | Varchar |  |
| nama\_admin | Varchar |  |
| role\_admin | Varchar |  |
| password\_admin | Varchar |  |

1. **Perancangan Antar Muka**

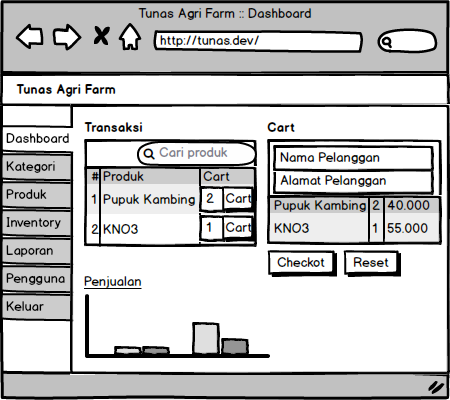
Berdasarkan perancangan aplikasi yang dibuat, maka akan di design perancangan aplikasi. Perancangan ini akan menjadi acuan bagi pihak yang akan mengembangkan dan menggunakan aplikasi.

1. Rancangan Halaman Login



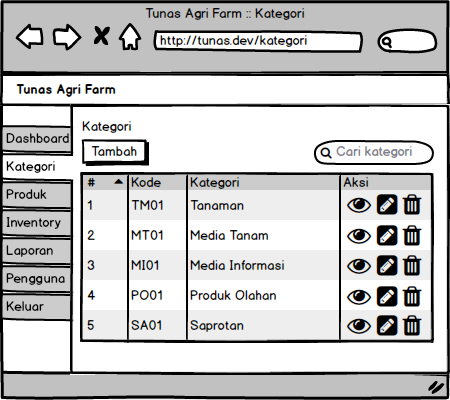
Gambar 4.15 Gambar Rancangan Masuk Aplikasi

1. Rancangan Halaman Dashboard

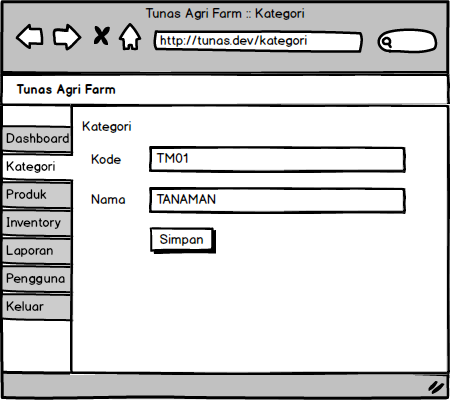


Gambar 4.16 Gambar Rancangan Dashbard

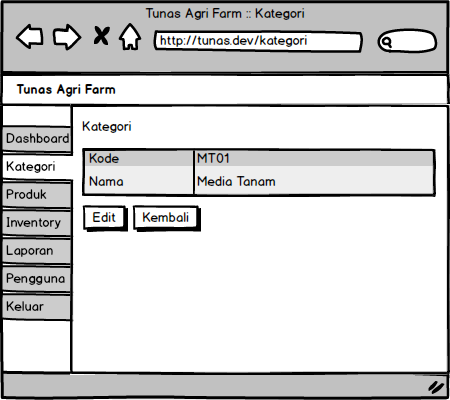
1. Rancangan Halaman Kategori



Gambar 4.17 Gambar Rancangan Data Kategori



Gambar 4.18 Gambar Rancangan Tambah Data Kategori

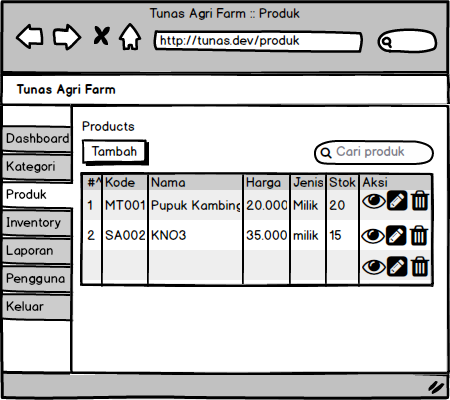


Gambar 4.19 Gambar Rancangan Detail Data Kategori

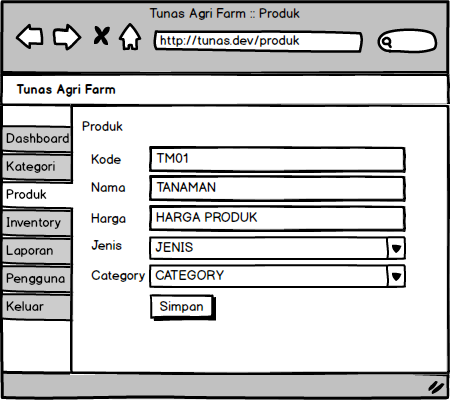


Gambar 4.20 Gambar Rancangan Edit Data Kategori

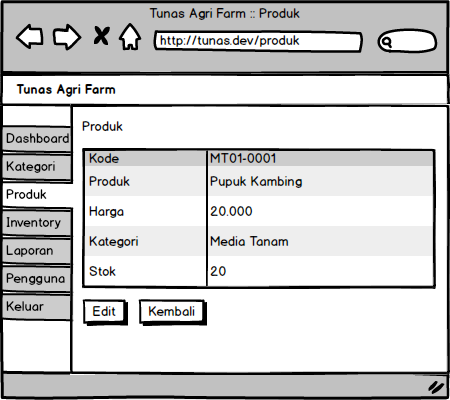
1. Rancangan Halaman Produk



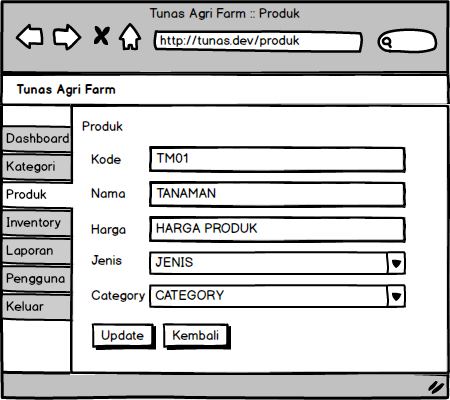
Gambar 4.21 Gambar Rancangan Data Produk



Gambar 4.22 Gambar Rancangan Tambah Data Produk

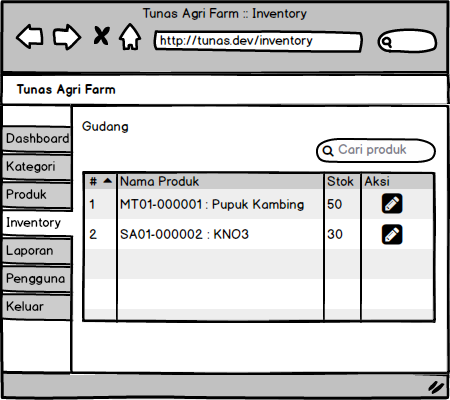


Gambar 4.23 Gambar Rancangan Detail Data Produk

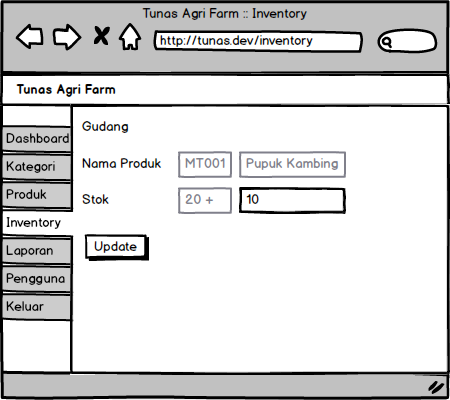


Gambar 4.24 Gambar Rancangan Edit Data Produk

1. Rancangan Halaman *Inventory*

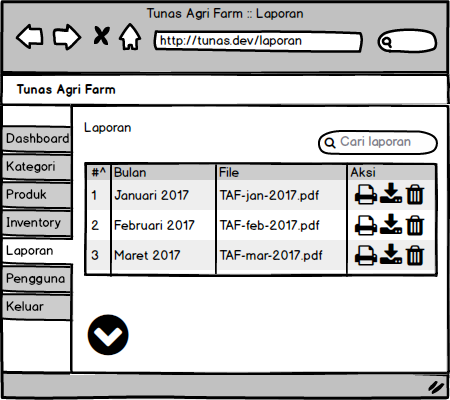


Gambar 4.25 Gambar Rancangan Data *Inventory*

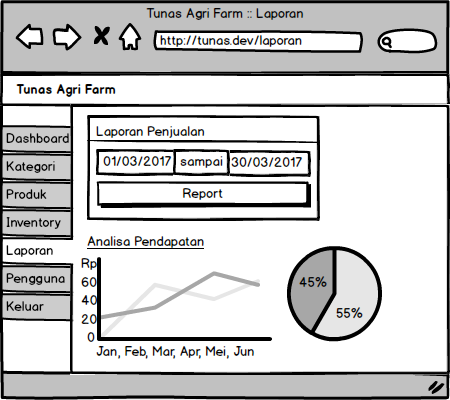


Gambar 4.26 Gambar Rancangan Update Data *Inventory*

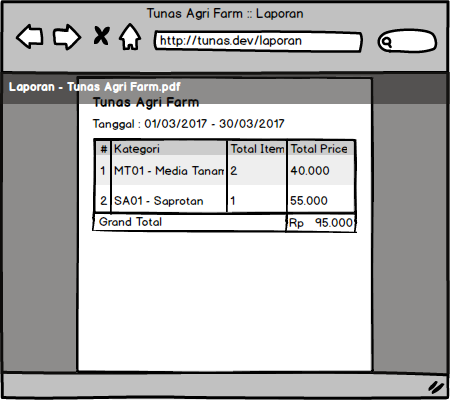
1. Rancangan Halaman Laporan



Gambar 4.27 Gambar Rancangan Data Laporan

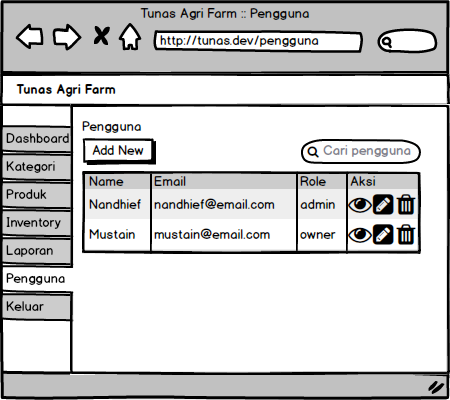


Gambar 4.28 Gambar Rancangan Analisa dan Laporan

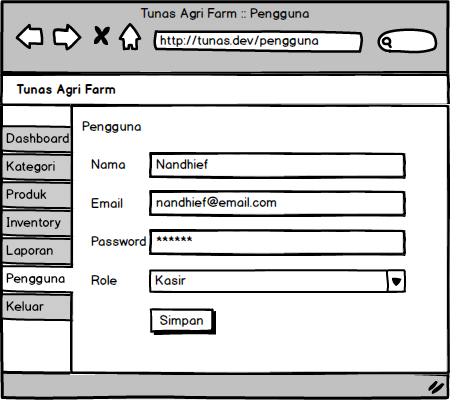


Gambar 4.29 Gambar Rancangan Cetak Data Laporan

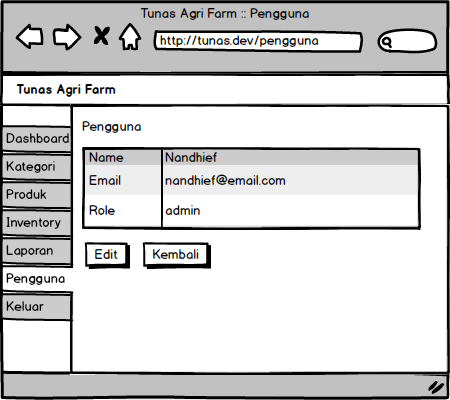
1. Rancangan Halaman Pengguna untuk Kepala Toko



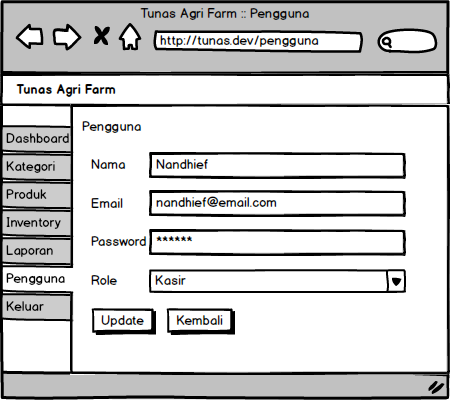
Gambar 4.30 Gambar Rancangan Data Pengguna



Gambar 4.31 Gambar Rancangan Tambah Data Pengguna



Gambar 4.32 Gambar Rancangan Detail Data Pengguna



Gambar 4.33 Gambar Rancangan Edit Data Pengguna

# **BAB V**

# **PENUTUP**

1. **Kesimpulan**

Berdasarkan hasil dari analisa permasalahan yang dibutuhkan CV Tunas Agri Farm dan merancang aplikasi *interface* pada bab sebelumnya, maka penulis mengambil kesimpulan sebagai berikut :

1. Perancangan aplikasi *inventory* dan penjualan menggambarkan struktural yang memperlihatkan proses otomisasi, sehingga kegiatan stok barang dan data penjualan tercatat efektif dan efisien,
2. Perancangan aplikasi ini juga di lengkapi dengan kalkulasi grafik dari penjualan yang mempermudah bagi CV Tunas Agri Farm untuk menganalisa perkembangan penjualan toko.
3. Perancangan aplikasi ini memiliki batasan hak akses untuk mengelola dengan masuk menggunakan email dan password yg sudah di daftarkan oleh kepala toko.
4. **Saran**

Berdasarkan kesimpulan diatas, saran penulis yang berhubungan dengan Aplikasi *Inventory* dan Penjualan adalah sebagai berikut :

1. Untuk inventory harus di kembangkan lagi dengan penambahan pengadaan barang dibagian *stock opname*,
2. Untuk mendukung aplikasi ini, diperlukan implementasi (*scripting*) atau pengembangan sistem terhadap perancangan aplikasi *inventory* dan penjualan dengan menggunakan pemprograman berbasis WEB.
3. Metode penjualan dalam pencarian produk dengan menggunakan QR-CODE *scanner*, untuk mempercepat saat transaksi,

# **DAFTAR PUSTAKA**

Adi Nugroho. 2010. Rekayasa Perangkat Lunak Berbasis Objek dengan Metode USDP. Andi. Yogyakarta.

Agus Ristono.2009. Manajemen Persediaan Edisi 1. Graham Ilmu: Yogyakarta.

Azhar Susanto. 2013. Sistem Informasi Akuntansi. Bandung: Lingga Jaya.

Jogiyanto, 2010. Analisis dan Desain Sistem Informasi, Edisi IV, Andi Offset, Yogyakarta.

Mulyadi. 2013.Sistem Akuntansi, Edisi Ketiga, Cetakan Keempat, Salemba Empat, Jakarta.

Simarmata. Janner (2010), Rekayasa Perangkat Lunak, Andi Offset, Yogyakarta.

Sulistiyowati, Leny. 2010. Panduan Praktis Memahami Analisis Laporan Keuangan, Jakarta: PT Elex Media Komputindo.

Swastha, Basu. 2010. Manajemen Penjualan: Pelaksanaan Penjualan, BPFE Yogyakarta.