**APLIKASI PENJUALAN BARANG BAHAN BANGUNAN (*POINT OF SALE*)BERBASIS WEB DI TOKO**

**BAHAN BANGUNAN SATRIA JAYA**

**KERJA PRAKTEK**

Diajukan Sebagai Salah Satu Syarat Kelulusan Mata Kuliah Kerja Praktek

Program Studi Teknik Informatika Fakultas Teknik Universitas Surya Kancana



**OLEH**

NAMA : HARIS ANWAR

NPM : 5520118042

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS SURYA KANCANA**

**CIANJUR**

**2021-2022**

# LEMBAR PERSETUJUAN

# LEMBAR KERJA PERSETUJUAN

Nama : HARIS ANWAR

NPM : 5520118042

Program Studi : Fakultas Teknik Informatika

Judul : APLIKASI PENJUALAN BARANG BAHAN BANGUNAN *(POINT OF SALE)* BERBASIS WEB DI TOKO BAHAN BANGUNAN SATRIA JAYA

Telah disetujui untuk disidangkan pada Sidang Kerja Praktek Pada Program Sarjana Strata-1 (S1), Program Studi Teknik Informatika Fakultas Teknik Universitas Suryakancana.

Cianjur, 27 Februari 2021

Pembimbing Kerja Praktek Pembimbing

Lapangan

Nama dan tanda tangan Hasbu Naim.S.SKom,MKom

NIDN.90290904394093

# LEMBAR PENGESAHAN

# LAPORAN KERJA PRAKTEK

Nama : HARIS ANWAR

NPM : 5520118042

Program Studi : Fakultas Teknik Informatika

Judul : APLIKASI PENJUALAN BARANG BAHAN BANGUNAN *(POINT OF SALE)* BERBASIS WEB DI TOKO BAHAN BANGUNAN SATRIA JAYA

Telah disidangkan dan dinyatakan Lulus Sidang Praktek pada Program Sarjana Strata-1 (S1), Program Studi Teknik Informatika Fakultas Teknik Universitas Suryakancana pada tanggal …..

Nama Penguji Tanda Tangan

1. Nama
2. Nama

Mengetahui:

Ketua Prodi Studi

Teknik Informatika

Tarmin Abdulghani ST, MT.

NIDN.

# LEMBAR PERNYATAAN KEASLIAN

# LAPORAN KERJA PRAKTEK

Nama : HARIS ANWAR

NPM : 5520118042

Program Studi : Fakultas Teknik Informatika

Judul : APLIKASI PENJUALAN BARANG BAHAN BANGUNAN *(POINT OF SALE)* BERBASIS WEB DI TOKO BAHAN BANGUNAN SATRIA JAYA

Dengan ini penulis menyatakan dalam laporan penelitian ini tidak terdapat karya yang pernah diajukan untuk memperoleh kelulusan **Kerja Praktek ataupun gelar kesarjanaan** di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh oranglain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka

Cianjur, 27 Februari 2021

Materai

Haris Anwar

NPM. 5520118042

# ABSTRAK

Pencatatan barang masuk, barang keluar, serta transaksi merupakan bagian yang dilakukan dalam operasional suatu toko, terutama toko yang bergerak di bidang perdagangan. Pencatatan barang masuk, barang keluar dilakukan oleh toko untuk memantau *stock* barang atau persediaan, serta transaksi yang terjadi di toko dicatat agar mempermudah proses bisnis dengan pelanggan atau konsumen. TOKO BARANG BAHAN BANGUNAN SATRIA JAYA merupa suatu toko yang bergerak di bidang perdagangan, pencatatan data barang masuk, barang keluar, serta transaksi dalam operasional toko.

Pada saat ini pencatatan barang masuk dari *supplier* ataupun barang keluar berupa *return* masih dicatat secara manual, begitu pun setiap transaksi penjualan dicatat secara manual, hal tersebut menimbulkan beberapa masalah diantaranya yaitu hilangnya data atau dokumen bukti penjualan ataupun data barang berupa kertas atau fisik, serta pada saat transaksi penjualan barang terjadi kesalahan penghitungan harga, klasifikasi atau kategori barang yang belum tersusun, membutuhkan waktu lama saat pencatatan barang,. Mengacu pada pentingnya proses bisnis yang berjalan, maka dibutuhkan suatu aplikasi penjualan barang (*point of sale*) yang dapat menangani permasalahan tersebut, sehingga proses bisnis dapat berjalan menjadi lebih efektif.

Dalam pembuatan perangkat lunak ini, digunakan metode pengembangan sistem berdasarkan paradigma *prototype* yang terdiri kominikasi, perencanaan, pemodelan. Bahasa pemrograman yang digunakan untuk membangun aplikasi penjualan barang (*point of sale*) yaitu menggunkan bahasa pemrograman *php, jquery, javascript,* dengan menggunakan framework *codeignaiter* dan *bootsrap* dan DBMS MYSQL dengan perangkat lunak XAMPP,. Untuk metoda perancangan sistem menggunakan konsep perancangan beriorentasi object dimana menggunakan *tools* perancangan yaitu UML.

Hasil dari perancangan pada penelitian ini, di implementasikan menjadi perangkat lunak ataupun aplikasi penjualan barang (*point of sale*) berbasis web, yang berfungsi untuk pengelolaan barang masuk, ataupun barang keluar, kategori barang, serta transaksi yang terkomputerisasi. Dengan pembuatan perangkat lunak ini diharapkan proses bisnis dapat berjalan dengan efektif.

Kata Kunci : aplikasi, barang masuk, barang keluar, transaksi, *php, MyQL, Prototype*

# KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT, karena hanya dengan limpahan rahmat dan ridha-Nya penulis dapat menyelesaikan laporan kerja praktek yang berjudul “APLIKASI PENJUALAN BARANG (*POINT OF SALE*) BAHAN BANGUNAN BERBASIS WEB DI TOKO BAHAN BANGUNAN SATRIA JAYA”.

Laporan ini merupakan salah satu persyaratan kelulusan pada matakuliah kerja praktek di Program Studi Teknik Informatika di Fakultas Teknik Universitas Suryakancana Cianjur.

Penelitian dan penyusunan laporan kerja praktek ini adalah suatu proses yang tidak lepas dari bantuan, bimbingan, serta dukungan berbagai pihak. Untuk itu penulis menguncapkan rasa terima kasih dan penghargaan yang sebesar-besanya kepada:

1. Bapak. Tarmin Abdulghani ST.MT selaku ketua jurusan fakultas Teknik Informatika
2. Bapak.Hasbu Naim S.S.Kom.,MKom selaku pembimbing kerja praktek
3. Ibu Siti Nazillah ST.M.Kom selaku koordinator kerja praktek
4. Bapak. H. Adhe S.E selaku pemilik Toko Bahan Bangunan Satria Jaya
5. Kedua Orangtua yang selalu memberi dukungan tanpa henti dan semangat kepada penulis

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh karena itu penulis mengharapkan masukan berupa saran dari semua pihak bagi perbaikan dan pengembangan penelitian ini. Akhir kata penulis berharap agar laporan kerja praktek ini dapat memberikan manfaat bagi perkembangan teknologi informasi saat ini.

Cianjur, 27 Februari

Penulis

# DAFTAR ISI

# BAB I

# PENDAHULUAN

# Latar Belakang

Kebutuhan akan suatu aplikasi merupakan hal yang sangat penting dan sangat berpengaruh dalam kehidupan. Peranan aplikasi sangat penting bagi kemajuan di segala bidang, terutama bidang perdagangan, teknologi diperuntukan bagi kemudahan manusia dalam mengerjakan segala kegiatan. Aplikasi juga merupakan salah satu aspek yang penting bagi toko, serta bagi semua aspek bidang, karena dapat membantu toko untuk meningkatakan produktifitas, baik dalam memperoleh, mengolah serta menggunakan aplikasi secara akurat.

Pada toko, pengelolaan data barang secara manual menimbulkan ketidak efektifan dalam pengecekan kembali barang, begitu pun transaksi manual yang seringkali menimbulkan ketidakpastian dalam jual beli, kepuasan konsumen selalu menjadi nomor satu bagi toko, terutama toko yang bergerak di bidang perdagangan. Perkembangan dan pemanfaatan teknologi, aplikasi saat ini sangat diperlukan terutama dalam pemanfaatan teknologi komputer, untuk pengolahan data juga menjadi perhatian di toko besar. Berbagai kegiatan toko bisa dikembangkan menjadi sistem yang kinerjanya menggunakan perangkat komputer atau kinerja di toko bisa ditingkatkan, seperti pengolahan data penjualan barang, pembelian barang dari vendor, serta pengelolaan data barang di gudang.

Mengenai hal tersebut tentunya pemanfaatan teknologi komputer sangat dibutuhkan di berbagai bidang, terutama bidang perdagangan dimana setiap barang masuk dan barang keluar, bisa terpantau secara *realtime*, selama ini bidang usaha, toko Bangunan Satria Jaya masih menggunakan tulis tangan, baik dalam pencatatan transaksi kepada konsumen ataupun pendataan barang bahan bangunan di gudang.

Untuk membantu dalam pengelolaan barang dan transaksi terhadap konsumen, maka penulis menemukan sebuah celah untuk membuat APLIKASI PENJUALAN BARANG (*POINT OF SALE*) BAHAN BANGUNAN BERBASIS WEB DI TOKO BAHAN BANGUNAN SATRIA JAYA, dengan adanya aplikasi penjualan barang (*point of sale*), pengelolaan akan terkoordinasi dengan baik, dan mengefensiensikan waktu, serta kemudahan dalam pengecekan barang dan transaksi terhadap konsumen,. Selain itu aplikasi ini dibuat sebagai sarana penelitian dan pengabdian penulis sebagai mahasiswa.

# Rumusan Masalah

Berdasarkan latar belakang yang penulis sampaikan diatas, maka dapat disimpulkan yaitu:

1. Bagaimana membuat sistem yang dapat mengelola barang masuk dan barang keluar ?
2. Bagaimana membuat sistem perhitungan penjualan barang atau transaksi penjualan?
3. Bagaimana membuat laporan transaksi penjualan dari tulis tangan, ke *digital* atau komputerisasi?

Dengan demikian dapat disimpulkan bahwa **TOKO BAHAN BANGUNAN SATRIA JAYA,** membutuhkan aplikasi yang dapat membantu dalam pengelolaan barang bahan bangunan serta transaksi yang terjadi,. Disini peranan mahasiswa sangat diperlukan, mengingat mahasiswa sebagai kaum pembaharuan, serta sebagai tolok ukur tridarma perguruan tinggi.

# Maksud Dan Tujuan

1. **Maksud**

Pembuatan aplikasi penjualan barang bahan bangunan (*point of sale*), ini dimaksudkan untuk membantu mengelola data barang, serta transaksi di TOKO BANGUNAN SATRIA JAYA yang saat ini masih menggunakan metode tulis tangan.

Aplikasi penjualan barang bahan bangunan (*point of sale*), ini diperlukan untuk mengelola data barang masuk, barang keluar, *stock* barang, serta transaksi dengan harapan mempermudah operasional kerja.

1. **Tujuan**

Tujuan dari pembuatan aplikasi penjualan (*point of sale*) barang bahan bangunan ini, adalah sebagai berikut :

1. Membuat sistem yang dapat mengelola barang masuk dan barang keluar, agar stock barang yang tersedia akan lebih mudah dipantau, dan di kendalikan.
2. Membuat sistem perhitungan penjualan barang atau transaksi penjualan, agar mempermudah proses transaksi dengan konsumen
3. Membuat laporan transaksi penjualan dari tulis tangan, ke *digital* atau komputerisasi

# Batasan Masalah

Agar pembuatan aplikasi penjualan barang bahan bangunan (*point of sale*) ini terarah dengan baik, maka dalam penulisan ini hanya akan membahas :

1. Pengelolaan data barang masuk
2. Pengelolaan data barang keluar
3. Transaksi penjualan
4. Laporan transaksi penjualan

# Batasan Masalah

* 1. Metode pengumpulan data yang sebagai berikut:

1. Wawancara, yaitu suatu kegiatan tanya jawab yang di lakukan oleh seseorang atau lebih, antara pembuat sistem dengan pihak toko. Ini dilakukan agar mendapatkan data yang akurat dan benar serta menghindari kesalahan informasi.
2. Observasi, pengumpulan data dengan mengamati langsung, melihat dan mengambil data yang dibutuhkan oleh pembuat aplikasi.
3. Tinjauan Pustaka, yaitu mencari suatu data yang bersumber dari para ahli, dengan mempelajari buku-buku serta *up to date* terhadap aplikasi.
   1. Metode Rekayasa Perangkat Lunak

Metode pada aplikasi penjualan barang bahan bangunan (point of sale) ini adalah metode prototype, karena merupakan model sederhana cocok digunakan untuk pengembangan perangkat lunak, dalam skala project kecil ke sedang, dengan melakukan pendekatan dengan client, mulai dari komunikasi hingga prototype dari aplikasi yang akan dibuat, setiap perubahan di deskripsikan

ke client, kemudian client akan merevisi jika rancangan aplikasi tidak sesuai, jika sesuai maka akan dilakukan ke tahap berikutnya.

Metode Prototype adalah metode pengembangan perangkat lunak yang memungkinkan adanya interaksi antara pengembang sistem dengan pengguna sistem, sehingga dapat mengatasi ketidakserasian antara pengembang dan pengguna (Pressman, 2012: 50). Adapun model pengembangan Prototype digambarkan pada gambar 1.1.

|  |
| --- |
|  |
| Gambar 1.1 *Prototype* Pressman (Pressman, 2015:42) |

Umumnya pelanggan mendefinisikan sejumlah sasaran perangkat lunak secara umum, tetapi tiak bisa mengidentifikasi spesifikasi kebutuhan yang rinci untuk fungsi-fungsi dan fitur-fitur yang nantinya akan dimiliki perangkat lunak yang akan dikembangkan. Dalam suatu kasus dan dalam situasi yang lain, paradigma pembuatan prototype (prototyping) mungkin menawarkan pendekatan yang paling baik (Pressman, 2012:50), dimulai dengan dilakukannya komunikasi antara tim pengembang perangkat lunak dengan pada pelanggan.

Tim perangkat lunak akan melakukan petemuan pertemuan dengan stakeholder untuk mendefinisikan sasaran keseluruhan untuk perangkat lunak yang akan dikembangkan, mengidentifikasi spesifikasi kebutuhan apa pun yang saat ini diketahui, dan menggambarkan area-area dimana didefinisi lebih jauh pada iterasi selanjutnyamerupakan keharusan.

Berikut tahap-tahap pada metode pengembangan sistem yang digunakan pada penelitian ini:

1. Komunikasi, merupakan tahapan awal dari model prototype guna mengidentifikasi permasalahan-permasalahan yang ada, serta informasi-informasi lain untuk di analisis dan diperlukan untuk membangun sistem.

2. Perencanaan merupakan tahapan yang dikerjakan dengan kegiatan penentuan Sumberdaya ataupun tenaga ahli yang dibutuhkan dalam membangun sistem, spesifikasi untuk pengembangan berdasarkan kebutuhan sistem, dan tujuan berdasarkan pada hasil komunikasi yang dilakukan agar pengembangan dapat sesuai dengan yang diharapkan.

3. Pemodelan merupakan, tahapan selanjutnya dimana representasi atau menggambarkan model sistem yang akan dikembangkan seperti proses dengan perancangan menggunakan Unified Modeling Language (UML) . Dalam tahap ini, Prototype yang dibangun dengan sistem rancangan sementara kemudian di evaluasi terhadap customer apakah sudah sesuai dengan yang diinginkan atau masih perlu untuk di evaluasi kembali. Setelah sistem dianggap sesuai dengan apa yang diharapkan customer, langkah berikutnya yaitu pembuatan aplikasi (pengkodingan) dari rancangan sistem yang dibuat diterjemahkan ke dalam bahasa pemrograman hingga menjadi sebuah sistem atau aplikasi.

4. Konstruksi merupakan tahapan yang digunakan untuk membangun prototype dan menguji-coba sistem yang dibangun. Proses instalasi dan penyediaan usersupport juga dilakukan agar sistem dapat berjalan dengan sesuai.

5. Penyerahan merupakan tahapan yang dibutuhkan untuk mendapatkan *feedback* dari pengguna, sebagai hasil evaluasi dari tahapan sebelumnya dan implementasi dari sistem yang dikembangkan.

# Tempat dan Waktu

Tempat Kerja Praktek dilaksanakan di :

Nama Instansi : TOKO BAHAN BANGUNAN SATRIA JAYA

Alamat : Jl.Aria Wiratanudatar KM.5 Sindanglaka, WA:0878 2030 2023,

Kabupaten Cianjur, Jawa Barat.

Waktu : Februari 2021 – Mei 2021

### Tabel 1.1 Jadwal Kegiatan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Uraian Langkah Kegiatan | Bulan | | | |
| Februari | Maret | April | Mei |
|
| 1 | Rumusan Masalah |  |  |  |  |
|
| 2 | Pengumpulan Data |  |  |  |  |
|
| 3 | Analisis |  |  |  |  |
|
| 4 | Desain |  |  |  |  |
|
| 5 | Implementasi |  |  |  |  |
|
| 6 | Pengujian |  |  |  |  |
|
| 7 | Penyusunan Laporan |  |  |  |  |
|
| 8 | Seminar |  |  |  |  |
|

# Sistematika Penulisan

Sistematika penulisan laporan kerja praktek ini dibagi menjadi 5 (lima) bab. Berikut penjelasan mengenai masing-masing bab :

**BAB I PENDAHULUAN**

Bab ini menjelaskan mengenai latar belakang masalah, rumusan masalah, maksud dan tujuan penelitian, batasan masalah yang berguna untuk memfokuskan pada pengelolaan data barang masuk, barang keluar dan transaksi sehingga dapat mencapai hasil yang optimal.

**BAB II TINJAUAN PUSTAKA**

Bab ini menjelaskan tentang teori-teori yang mendukung laporan kerja praktek. Pembahasan landasan teori ini meliputi penjelasan tentang definisi-definisi, penjelasan mengenai perangkat lunak yang digunakan dalam perancangan sistem informasi dan istilah istilah yang terkait didalamnya.

**BAB III ANALISIS DAN PERANCANGAN**

Bab ini menjelaskan uraian tentang analisa sistem yang akan dibuat dan bagaimana merancangnya sehingga menjadi sebuah sistem. Selain itu mencari informasi-informasi yang berkaitan dengan sistem yang akan dibangun.

**BAB IV IMPLEMENTASI & PENGUJIAN**

Bab ini menjelaskan mengenai pembuatan sistem, yang merupakan pengimplementasian dari hasil analisa dan perancangan yang berisi gambaran struktur menu serta tampilan layar dan langkah pengujian dari sistem yang akan dibangun.

**BAB V PENUTUP**

Bab ini berisi kesimpulan tentang seluruh pembahasan dan pemecahan masalah yang telah dilakukan serta mengenai hal-hal yang terdapat dalam bab-bab sebelumnya dan saran-saran untuk mengembangkan aplikasi berikutnya.

# BAB II

# TINJAUAN PUSTAKA

# Tinjauan Pustaka

Tinjauan pustaka ini membahas tentang gambaran toko, tempat survey yang didalamnya mencakup sejarah singkat toko, struktur organisasi toko dan deskripsi kerja.

## Sejarah Singkat Toko Satria Jaya

Pada tahun 1960 berdiri toko bangunan, yang bahan bernama toko Bahan Bangunan Bapak Fakih, Bapak Fakih adalah kakek dari Bapak Haji Deden (Pemilik Toko Satria Jaya), Bapak Fakih mempunyai anak dua orang, pada tahun 1972 Bapak Fakih mewariskan toko bangunan tersebut kepada anak nya yaitu Bapak Haji Cece dan toko bangunan itu bernama Toko Bangunan Susan Jaya yang beralamat di Jalan Aria Wiratanudatar Cimenteng Cianjur.

Alhamdulillah toko bahan bangunan tersebut sekarang sudah mempunyai 4 cabang yang diMengelola oleh anak-anak nya, Toko Bahan Bangunan Satria Jaya adalah cabang ke 3 (tiga) dari Toko Susan Jaya. Toko Bahan Bangunan Satria Jaya Cabang ke-3 (tiga) berdiri pada tahun 2007 yang beralamat di Jalan Aria Wiratanudatar Sindanglaka Cianjur, yang menjual bahan bangunan seperti kayu, bata, besi, semen, cat, paku, pasir, kramik, dan lain-lain.

Alhamdulillah Toko Satria Jaya sudah memiliki banyak pelanggan, mulai dari pelanggan biasa, sampai pelanggan menengah ke atas, harga jual bahan bangunan di Toko Satria Jaya bersaing dengan toko lain. Pemelik Toko Bahan Bangunan Satria Jaya berharap, semoga Toko Satria Jaya bisa berkembang dan tentunya bisa diwariskan ke generasi selanjutnya

## Struktur Organisasi Satria Jaya

|  |
| --- |
| **Pemilik**  Bapak H.Deden S.E  **Kepala Bagian**  Ibu Hj. Devi ST  **Kasir**  Ibu Evi  **Bagian Kepegawaian**  Bapak Asep  **Bagian Umum**  Bapak Acep |
| Gambar 2.1 Struktur Organisasi Toko Bahan Bangunan Satri Jaya |

## *Job* Deskripsi

1. Pemilik
2. Mengotrol semua aktifitas penjualan
3. Mengotrol semua aktifitas *stock* barang
4. Melakukan pengecekan laporan penjualan
5. Melakukan pengecekan laporan *stock* barang
6. Kepala Bagian
7. Memastikan pelaksanaan sebuah program
8. Menangani kinerja karyawan
9. Menjadi pusat koordinasi
10. Bagian Kepegawaian
11. Menjalankan prosedur penjadwalan pegawai
12. Mengelola penggajian pegawai
13. Menjalankan prosedur budaya kerja
14. Kasir
15. Menjalankan proses penjualan
16. Melakukan proses pengelolaan *stock* barang
17. Melakukan pencatatan atas semua transaksi
18. Membantu pelanggan dalam memberikan informasi mengenai suatu produk
19. Bagian Umum
20. Mengelola sarana dan prasaran toko
21. Menerima keluhan dari pelanggan atau *customer*
22. Bertanggung jawab terhadap alat yang digunakan di toko

## Landasan Teori

## Definisi Aplikasi

Definisi Aplikasi menurut beberapa ahli adalah sebagai berikut

1. Arti dari Aplikasi Menurut Supriyanto (2005:2) adalah program yang memiliki aktifitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu.
2. Dalam Kamus Besar Bahasa Indonesia Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu.
3. Menurut *Wikipedia*, aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.

Dari definisi diatas penulis dapat menyimpulkan bahwa aplikasi merupakan penerapan dari rancangan sistem untuk mengolah data atau mengolah dokumen yang dilakukan oleh pengguna sistem, yang di dalamnya terdapat intruksi atau pernyataan sehingga sehingga komputer bisa memproses input menjadi output yang di inginkan.

## Definisi Penjualan

Menurut Kamus Besar Bahasa Indonesa – penjualan adalah pengeluaran yang langsung berhubungan dengan usaha pemasaran produk;. Menurut Basu Swastha (2001:8) penjualan merupakan ilmu dan seni mempengaruhi pribadi yang dilakukan oleh penjual untuk mengajak orang lain agar bersedia membeli barang yang ditawarkan. Adapun faktor-faktor yang mempengaruhi penjulalan adalah kondisi dan kemampuan penjual, kondisi pasar, modal, dan kondisi organisasi toko.

Dari definisi diatas penulis dapat menyimpulkan bahwa penjualan merupakan suatu kegiatan menjual *product* barang atau jasa dengan tujuan utama yaitu mendatangkan keuntungan.

## Definisi Barang

Menurut *Wikipedia* Barang adalah suatu [objek](https://id.wikipedia.org/wiki/Objek) fisik yang dapat dilihat dan disimpan atau [jasa](https://id.wikipedia.org/wiki/Jasa) yang memiliki [nilai](https://id.wikipedia.org/wiki/Nilai). Nilai suatu barang akan ditentukan karena barang itu mempunyai kemampuan untuk dapat memenuhi [kebutuhan](https://id.wikipedia.org/wiki/Kebutuhan) atau keinginan pelanggan baik secara individu atau bisnis. [[1]](https://id.wikipedia.org/wiki/Barang#cite_note-1) Contoh barang adalah pakaian, makanan, minuman, komputer, dan telepon pintar.

Dari definisi diatas penulis dapat menyimpulkan barang merupakan *object* yang dapat dilihat dengan kasat mata dan memiliki suatu nilai ataupun bobot.

## Definisi Bahan

Menutut Kamus Besar Bahasa Indonesia bahan merupakan barang yang akan dibuat menjadi satu benda tertentu; bakal; 2 (segala) sesuatu yang dapat dipakai atau diperlukan untuk tujuan tertentu, seperti untuk pedoman atau pegangan, untuk mengajar, memberi ceramah; 3 sesuatu yang menjadi sebab (pangkal) atau sikap (perbuatan): -- tertawaan; -- pertikaian (perselisihan); 4 barang yang akan dipakai untuk bukti (keterangan, alasan, dan sebagainya): ia sedang mengumpulkan -- untuk menyusun tesisnya;

Dari definisi diatas penulis dapat menyimpulkan bahan merupakan material atau barang yang akan dibuat oleh seseorang atau sekelompok orang, melalui proses-proses tertentu hingga menjadi suatu karya yang dapat digunakan.

## Definisi Bangunan

Menrut *Wikipedia* Bangunan adalah struktur buatan manusia yang terdiri atas [dinding](https://id.wikipedia.org/wiki/Dinding) dan [atap](https://id.wikipedia.org/wiki/Atap) yang didirikan secara permanen di suatu tempat. Bangunan juga biasa disebut dengan [rumah](https://id.wikipedia.org/wiki/Rumah) dan [gedung](https://id.wikipedia.org/wiki/Gedung), yaitu segala sarana, prasarana atau [infrastruktur](https://id.wikipedia.org/wiki/Infrastruktur) dalam kebudayaan atau kehidupan manusia dalam membangun peradabannya. Bangunan memiliki beragam bentuk, ukuran, dan fungsi, serta telah mengalami penyesuaian sepanjang sejarah yang disebabkan oleh beberapa faktor, seperti bahan bangunan, kondisi cuaca, harga, kondisi tanah, dan alasan estetika

Dari uraian definisi diatas, penulis dapat menyimpulkan Bangunan merupakan wujud fisik hasil pekerjaan konstruksi yang menyatu dengan tempat.

## Definisi POS *(Point Of Sale)*

Menurut Sugihartono, Satoto dan Widianto (2015:445), *Point of Sale* (POS) merupakan aplikasi yang digunakan dalam membantu pengolahan data-data seperti data pembelian, penjualan eceran, transaksi hutang, transaksi retur pembelian, dan pelaporan transaksi yang dibutuhkan dalam pengambilan keputusan oleh para pebisnis.

POS terdiri dari perangkat keras *(PC, receipt printer, cash drawer, terminal pembayaran, barcode scanner*) dan perangkat lunak (manajemen *stock*, pelaporan, pembelian, manajemen pelanggan, standar keamanan transaksi, dan proses retur) dimana kedua komponen tersebut digunakan dalam setiap proses transaksi yang terjadi (Yuarita dan Marisa, 2017:168).

Perangkat lunak POS adalah komponen utama dari POS yang akan menentukan proses berjalannya sistem tersebut, seperti apa yang harus dilakukan dan bagaimana cara melakukannya. Akan tetapi, perangkat keras POS dibutuhkan dalam membantu menjalankan fungsi POS, membantu proses transaksi pembayaran dan membuat tanda terima untuk pelanggan.

Perangkat keras dapat disesuaikan dengan kebutuhan organisasi agar dapat membantu mempercepat proses pemasukkan barang hingga proses pelayanan penjualan (Novita dan Djatikusuma, 2010). Manfaat penggunaan *Point of Sale* disebuah organisasi adalah untuk mengolah transaksi-transaksi, mengurangi biaya serta menghasilkan pendapatan yang menguntungkan bagi organisasi tersebut sebagai salah satu produk maupun pelayanan yang diberikan organisasi kepada pelanggannya. Penggunaan *Point of Sale* membantu mempertahankan persediaan stok barang di tingkat paling rendah agar konsisten dengan jenis barang yang tersedia sehingga organisasi dapat menghindari terjadinya stok habis dan pelanggan yang kecewa akan produk habis (Sugihartono, Satoto dan Widianto, 2015:445).

Dari definisi diatas penulis dapat menyimpulkan bahwa POS (*Poin Of Sale)* merupakan sistem atau aplikasi yang penerapannya di dampingi oleh hardware yang diperlukan guna untuk membantu dalam pengolahan data yang digunakan pada proses transaksi.

## Definisi Toko

Menurut Wikipedia Toko adalah sebuah tempat tertutup yang di dalamnya terjadi kegiatan perdagangan dengan jenis benda atau barang yang khusus, misalnya toko buku, toko buah, dan sebagainya. Secara fungsi ekonomi, istilah "toko" sesungguhnya hampir sama dengan "kedai" atau "warung". Akan tetapi pada perkembangan istilah, kedai dan warung cenderung bersifat tradisional dan sederhana, dan warung umumnya dikaitkan dengan tempat penjualan makanan dan minuman.

Secara bangunan fisik, toko lebih terkesan mewah dan modern dalam arsitektur bangunannya daripada warung. Toko juga lebih modern dalam hal barang-barang yang dijual dan proses transaksinya.

Dari definisi diatas penulis dapat menyimpulkan bahwa, toko merupakan suatu tempat dimana terjadinya suatu kegiatan jual beli atau transaksi berupa barang antara penjual dan pembeli, dimana mempunyai tujuan suatu tujuan tertentu untuk mecapai kesepakatan bersama.

## Definisi PHP (*Hypertext Preprocessor*)

PHP adalah bahasa pemrograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. (Sibero, 2011 “PHP (Hypertext Preprocessor)

PHP adalah sebuah bahasa pemrograman yang berbentuk scripting, sistem kerja dari program ini adalah sebagai interpreter bukan sebagai compiler”. (Nugroho, 2013)

Dari definisi diatas penulis dapat menyimpulkan bahwa, bahasa pemrograman php merupakan bahasa pemrograman yang digunakan untuk mengembangkan situs *web* statis atau situs *web* dinamis atau Aplikasi *Web*. PHP singkatan dari *Hypertext Pre-processor*, yang sebelumnya disebut Personal Home Pages.

## Definisi CSS (*Cascading Style Sheet*)

(Bekti, 2015) Berdasarkan pendapat yang telah dikemukakan dapat disimpulkan bahwa CSS (Cascading Style Sheets) adalah salah satu jenis bahasa pemrograman untuk mengatur proses pengolahan pada komponen tampilan *web* menjadi bentuk *web* yang lebih indah dan menarik.

CSS merupakan singkatan dari *Cascading Style Sheet*. Kegunaannya adalah untuk mengatur tampilan dokumen HTML, contohnya seperti pengaturan jarak antar baris, teks, warna dan format border bahkan penampilan file gambar. (Jayan, 2010)

Dari definisi diatas penulis dapat menyimpulkan CSS, merupakan *script code* yang berfungsi untuk mengatur *layout* web, tulisan, warna, seluruh *style* tampilan halaman web.

## Definisi HTML (*Hyper Text Markup Language*)

*Hyper Text Markup Language* atau HTML adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen web. (Sibero, 2013)

HTML (*Hyper Text Markup Language*) adalah simbol-simbol atau tag-tag yang dituliskan dalam sebuah file yang dimaksudkan untuk menampilkan halaman pada *web browser*. Tag-tag HTML selalu diawali dengan dan diakhiri dengan dimana x tag HTML seperti b, i, u dan sebagainya. Namun ada juga tag yang tidak diakhiri dengan tanda, Astamal (2006:8)

Dari definisi diatas penulis dapat menyimpulkan CSS, merupakan *script code* yang berfungsi untuk mengatur *layout* web, tulisan, warna, seluruh *style* tampilan halaman web.

## Definisi Basis Data (*Database*)

*Database* adalah koleksi terpadu dari data-data yang saling berkaitan dari suatu *enterprise* (toko, instansi pemerintahan atau swasta). (Ladjamudin, 2013), *Database* (Basis Data) merupakan sekumpulan data yang sangat kompleks, kemudian data tersebut memiliki hubungan antara data yang satu dengan data yang lainya. (Nugroho, 2013)

Dari definisi diatas penulis dapat menyimpulkan *Database*, merupakan sekumpulan data yang tersimpan secara *digital* yang mempunyai hubungan antara satu dengan data yang lain

## Definisi DBMS (*Database Management System*)

DBMS adalah “*Database Management System* (DBMS) merupakan kumpulan program aplikasi yang digunakan untuk mengelola basis data. Taufik, 2013). DBMS adalah kumpulan program yang digunakan untuk mendefinisikan, mengatur, dan memproses *database*, sedangkan *database* itu sendiri esensinya adalah sebuah struktur yang dibangun untuk keperluan penyimpanan data. (Budi Raharjo, 2011).

Dari definisi diatas penulis dapat menyimpulkan DBMS , merupakan suatu tempat atau sekumpulan berkas data yang sudah terkomputerisasi yang bertujuan untuk memelihara data atau informasi.

## Definisi UML (*Unified Modeling Language*)

Menurut Hend (2006:5) “*Unified Modeling Language* (UML) adalah bahasa yang telah telah menjadi standard untuk *visualisasi*, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak”. UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misanya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. (He UML adalah metodologi kolaborasi antara metoda-metoda Booch, OMT (Object Modeling Technique), serta OOSE (*Object Oriented Software Enggineering*) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa ‘pemrograman berorientasi objek’ (OOP). (Nugroho, 2009) rlawati, 2011).

Diagram berbentuk grafik yang menunjukan simbol elemen model yang

disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis-jenis diagram antara lain:

* 1. *Use Case* Diagram

*Use Case* Diagram merepresentasikan sejumlah external *actors* dan hubungannya ke use case yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity* diagram.

* 1. *Class* Diagram

*Class* Diagram adalah pandangan aplikasi yang bersifat statis. *Class* Diagram tidak hanya menggambarkan *visualisasi*, tetapi juga menggambarkan dan mendokumentasikan aspek yang berbeda dalam sistem, tetapi juga untuk kontruksi eksekusi kode dalam *software* atau aplikasi.

Class Diagram digunakan untuk mengelompokan hal-hal inti dari setiap proses yang ingin dilakukan. Semua proses dimasukkan ke dalam tiap-tiap *class* dan saling dihubungkan pada *class-class* lainnya yang saling berhubungan.

* 1. *Activity* Diagram

Diagram ini menggambarkan rangkaian aliran dari aktivitas didalam sebuah sistem yang bersifat dinamis, diagram ini digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

* 1. *Sequence* Diagram

Diagram ini menggambarkan kolaborasi dinamis antara sejumlah *object* yang bersifat dinamis. Kegunaanya untuk menunjukkan rangkaian pesan yang dikirim terhadap *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik

tertentu dalam eksekusi sistem.

* 1. *Communication* Diagram

Diagram ini berfungsi sebagai pengganti diagram kolaborasi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.

* 1. *Component* Diagram

Diagram komponen ini memperlihatkan organisasi serta kebergantungan

sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.

* 1. *Deployment* Diagram

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*) yang bersifat statis. Memuat simpul-simpul berserta komponen-komponen yang ada di dalamnya.Diagram deployment berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen.Diagram ini sangat berguna saat aplikasi berlaku sebagai aplikasi yang

dijalankan pada banyak mesin (distributed computing).

* 1. *Package* Diagram

Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian diagram komponen yang bersifat statis.

* 1. *Statechart* Diagram

Menggambarkan 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.

Dari definisi diatas penulis dapat menyimpulkan UML (*Unified Modeling Language),* Definisi UML merupakan model atau bahasa yang sudah menjadi standar pada *visualisasi*, perancangan, dan juga pendokumentasian sistem aplikasi yang akan dibangun.

## Definisi Motode Pengujian (*Blackbox*)

“*Blackbox* *testing* merupakan tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja *internal*nya”. Sedangkan menurut Mustaqbal, dkk (2015:34) “*BlackBox Testing* befokus pada spesifikasi fungsional dari perangkat lunak, kumpulan kondisi input danmelakukan pengetesan pada fungsional program”. Menurut Rizky (2011:264) “*Blackbox testing* merupakan menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”. Menurut Rosa dan Salahuddin (2015:275).

Menurut Simarmata Janner (2010 : 316) Klasifikasi *blackbox testing* mencakup beberapa pengujian yaitu :

1. Pengujian fungsional (*functional testing*)
2. Pengujian tegangan (*stress testing*)
3. Pengujian beban (*load testing*)
4. Pengujian khusus (*ad-hoc testing*)

Dari definisi diatas penulis dapat menyimpulkan metode pengujian *Blackbox*, merupakan metode untuk menguji kelancaran program yang telah dibuat sebelumnya. Pengujian dilakukan agar program tidak terjadi kesalahan ketika dijalankan.

# *Tools* yang digunakan

Untuk memudahkan dalam pembuatan sistem, maka diperlukan beberapa alat atau *tools* yang digunakan dalam pembuatan aplikasi penjualan barang bahan bangunan, serta untuk proses pelaksanaan pembuatan sistem yang akan dibangun.

## *Bootstrap*

Menurut para ahli Bootstrap merupakan kerangka kerja front-end atau *design* untuk pengembangan *web* yang lebih cepat dan mudah tentunya gratis. Bootstrap mencakup *template* desain berbasis HTML dan CSS untuk tipografi, bentuk, tombol, tabel, navigasi, modals, korsel gambar dan banyak lainnya, serta plugin JavaScript opsional. Bootstrap juga memberikan sobat kemampuan untuk membuat desain responsif dengan mudah.

Responsive *Layout* maka aplikasi *web* yang didesain dengan menggunakan Bootstrap akan langsung menyesuaikan dengan lebar dari media perambahannya. Sehingga *Framework* ini support untuk semua jenis device baik dari *smartphone, tablet*, laptop ataupun PC Desktop. Di samping itu, Boostrap juga sudah support untuk HTML 5 dan C.

## Balsamiq *Mockup*

Balsamiq mockup merupakan suatu program aplikasi yang digunakan, dalam pembuatan tampilan *user interface* sebuah aplikasi. *Software* ini sudah menyediakan *tools* yang dapat memudahkan dalam membuat desain *prototyping* aplikasi yang akan dibuat. Berfokus pada konten yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna.

Menggambar sketsa (*wireframe*) atau *prototype* rancangan desain *website* di atas kertas balsamiq *mockups* membantu seorang *web designer* dapat membuat tampilan *web* dalam bentuk gambar di komputer. Tujuannya agar membuat tampilan (*design*) *website* menarik juga dapat menyesuaikan dengan kebutuhan *customer* (pelanggan). Dengan alat pembuat *mockup* maka seorang *web designer* dapat menganalisa tata letak, desain dan fungsi.

## *CodeIgniter*

CodeIgniter adalah salahsatu *framework* PHP yang dibuat berdasarkan *model view Controlleer* (MVC). CI memiliki library lengkap untuk mengerjakan operasi-operasi yang dibutuhkan oleh aplikasi berbasis web misalnya mengakses *database*, *memvalidasi form* sistem yang akan dikembangkan mudah.

Codelgniter (*CI*) merupakan suatu *Framework* pengembangan aplikasi (*Application Development Framework*) dengan menggunakan bahasa PHP, pengembangan dapat langsung menghasilkan program dengan cepat, dengan mengikuti kerangka kerja membuat yang telah disiapkan oleh *Framework* CI. Dengan menggunkan *Framework* tidak perlu membuat program dari awal, tetapi sudah diberikan *library* fungsi-fungsi yang sudah diorganisasi untuk dapat membuat suatu program dengan cepat. Hanya perlu memasukkan data yang akan diproses dan bagaimana menampilkannya.

## *Google Chrome*

Google Chrome merupakan suatu aplikasi peramban yang digunakan untuk menjelajah dunia maya seperti halnya Firefox, Opera ataupun Microsoft Edge. Jika Firefox dikembangkan oleh Mozilla, Google Chrome dibuat dan dirancang oleh Google, toko internet terbesar di dunia yang juga empunya Android.

Proyek yang dimiliki yakni *open source* digunakan oleh Google disebut Chromium, menggunakan mesin rendering Webkit sampai dengan versi 27 dan dirancang untuk bekerja dengan kecepatan di atas rata-rata namun tetap ringan dijalankan di perangkat desktop dan mobile.

## Visual Studio *Code*

Visual Studio *Code* (*VS Code*) merupakan suatu teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac,* dan *Windows*. Teks editor ini secara langsung mendukung bahasa pemrograman *JavaScript, Typescript,* dan *Node.js,* serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang *via marketplace* Visual Studio Code (seperti *C++, C#, Python, Go, Java*, dst).

Banyaknya fitur yang disediakan oleh Visual Studio Code, diantaranya *Intellisense*, *Git Integration, Debugging*, dan fitur ekstensi yang menambah kemampuan teks editor. Fitur tersebut akan terus bertambah seiring dengan bertambahnya versi Visual Studio Code. Pembaruan versi Visual Studio Code ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan VS Code dengan teks editor-teks editor yang lain.

## MySQL

MySQL merupakan suatu DBMS singkatan dari (*Database Management System*) menggunakan perintah SQL (*Structured Query Language*) yang banyak digunakan saat ini dalam pembuatan aplikasi berbasis website. MySQL dibagi menjadi dua lisensi, pertama adalah *Free Software* dimana perangkat lunak dapat diakses oleh siapa saja. Dan kedua adalah *Shareware* dimana perangkat lunak berpemilik memiliki batasan dalam penggunaannya.

MySQL termasuk ke dalam RDBMS singkatan daari (*Relational Database Management System*). Sehingga, menggunakan tabel, kolom, baris, di dalam struktur database -nya. Jadi, dalam proses pengambilan data menggunakan metode relational database. Dan juga menjadi penghubung antara perangkat lunak dan database server.

## Xampp

XAMPP merupakan suatu paket perangkat lunak (*software*) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB (sekarang), PHP, dan Perl. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah cross platform sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS Windows, Mac OS, dan juga Solaris.

XAMPP merupakan sebuah web server multi-platform yang bebas lisensi

atau gratis, yang terdiri dari Apache HTTP Server, database MySQL dan penerjemah untuk script yang dituliskan dalam PHP dan bahasa pemrograman. XAMPP dipakai untuk membuat web server lokal di komputer. Hal ini akan memudahkan Anda dalam mengembangkan, mendesain, dan *keperluan testing website*.

## Visual Paradigm

Visual Paradigm suatu *software* model sistem visualisasi yang memungkinkan model yang telah dibuat dapat digunakan sebagai representasi proyek-proyek lain dilengkapi dengan beberapa fitur lengkap.

Visual Paradigm sebuah software model dengan sistem *visualisasi* memungkinkan model yang telah dibuat dapat digunakan sebagai representasi proyek-proyek lain dilengkapi dengan beberapa fitur yang ada didalamnya sampai pada menganalisa sebuah proyek yang akan dikerjakan. Diagram dapat disusun sedemikian rupa sehingga dapat dipustakakan menjadi proyek per proyek yang saling berkaitan.

# BAB III

# ANALISIS DAN PERANCANGAN SISTEM

# Analisis Sistem

Aplikasi penjualan bahan bangunan *Point Of Sale* (POS) yang dibangun merupakan suatu sistem yang dapat melakukan proses transaksi, pengelolaan barang yakni persediaan barang masuk (*stock in)*, persediaan barang keluar (*stock out)*, aplikasi *Point Of Sale* (POS) umumnya terdapat perangkat keras dan perangkat lunak yang digunakan seperti *scanner* *barcode*  ketika melakukan *checkout* barang pada aplikasi yang digunakan,.

Aplikasi panjualan bahan bangunan *Point Of Sale* (POS) ber *evolusi* setiap tahunnya, sesuai dengan perkembangan ilmu pengetahuan dan teknologi, dan banyak digunakan di berbagai toko, terutama toko yang bergerak di bidang perdagangan untuk memperoleh kemudahan proses perhitungan barang pesanan oleh pelanggan atau *customer*, pencarian barang, pembuatan laporan secara cepat. Proses transaksi dapat dikategorikan sebagai berikut

1. Pengelolaan *stock in* barang

Pada tahap pengelolaan *stock in* barang, *suppplier* mengirimkan barang ke toko satria jaya, pada proses ini terjadi perubahan pada *stock* *in* barang atau barang bartambah, dan akan mempengaruhi perubahan *stock* di gudang

1. Pengelolaan *stock out* barang

Pada tahap pengelolaan *stock out*  barang, suppplier mengirimkan barang ke toko satria jaya, dikarenakan barang tidak sesuai, maka barang dicatat pada *stock out* barang, pada proses ini terjadi perubahan pada *stock in* barang atau barang berkurang, sedangkan pencatatan pada *stock out* barang bertambah, dan akan mempengaruhi perubahan *stock* di gudang.

1. Penjualan barang

Pada tahap ini penjualan barang terjadi antara penjual dan pembeli (customer), proses penjualan ini akan mempengaruhi perubahan terhadap stock barang di gudang,.

# Analisis Masalah

Masalah yang timbul terjadi pada operasional toko, umumnya setiap masalah yang timbul bermacam - macam tergantung kondisi dalam toko, dengan harapan setiap masalah yang terjadi dapat ditemukannya solusi atau pemecahan masalah.

Masalah yang timbul di toko Bangunan Satria Jaya yakni terjadi perhitungan barang secara manual, transaksi terhadap *customer* atau pelanggan masih menggunkan kalkulator, pencarian barang secara manual, pencatatan laporan menggunakan media tulis tangan, kehilangan data atau catatan, potongan harga yang keliru atau tidak sesuai perhitungan*,*.

Terjadinya proses pengelolaan *stock* barang yang masih tidak sesuai dengan *stock* barang yang tersedia di gudang, serta proses penjualan barang terhadap pelanggan atau *customer*, umumnya mempengaruhi *stock* barang di gudang, namun yang terjadi kadang kala tidak dilakukan proses perubahan terhadap *stock* di gudang, karena petugas toko lupa untuk melakukan perubahan *stock* di gudang.

# Analisis sitem yang sedang berjalan

Berikut ini adalah proses bisnis berdasarkan, analisa sistem yang sedang berjalan, sebagai berikut:

1. Pengelolaan *stock* barang
2. *Supplier* melakukan pengiriman barang untuk toko satria jaya
3. Toko Satria Jaya melakukan pemeriksaan atau pengecekan barang yang dikirim oleh *supplier*
4. Jika barang yang diterima sesuai (tidak terdapat cacat atau rusak), maka barang dicatat pada *stock in*  sebagai penerimaan barang di gudang, jika barang yang diterima tidak sesuai, cacat atau rusak maka barang dicatat sebagai *stock out*  dan dilakukan pengiriman kepada *supplier*

|  |
| --- |
|  |
| Gambar 3.1 Analisis Proses Bisnis Pengelolaan *Stock* Barang |

Berikut ini merupakan proses pengelolaan *stock* barang, berdasarkan analis sistem berjalan dari uraian diatas

1. Penjualan
2. *Customer*  atau pelanggan melakukan pesanan barang
3. Toko Satria Jaya menerima pesanan dan menghitung total pemesanan.
4. *Customer* atau pelanggan, mengecek barang yang di pesan, jika sesuai maka dilakukan pembayaran, jika tidak sesuai maka melakukan pesanan kembali barang.
5. Toko Satria Jaya menerima pembayaran

|  |
| --- |
|  |
| Gambar 3.2 Analisis Proses Bisnis dari Pelanggan ke Toko Satria Jaya |

# Analisis kebutuhan non fungsional

Kebutuhan *non* Fungsional adalah kebutuhan yang menitikberatkan pada properti prilaku yang dimiliki oleh sistem, atau suatu fungsi yang menggambarkan kebutuhan yang diperlukan untuk menjalankan aplikasi yang dibangun. Untuk menjalankan aplikasi penjualan bahan bangunan POS *(Point Of Sale)* ada beberapa kebutuhan yang harus dipersiapkan diantaranya yaitu kebutuhan perangkat keras, kebutuhan perangkat lunak, dan pengguna yang menggunakan aplikasi. Analisa kebutuhan non fungsional dibangun dengan tujuan aplikasi dapat di digunakan oleh pemilik atau karyawan Satria Jaya.

## Analisa Pengguna

Analisa pengguna merupakan suatu aktivitas yang menunjukan siapa saja yang menggunakan sistem dan apa saja yang bisa dilakukan oleh pengguna di dalam sistem. Berikut ini merupakan pengguna dari aplikasi penjualan bahan bangunan *Poin Of Sale* Satria Jaya:

1. Admin

Admin memiliki tugas untuk mengelola dan menggunakan sistem secara keseluruhan, adapun syarat yang harus dimiliki yaitu sebagai berikut:

1. Paham dan dapat mengoperasikan komputer dengan baik dan benar
2. Teliti dan jujur
3. Memahami alur secara keseluruhan dari aplikasi penjualan bahan bangunan *Point Of Sale*
4. Kasir

Kasir memiliki tugas untuk menggunakan sistem, adapun syarat yang harus dimiliki yaitu sebagai berikut:

1. Paham dan dapat mengoperasikan komputer dengan baik dan benar
2. Teliti dan jujur
3. Bisa menggunakan aplikasi penjualan bahan bangunan *Point Of Sale*

## Analisa kebutuhan perangkat keras

Analisa kebutuhan perangkat keras atau hardware bertujuan agar aplikasi penjualn bahan bangunan poin of sale dapat berjalan dengan baik,spesifikasi yang dibutuhkan untuk menjalankan aplikasi penjualan bahan bangunan point of sale yaitu sebagai berikut:

### Tabel 3.1 Tabel Kebutuhan Perangkat Keras

|  |  |  |
| --- | --- | --- |
| No | Perangkat | Spesifikasi |
| 1 | *Processor* | *Dual core i3 up to* |
| 2 | *Memory Ram* | *3 GB up to* |
| 3 | *Hardisk* | 500 GB *up to* |
| 4 | *Monitor* | *Standart* |
| 5 | *Keyboard* | *Standart* |
| 6 | *Mouse* | *Standart* |

## Analisa kebutuhan perangkat lunak

Analisa perangkat lunak merupakan suatu perangkat lunak yang berkaitan dengan operasional kerja komputer, dan dapat digunakan oleh komputer untuk menjadi pendukung agar sistem dapat berjalan dengan baik,diantaranya dalah sebagai berikut

1. Sistem Operasi

Sistem operasi digunakan sebagai perantara antara perangkat lunak dan perangkat keras komputer terdapat berbagai macam sistem operasi baik itu *Windows* 7/8/10 atau *linux* dalam aplikasi ini sistem operasi yang di rekomendasikan yaitu *windows* 10 karena mudah dalam operasional maupun sistem yang *up date*. Sistem operasi menjadi pengatur atau bertanggung jawab dalam penggunaan perangkat keras dan bertugas untuk menjalankan perangkat lunak atau aplikasi yang digunakan

1. Database

Dengan menggunakan *database*, maka pengguna diberikan fasilitas untuk melakukan penyimpanan, pengolahan, dan transformasi data dalam basis data, dalam aplikasi ini *database* yang di rekomendasikan yakni MYSQL karena mudah dalam penggunaan dan lebih familiar.

1. Server lokal

Dengan menggunakan *server* lokal maka pengguna diberikan fasilitas untuk menggunakan database mysql, untuk *server* lokal yang direkomendasikan yakni xampp, karena lebih *portable*,familiar dan mudah untuk digunakan.

# Analisis kebutuhan fungsional

Analisis aplikasi penjualan barang bahan bangunan *point of sale* yang akan di bangun pada Toko Bahan Bangunan Satria Jaya dirancang menggunakan UML (*Unifed Modeling Leanguage*), diantaranya *Secenario-Based Modeling* yang meliputi *Use Case Secenario* (Naratif). *Actor List*, *Use Case List, Use Case Diagram,* dan *Swimlane Diagram*. *Data modeling* yang meliputi *Class Diagram,* dan *Behavioral Modeling* yaitu *Sequence Diagram*.

## *Actor List*

Aktor merupakan orang yang menggunakan sistem atau perangkat baik *software/hardware*, serta memiliki sifat dan fungsi yang dijelaskan dalam sebuah konteks yang memiliki peran terhadap sistem, berikut adalah pendefinisian aktor pada aplikasi penjualan barang bahan bangunan (*pont of sale*) di toko bangunan satria jaya

### Tabel 3.2 *Actor List*

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | Admin | Orang yang mengelola sistem dan akun untuk pengguna atau *user*, serta menerima laporan dari setiap transaksi pada sistem yang berjalan. |
| 2 | Kasir | Orang yang mempunyai hak akses untuk masuk ke sistem dan melakukan proses transaksi. |

## *Usecase Scenario* (Naratif)

*Usecase scenario* merupakan urutan langkah-langkah dalam proses yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Berikut merupakan use case scenario aplikasi penjualan bahan bangunan *point of sale* Satria Jaya sebagai berikut:

1. *Usecase Scenario Login*

### Tabel 3.3 *Usecase Scenario Login*

|  |  |
| --- | --- |
| **Identifkasi** | |
| **Nama** | *Login* |
| **Aktor** | Admin,Kasir |
| **Deskripsi** | *Usecase* ini menggambarkan sistem *login*, dimana setiap aktor melakukan *login* sesuai dengan hak aksesnya, *username* dan *password* di input untuk dapat mengakses halaman utama. |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reakasi Sistem** |
| 1. Membuka Aplikasi | 1. Menampilkan Halaman *Login* |
| 1. Mengisi form *login*, lalu klik “*Login*” atau “*Cancel*” jika batal | 1. Validasi Username dan Password |
|  | 1. Menampilkan pesan “*Login* Berhasil Selamat Datang” jika *username* dan *password* cocok |
| 1. Klik oke | 1. Menampilkan Halaman Utama |
| **Skenario Alternatif (Proses Gagal)** | |
|  | 1. Menampikan pesan “*username password* salah” |
| 1. Klik oke | 1. Menampilkan kembali form *login* |
| **Kondisi Akhir** | Setiap aktor atau pengguna masuk kedalam sistem sesuai hak aksesnya. |

### Tabel 3.4 Usecase Scenario Mengelola *Supplier*

|  |  |
| --- | --- |
| **Identifkasi** | |
| **Nama** | Mengelola *Supplier* |
| **Aktor** | Kasir |
| **Deskripsi** | *Usecase* ini menggambarkan aktivitas admin dalam mengelola *supplier* atau pemasok barang seperti tambah,ubah,hapus *supplier.* |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reakasi Sistem** |
|  | 1. Memeriksa status *login* |
| 1. Memasukan data *supplier* sesuai dengan kolom yang ada | 1. Memeriksa validasi sesuai kolom yang telah di isi |
| 1. Klik Simpan Data | 1. Menyimpan data *supplier* ke basis data |
|  | 1. Menampilkan pesan “Data *Supplier* Berhasil Ditambahkan” |
| **Skenario Alternatif (Proses Gagal)** | |
|  | 1. Menampilkan pesan “Data yang anda masukan tidak *valid*” |
| 1. Memperbaiki data masukan yang tidak *valid* | 1. Menampilkan pesan “Data *Supplier* Berhasil Ditambahkan” |
| **Kondisi Akhir** | Kasir mempunyai akses menambah,ubah,hapus data *supplier* atau pemasok barang |

### Tabel 3.5 Usecase Scenario Mengelola *Customer*

|  |  |
| --- | --- |
| **Identifkasi** | |
| **Nama** | Mengelola *Customer* |
| **Aktor** | Kasir |
| **Deskripsi** | *Usecase* ini menggambarkan aktivitas kasir dalam mengelola *customer* atau pelanggan seperti tambah,ubah,hapus *customer.* |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reakasi Sistem** |
|  | 1. Memeriksa status *login* |
| 1. Memasukan data *customer* sesuai dengan kolom yang ada | 1. Memeriksa validasi sesuai kolom yang telah di isi |
| 1. Klik Simpan Data | 1. Menyimpan data *customer* ke basis data |
|  | 1. Menampilkan pesan “Data *Customer* Berhasil Ditambahkan” |
| **Skenario Alternatif (Proses Gagal)** | |
|  | 1. Menampilkan pesan “Data yang anda masukan tidak *valid*” |
| 1. Memperbaiki data masukan yang tidak *valid* | 1. Menampilkan pesan “Data *Customer* Berhasil Ditambahkan” |
| **Kondisi Akhir** | Kasir mempunyai akses menambah,ubah,hapus data *customer* atau pelanggan. |

### Tabel 3.6 Usecase Scenario Mengelola *Product*

|  |  |
| --- | --- |
| **Identifkasi** | |
| **Nama** | Mengelola *Product* |
| **Aktor** | Kasir |
| **Deskripsi** | *Usecase* ini menggambarkan aktivitas kasir dalam mengelola *product* seperti tambah, ubah, hapus *product* dimana setiap *product* mempunyai nama barang, kategori,satuan barang. |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reakasi Sistem** |
|  | 1. Memeriksa status *login* |
| 1. Klik menu *product* | 1. Menampikan menu barang, kategori, satuan barang |
| 1. Klik menu kategori | 1. Menampilkan kolom kategori |
| 1. Mengisi kolom kategori | 1. Memeriksa validasi sesuai kolom yang telah di isi |
| 1. Klik Simpan Data | 1. Menyimpan data kategorike basis data |
|  | 1. Menampilkan pesan “Data Kategori BarangBerhasil Ditambahkan” |
| 1. Klik menu satuan | 1. Menampilkan kolom satuan |
| 1. Mengisi kolom satuan | 1. Memeriksa validasi sesuai kolom yang telah di isi |
| 1. Klik Simpan Data | 1. Menyimpan data satuanke basis data |
|  | 1. Menampilkan pesan “Data Satuan BarangBerhasil Ditambahkan” |
| 1. Klik menu barang | 1. Menampilkan kolom barang |
| 1. Mengisi kolom barang | 1. Memeriksa validasi sesuai kolom yang telah di isi |
| 1. Klik Simpan Data | 1. Menyimpan data barangke basis data |
|  | 1. Menampilkan pesan “Data BarangBerhasil Ditambahkan” |
| **Skenario Alternatif (Proses Gagal)** | |
| 1. Mengisi kolom kategori | 1. Menampilkan pesan “Data yang anda masukan tidak *valid*” |
| 1. Memperbaiki data masukan yang tidak *valid* | 1. Menampilkan pesan “Data KategoriBerhasil Ditambahkan” |
| 1. Mengisi kolom Satuan | 1. Menampilkan pesan “Data yang anda masukan tidak *valid*” |
| 1. Memperbaiki data masukan yang tidak *valid* | 1. Menampilkan pesan “Data Satuan BarangBerhasil Ditambahkan” |
| 1. Mengisi kolom Barang | 1. Menampilkan pesan “Data yang anda masukan tidak *valid*” |
| 1. Memperbaiki data masukan yang tidak *valid* | 1. Menampilkan pesan “Data BarangBerhasil Ditambahkan” |
| **Kondisi Akhir** | Kasir mempunyai akses menambah,ubah,hapus data *product* yaitu meliputi barang, kategori dan satuan barang. |

### Tabel 3.7 Usecase Scenario Mengelola Transaksi

|  |  |
| --- | --- |
| **Identifkasi** | |
| **Nama** | Mengelola *Product* |
| **Aktor** | Kasir |
| **Deskripsi** | *Usecase* ini menggambarkan aktivitas kasir dalam mengelola transaksi, penjualan terhadap *customer* atau pelanggan, serta pengelolaan transaksi barang masuk (*stock in)*, terhadap supplier, ataupun berupa retur barang (*stock out)*, kasir mempunyai akses cari, tambah, ubah, hapusdata, menu transaksi meliputi penjualan, barang masuk (*stock in),* retur barang (*stock out)*,. |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reakasi Sistem** |
|  | 1. Memeriksa status *login* |
| 1. Klik menu transaksi | 1. Menampikan menu kasir, pengelolaan barang masuk (*stock in)*, pengelolaan barang keluar (*stock out).* |
| 1. Klik menu penjualan | 1. Sistem menampilkan halaman antarmuka transaksi penjualan barang, kasir mempunyai akses cari, tambah, ubah, hapus barang dan cetak transaksi. |
| 1. Memasukan data yang tersedia pada menu kasir,sesuai dengan kolom yang ada | 1. Memeriksa validasi sesuai kolom yang telah di isi |
| 1. Klik proses transaksi | 1. Menyimpan data transaksi, dan menampilkan data transaksi penjualan atau *struct* |
| 1. Klik tombol cetak | 1. Sistem mengeluarkan *output* data transaksi penjualan |
| 1. Klik menu pengelolaan barang masuk (*stock in)* | 1. Sistem menampilkan halaman antarmuka kelola barang masuk (*stock in)*, kasir mempunyai akses cari, tambah, ubah, hapus, simpan barang. |
| 1. Mengisi kolom yang tersedia pada menu *stock in* barang | 1. Memeriksa validasi sesuai kolom yang telah di isi |
| 1. Klik Simpan Data | 1. Menyimpan data *stock in* barangke basis data |
|  | 1. Menampilkan pesan “Data Barang MasukBerhasil Ditambahkan” |
| 1. Klik menu pengelolaan barang keluar (*stock out)* | 1. Sistem menampilkan halaman antarmuka kelola barang keluar (*stock out)*, kasir mempunyai akses cari, tambah, ubah, hapus, simpan barang. |
| 1. Mengisi kolom yang tersedia pada menu *stock out*  barang | 1. Memeriksa validasi sesuai kolom yang telah di isi |
| 1. Klik Simpan Data | 1. Menyimpan data *stock out* barangke basis data |
|  | 1. Menampilkan pesan “Data Barang KeluarBerhasil Ditambahkan” |
| **Skenario Alternatif (Proses Gagal)** | |
| 1. Mengisi kolom penjualan | 1. Menampilkan pesan “Data yang anda masukan tidak *valid*” |
| 1. Memperbaiki data masukan yang tidak *valid* | 1. Menyimpan data penjualan, dan menampilkan data transaksi penjualan atau *struct* |
| 1. Klik tombol cetak | 1. Sistem mengeluarkan *output* data transaksi penjualan |
| 1. Mengisi kolom *stock in* barang | 1. Menampilkan pesan “Data yang anda masukan tidak *valid*” |
| 1. Memperbaiki data masukan yang tidak *valid* | 1. Menampilkan pesan “Data Barang MasukBerhasil Ditambahkan” |
| 1. Mengisi kolom *stock out* barang | 1. Menampilkan pesan “Data yang anda masukan tidak *valid*” |
| 1. Memperbaiki data masukan yang tidak *valid* | 1. Menampilkan pesan “Data Barang KeluarBerhasil Ditambahkan” |
| **Kondisi Akhir** | Kasir mempunyai akses malakukan transaksi, cetak transaksi, menambah, ubah, hapus data barang |

### Tabel 3.8 Usecase Scenario Mengelola *User*

|  |  |
| --- | --- |
| **Identifkasi** | |
| **Nama** | Mengelola *User* |
| **Aktor** | Admin |
| **Deskripsi** | *Usecase* ini menggambarkan aktivitas admin dalam mengelola *user* atau pengguna sistem, admin mempunyai akses tambah, ubah, hapus *user* atau pengguna sistem,*.* |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reakasi Sistem** |
|  | 1. Memeriksa status *login* |
| 1. Memasukan data *user* atau penggunasesuai dengan kolom yang ada | 1. Memeriksa validasi sesuai kolom yang telah di isi |
| 1. Klik Simpan Data | 1. Menyimpan data *user* ke basis data |
|  | 1. Menampilkan pesan “Data *user* Berhasil Ditambahkan” |
| **Skenario Alternatif (Proses Gagal)** | |
|  | 1. Menampilkan pesan “Data yang anda masukan tidak *valid*” |
| 1. Memperbaiki data masukan yang tidak *valid* | 1. Menampilkan pesan “Data *user* atau PenggunaBerhasil Ditambahkan” |
| **Kondisi Akhir** | Admin mempunyai akses menambah,ubah,hapus data *user* atau pengguna sistem |

### Tabel 3.9 Usecase Scenario MengelolaLaporan

|  |  |
| --- | --- |
| **Identifkasi** | |
| **Nama** | MengelolaLaporan |
| **Aktor** | Admin |
| **Deskripsi** | *Usecase* ini menggambarkan aktivitas admin dalam mengelolaLaporan meliputi Laporan barang atau *stock* dan penjualan, admin mempunyai akses cetak dan hapus laporan |
| **Skenario Utama** | |
| **Aksi Aktor** | **Reakasi Sistem** |
|  | 1. Memeriksa status *login* |
| 1. Klik menu laporan | 1. Menampikan menu laporan penjualan dan laporan barang atau *stock* |
| 1. Klik menu laporan penjualan | 1. Sistem menampilkan halaman atau antarmuka penjualan barang, admin mempunyai akses cari, cetak, dan hapus laporan berdasarkan tanggal penjualan. |
| 1. Klik tombol cetak | 1. Sistem menampilkan halaman penjualan yang akan di cetak, kemudian sistem mengeluarkan data transaksi penjualan |
| 1. Klik tombol laporan barang atau *stock* | 1. Sistem menampilkan halaman atau antarmuka *stock* barang, admin mempunyai akses cari, cetak laporan secara keseluruhan |
| 1. Klik tombol cetak | 1. Sistem menampilkan halaman *stock* barang yang akan di cetak, kemudian sistem mengeluarkan data *stock* barang |
| **Skenario Alternatif (Proses Gagal)** | |
|  | 1. Menampilkan “Data yang anda cari kosong” |
| 1. Memperbaiki data masukan yang sesuai | 1. Table menampilkan data yang di cari di urutkan berdasarkan huruf abjad |
| **Kondisi Akhir** | Admin mempunyai akses cari, cetak, dan hapus pada menu laporan. |

## *Usecase List*

Dafar *usecase* menjelaskan tentang *usecase* yang ada dalam sistem, daftar *usecase* berupa kode *usecase,* nama *usecase* dan keterangan dari tiap *usecase.* Adapun *usecase list* yang terdapat pada aplikasi penjualan bahan bangunan Toko Satria Jaya, sebagai berikut:

|  |  |  |
| --- | --- | --- |
| Kode | Sistem *Usecase* | Keterangan |
| U1 | *Login* | Digunakan untuk memastikan setiap *user* masuk kedalam sistem, sesuai hak aksesnya berdasarkan *username* dan *password* yang disertakan. |
| U2 | Kelola *supplier* | Digunakan oleh aktor untuk mengelola data *supplier* |
| U3 | Kelola *customer* | Digunakan oleh aktor untuk mengelola data *customer* atau pelanggan. |
| U4 | Kelola Produk | Digunakan oleh aktor untuk mengelola data produk meliputi *item*, kategori, satuan barang. |
| U5 | Kelola Transaksi | Digunakan oleh aktor untuk mengelola data transaksi meliputi penjualan, barang masuk (*stock in*), dan pengembalian barang (*stock out*). |
| U6 | Kelola *User* | Digunakan oleh aktor untuk mengelola data *user* atau pengguna sistem |
| U7 | Kelola Laporan | Digunakan oleh aktor untuk menerima laporan meliputi laporan penjualan dan laporan barang atau *stock* barang |

### Tabel 3.10 *Usecase* *List*

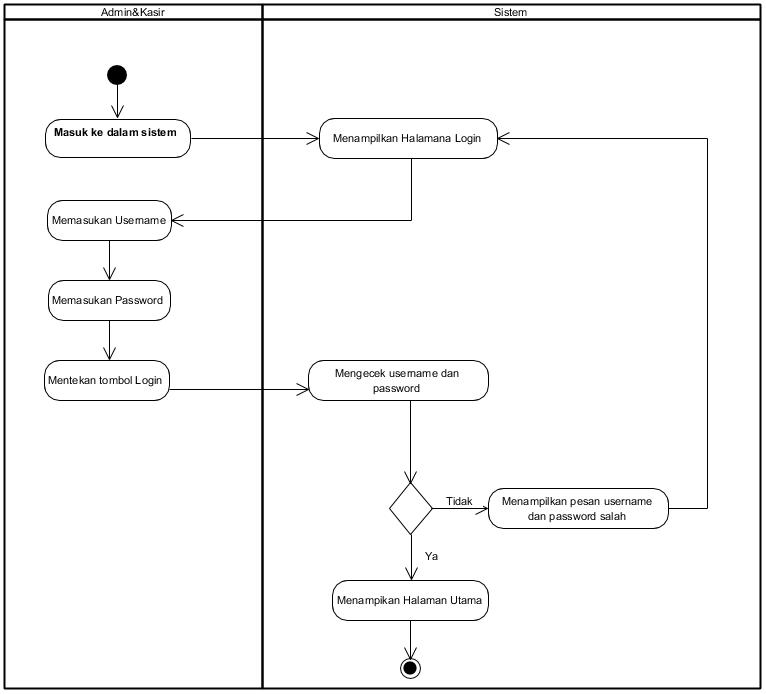
## *Usecase Diagram*

|  |
| --- |
|  |
| Gambar 3.3 *Usecase* Diagram |

Gambar 3.3 *Usecase* Diagram ini terdapat 7 fungsi dan 2 aktor yaitu admin dan kasir, dimana *usecase login* dapat di akses oleh aktor tersebut, sedangakan untuk *usec case* kelola *customer,* produk, dan kelola transaksi hanya dapat diakses oleh aktor kasir, sedangkan untuk *usecase* kelola *user* dan laporan hanya dapat diakses oleh aktor admin.

Dimana untuk fungsi atau *usecase* U4.Produk merupakan generalisasi dari fungsi kategori, item dan satuan barang , untuk *usecase* U5.Kelola Transaksi merupakan generalisasi dari fungsi penjualan, *stock in* barang, *stock out* barang, untuk *usecase* U7.Kelola Laporan merupakan generalisasi dari fungsi laporan penjualan dan laporan *stock* barang,

## *Swimline Diagram*

1. *Swimlne Diagram Login*

|  |
| --- |
|  |
| Gambar 3.4 *Usecase Diagram Login* |

Keterangan:

Gambar 3.4 *Swimeline Diagram Login* menunjukan aktfitas aktor yaitu *user* yang terdiri dari admin dan kasir, *user* masuk ke dalam sistem menampilkan halaman login, lalu *user* memasukan *username* dan *password* dan menekan tombol login, sistem akan memvalidasi data yang dimasukan, jika data tidak sesuai makan akan menampilkan pesan kesalahan, dan jika sesuai maka user akan masuk ke dalam sistem sesuai hak aksesnya.

1. *Swimlne* Diagram Kelola *Supplier*

|  |
| --- |
|  |
| Gambar 3.5 *Usecase Diagram Supplier* |

Keterangan:

Gambar 3.5 *Swimeline Diagram* Kelola *Supplier* menunjukan aktifitas *user* yaitu kasir saat masuk dalam sistem, meliputi proses cari, tambah, ubah, dan hapus data kelola *supplier*, proses pengisian *form supplier*, sistem akan mem-*validasi* data yang dimasukan oleh *user*, jika sesuai maka data akan tersimpan dalam basis data, jika tidak sesuai sistem akan menampilkan pesan kesalahan inputan data.

1. *Swimlne* Diagram Kelola Produk

|  |
| --- |
|  |
| Gambar 3.6 *Usecase Diagram* Produk |

Keterangan:

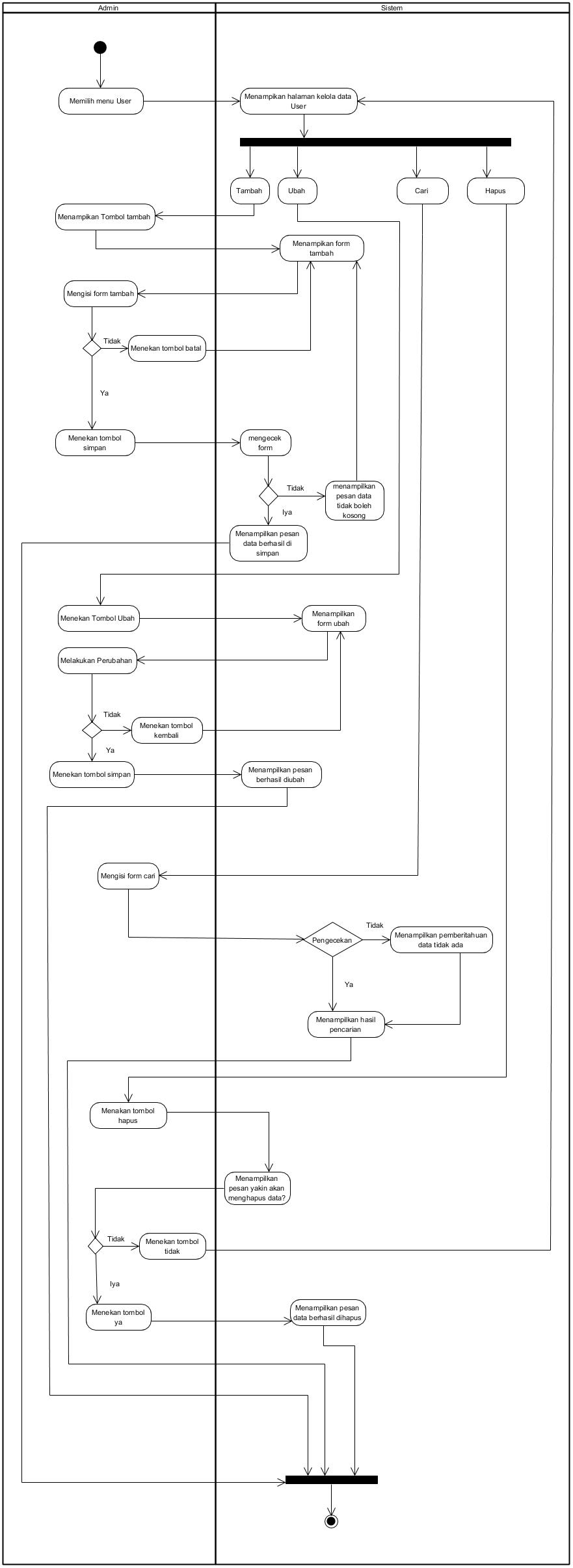
Gambar 3.6 *Swimeline Diagram* Kelola Produk meliputi kategori, item, dan satuan barang, menunjukan aktifitas *user* yaitu kasir saat masuk dalam sistem, meliputi proses cari, tambah, ubah, dan hapus data kelola produk, proses pengisian pada tiap-tiap menu pada *form* produk, sistem akan mem-*validasi* data yang dimasukan oleh *user*, jika sesuai maka data akan tersimpan dalam basis data, jika tidak sesuai sistem akan menampilkan pesan kesalahan inputan data.

1. *Swimlne* Diagram Kelola Transaksi

|  |
| --- |
|  |
| Gambar 3.7 *Usecase Diagram* Transaksi |

Keterangan:

Gambar 3.7 *Swimeline Diagram* Kelola Transaksi meliputi penjualan, *stock-in* barang, dan *stock-out* barang, menunjukan aktifitas *user* yaitu kasir saat masuk dalam sistem, meliputi proses cari, tambah, ubah, dan hapus data kelola transaksi, proses pengisian pada tiap-tiap menu pada *form* transaksi, sistem akan mem-*validasi* data yang dimasukan oleh *user*, jika sesuai maka data akan tersimpan dalam basis data, jika tidak sesuai sistem akan menampilkan pesan kesalahan inputan data.

1. *Swimlne* Diagram Kelola *User*

|  |
| --- |
|  |
| Gambar 3.8 *Usecase Diagram User* |

Keterangan:

Gambar 3.8 *Swimeline Diagram* Kelola *User* menunjukan aktifitas *user* yaitu admin saat masuk dalam sistem, meliputi proses cari, tambah, ubah, dan hapus data kelola *user*, proses pengisian *form user*, sistem akan mem-*validasi* data yang dimasukan oleh *user*, jika sesuai maka data akan tersimpan dalam basis data, jika tidak sesuai sistem akan menampilkan pesan kesalahan inputan data.

1. *Swimlne* Diagram Kelola Laporan

|  |
| --- |
|  |
| Gambar 3.9 *Usecase Diagram* Laporan |

Gambar 3.9 *Swimeline Diagram* Kelola Laporan meliputi laporan penjualan, dan laporan *stock* barang, menunjukan aktifitas *user* yaitu admin saat masuk dalam sistem, meliputi proses cari, cetak, dan hapus data, proses pencarian data berdasarkan pengisian data yang dimasukan oleh user, sistem akan mem-*validasi* data yang dimasukan oleh *user*, jika sesuai maka data akan ditemukan, jika tidak sesuai sistem akan menampilkan pesan bahwa data tidak ada atau tidak tersdia.

## *Sequence Diagram*

1. *Sequence Diagram Login*

|  |
| --- |
|  |
| Gambar 3.10 *Usecase Diagram Login* |

Keterangan:

Gambar 3.10 *Sequence Diagram Login* menggambarkan *user* yang terdiri dari admin dan kasir, *user* memasukan *username* dan *password* untuk masuk ke dalam sistem.

1. *Sequence Diagram Supplier*

|  |
| --- |
|  |
| Gambar 3.11 *Usecase Diagram Supplier* |

Keterangan:

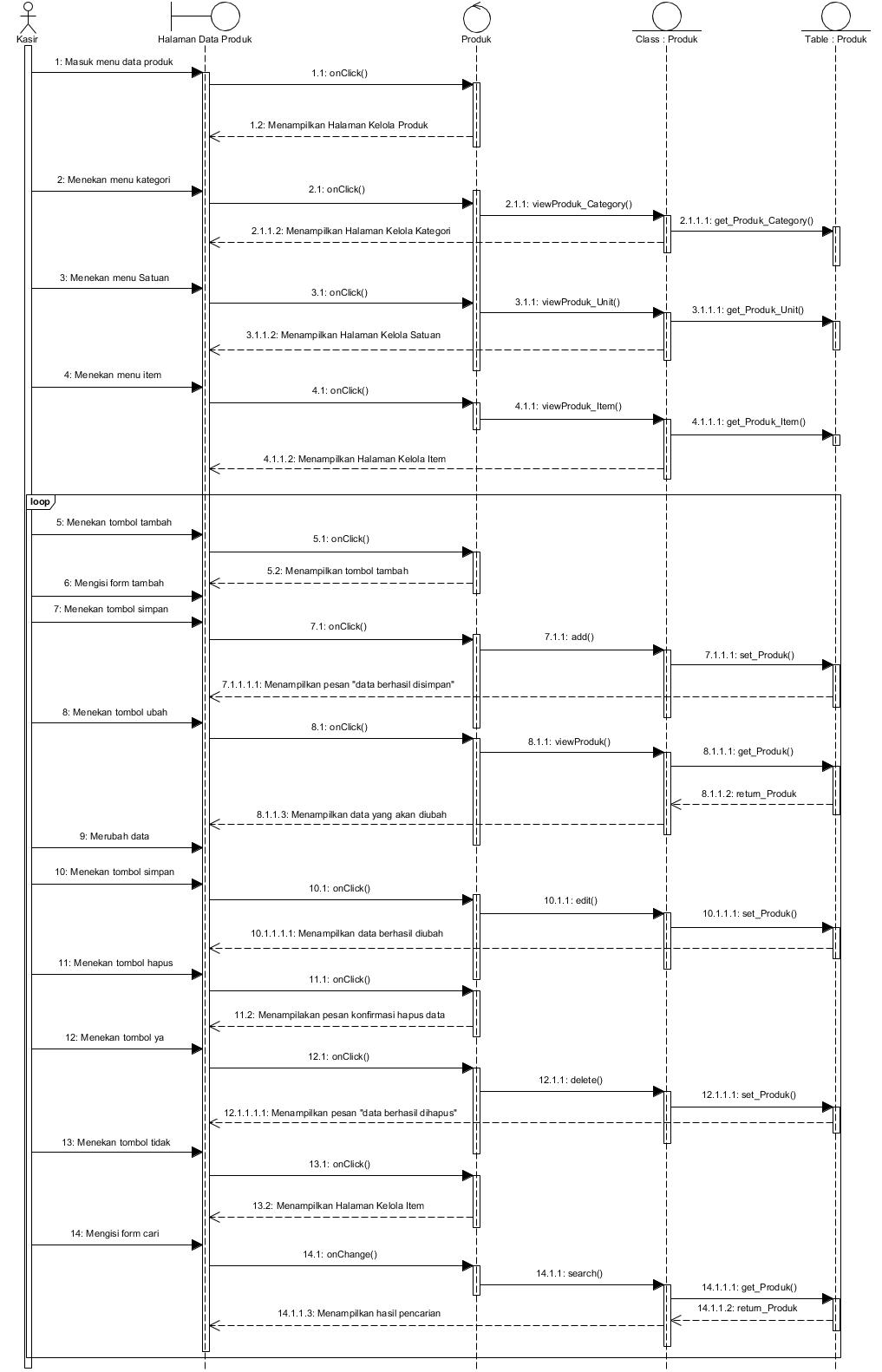
Gambar 3.11 *Sequence Diagram Supplier* menggambarkan kasir memiliki hak akses untuk melakukan kelola *supplier* meliputi tambah, ubah, hapus data *supplier*

1. *Sequence Diagram Customer*

|  |
| --- |
|  |
| Gambar 3.12 *Usecase Diagram Customer* |

Keterangan:

Gambar 3.12 *Sequence Diagram Customer* menggambarkan kasir memiliki hak akses untuk melakukan kelola *customer* meliputi tambah, ubah, hapus data *customer*

1. *Sequence Diagram* Produk

|  |
| --- |
|  |
| Gambar 3.13 *Usecase Diagram* Produk |

Keterangan:

Gambar 3.13 *Sequence Diagram* Produk menggambarkan kasir memiliki hak akses untuk kelola produk, kelola produk memiliki menu kategori, satuan, dan item produk, setiap menu bisa melakukan tambah, ubah, hapus yang disertakan bentuk *loop* yang berarti menunjukan proses yang sama terjadi pada setiap menu tersebut.

1. *Sequence Diagram* Transaksi

|  |
| --- |
|  |
| Gambar 3.14 *Usecase Diagram* Transaksi |

Keterangan:

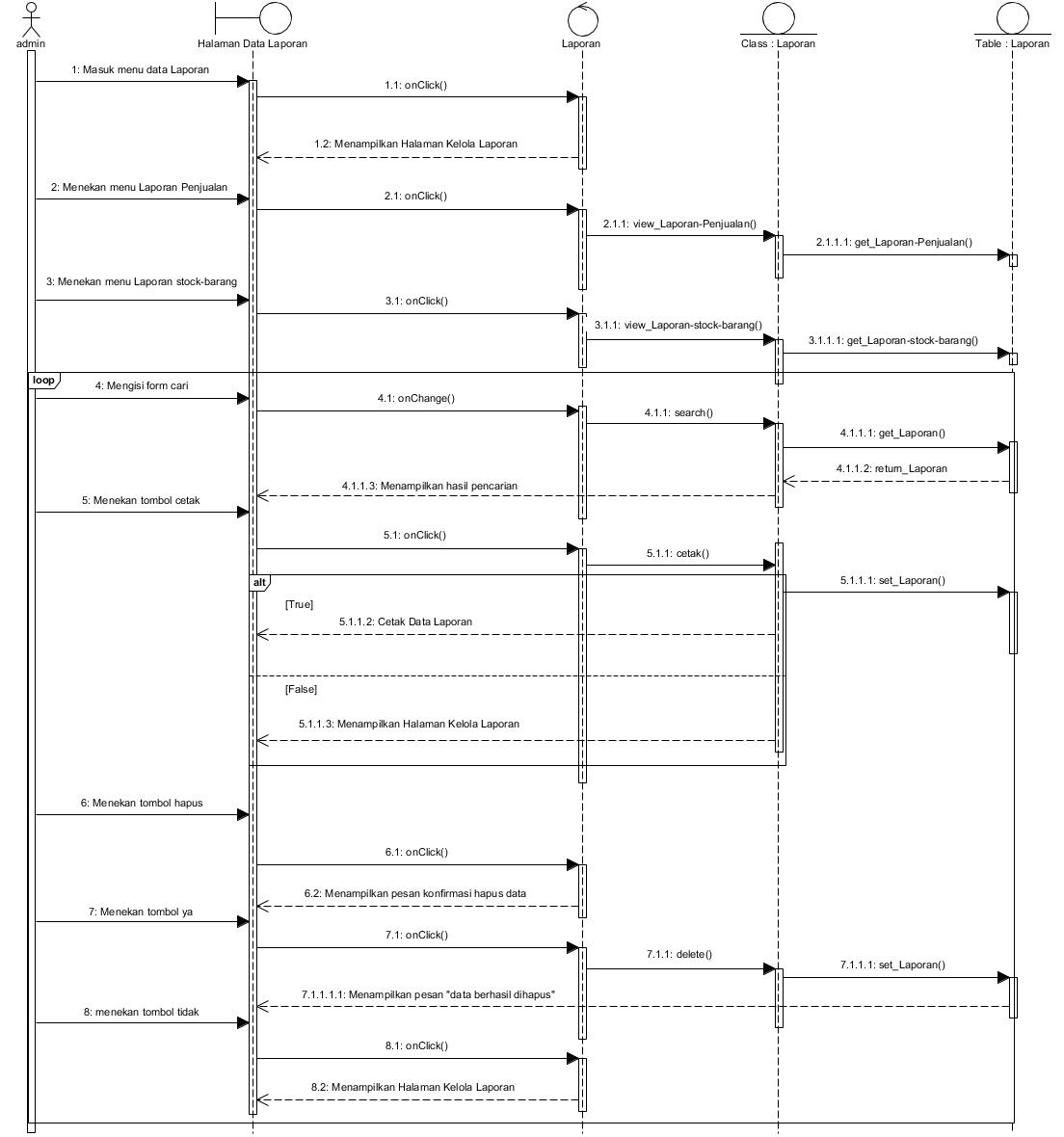
Gambar 3.14 *Sequence Diagram* Transaksi menggambarkan kasir memiliki hak akses untuk kelola transaksi, kelola transaksi memiliki menu penjualan, *stock-in*, dan *stock-out* barang, setiap menu bisa melakukan tambah, ubah, hapus yang disertakan bentuk *loop* yang berarti menunjukan proses yang sama terjadi pada setiap menu tersebut.

1. *Sequence Diagram User*

|  |
| --- |
|  |
| Gambar 3.15 *Usecase Diagram User* |

Keterangan:

Gambar 3.15 *Sequence Diagram User* menggambarkan admin memiliki hak akses untuk melakukan kelola *user* sistem meliputi tambah, ubah, hapus data *user.*

1. *Sequence Diagram* Laporan

|  |
| --- |
|  |
| Gambar 3.16 *Usecase Diagram* Laporan |

Keterangan:

Gambar 3.16 *Sequence Diagram* Laporanmenggambarkan admin memiliki hak akses untuk kelola laporan, kelola laporan memiliki menu laporan penjualan dan laporan *stock*  barang, setiap menu bisa melakukan cari, cetak, dan hapus yang disertakan bentuk *loop* yang berarti menunjukan proses yang sama terjadi pada setiap menu tersebut.

# DAFTAR PUSTAKA

<http://weekly.cnbnews.com/news/article.html?no=124000>

Oliver, J. (2019). Bab Ii Tinjauan Pustaka Aplikasi. *Hilos Tensados*, *1*, 1–476. [http://repository.potensi-utama.ac.id/jspui/bitstream/123456789/2990/6/BAB II.pdf](http://repository.potensi-utama.ac.id/jspui/bitstream/123456789/2990/6/BAB%20II.pdf) ww.yukoding.com

Belakang, L. (2005). *Bab I*. *Sasmitodia*, 1–5.

https://www.google.com/

*SISTEM INFORMASI KEPEGAWAIAN DISKOMINFO(rev1)*. (n.d.).

<https://www.wikipedia.org/>

<https://kbbi.web.id/>

García Reyes, L. E. (2013). 済無No Title No Title. Journal of Chemical Information and Modeling, 53(9), 1689–1699.