**SISTEM INFORMASI INVENTORI GUDANG BERBASIS WEBSITE MENGGUNAKAN MODEL FOUNTAIN**

**PROPOSAL SKRIPSI**

Logo

Description automatically generated

Oleh:

**MUHAMMAD EKO PUJIANTO**

06.2017.1.06756

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI ADHI TAMA SURABAYA**

**2022**

**SISTEM INFORMASI INVENTORI GUDANG BERBASIS WEBSITE MENGGUNAKAN MODEL FOUNTAIN**

**PROPOSAL SKRIPSI**

Diajukan Guna Memenuhi Persyaratan

Untuk Memperoleh Gelar Sarjana Komputer

Pada

Jurusan Teknik Informatika

Fakultas Teknik Elektro dan Teknologi Informasi

Institut Teknologi Adhi Tama Surabaya

Surabaya, Juli 2022

Mengetahui / Menyetujui,

|  |  |
| --- | --- |
| **Dosen Pembimbing** **I**  **Rahmi Rizkiana Putri, S.ST.,M.Kom**  NIP : 173133 | **Dosen Pembimbing II** |
| Jurusan Teknik Informatika  Ketua,  **Andy Rachman,S.T.,M.Kom.**  NIP : 011125 | |

# LEMBAR PENGESAHAN

**PROPOSAL SKRIPSI BERJUDUL**

**SISTEM INFORMASI INVENTORI GUDANG BERBASIS WEBSITE MENGGUNAKAN MODEL FOUNTAIN**

Yang dipersiapkan dan disusun oleh

**MUHAMMAD EKO PUJIANTO**

06.2017.1.06756

Dipertahankan di depan penguji skripsi

Jurusan Teknik Informatika

Institut Teknologi Adhi Tama Surabaya

Tanggal : …. 2022

Dinyatakan telah memenuhi syarat untuk diterima

PENGUJI

# KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan YME. Karena berkat rahmat dan karunianya, penulis dapat menyelesaikan laporan proposal skripsi yang berjudul “Sistem Informasi Inventori Gudang Berbasis Website Menggunakan Model Fountain”. Laporan Proposal ini diajukan sebagai salah satu syarat untuk menempuh skripsi Strata-1 di Jurusan Teknik Informatika – ITATS.

Dalam penulisan laporan proposal skripsi ini, telah banyak pihak yang ikut serta membantu menyelesaikannya. Baik yang memberikan bantuan dan dukungan dalam bentuk moril maupun materil. Oleh karena itu, tidak lupa penulis mengucapkan terimakasih kepada seluruh pihak yang telah membantu menyelesaikan laporan proposal skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis sepenuhnya menyadari bahwa penulisan laporan proposal skripsi ini tak luput dari kekurangan. Kritik dan saran yang membangun senantiasa penulis nantikan demi perbaikan penelitian berikutnya. Akhirnya, besar harapan penulis agar penelitian ini dapat bermanfaat bagi berbagai kalangan.

Surabaya, ….. 2022

Penulis

# DAFTAR ISI

[LEMBAR PENGESAHAN i](#_Toc109940346)

[KATA PENGANTAR ii](#_Toc109940347)

[DAFTAR ISI iii](#_Toc109940348)

[DAFTAR TABEL v](#_Toc109940349)

[DAFTAR GAMBAR vi](#_Toc109940350)

[BAB I PENDAHULUAN 1](#_Toc109940351)

[1.1 Latar Belakang 1](#_Toc109940352)

[1.2 Perumusan Masalah 4](#_Toc109940353)

[1.3 Tujuan 5](#_Toc109940354)

[1.4 Batasan Masalah 5](#_Toc109940355)

[1.5 Metodologi 5](#_Toc109940356)

[1.6 Sistem Penulisan 7](#_Toc109940357)

[BAB II LANDASAN TEORI 9](#_Toc109940358)

[2.1 Sistem Informasi 9](#_Toc109940359)

[2.2 Inventori 9](#_Toc109940360)

[2.3 Gudang 10](#_Toc109940361)

[2.4 Website 10](#_Toc109940362)

[2.5 Model Fountain 10](#_Toc109940363)

[2.6 Bahasa Pemrograman 15](#_Toc109940364)

[2.7 UML 18](#_Toc109940365)

[2.7.1 Use Case Diagram 19](#_Toc109940366)

[2.7.2 Activity Diagram 21](#_Toc109940367)

[2.7.3 Class Diagram 22](#_Toc109940368)

[2.8 Tools yang digunakan 24](#_Toc109940369)

[2.9 Skala Likert 25](#_Toc109940370)

[BAB III PERANCANGAN SISTEM 28](#_Toc109940371)

[3.1 Analysis 28](#_Toc109940372)

[3.1.1 Wawancara 28](#_Toc109940373)

[3.1.2 Observasi 30](#_Toc109940374)

[3.2 Specification Requirement 30](#_Toc109940375)

[3.2.1 User Specification Requirement 30](#_Toc109940376)

[3.2.2 Software Specification Requirements 31](#_Toc109940377)

[3.3 Design 32](#_Toc109940378)

[3.3.1 Preliminary Design 32](#_Toc109940379)

[3.3.2 Detailed Design 47](#_Toc109940380)

[DAFTAR PUSTAKA 48](#_Toc109940381)

# DAFTAR TABEL

[Tabel 1 Diagram Use Case 19](#_Toc109940212)

[Tabel 2 Activity Diagram 21](#_Toc109940213)

[Tabel 3 Skala Likert 26](#_Toc109940214)

[Tabel 4 Nilai Prosentase Skala Likert 27](#_Toc109940215)

[Tabel 5 Software Requirement 31](#_Toc109940216)

[Tabel 6 Use Case Register 35](#_Toc109940217)

[Tabel 7 Narasi Use Case untuk Mendaftarkan Staff Gudang 37](#_Toc109940218)

[Tabel 8 Narasi Use Case untuk Login 38](#_Toc109940219)

[Tabel 9 Narasi Use Case untuk Mengelola Data Master Produk 39](#_Toc109940220)

[Tabel 10 Narasi Use Case untuk Mengelola Barang Masuk 41](#_Toc109940221)

[Tabel 11 Narasi Use Case untuk Mengelola Barang Keluar 42](#_Toc109940222)

[Tabel 12 Narasi Use Case untuk Memantau Stok 44](#_Toc109940223)

[Tabel 13 Narasi Use Case untuk Menerbitkan Perintah Barang Keluar 46](#_Toc109940224)

# DAFTAR GAMBAR

[Gambar 2.5.1 Fountain Model 12](#_Toc111058643)

[Gambar 3.3.1 Alur Barang Masuk 33](#_Toc111058644)

[Gambar 3.3.2 Alur Barang Keluar Gudang 33](#_Toc111058645)

[Gambar 3.3.3 Use Case Diagram Sistem Inventori 34](#_Toc111058646)

[Gambar 3.3.4 Activity Diagram Register 48](#_Toc111058647)

# PENDAHULUAN

## Latar Belakang

Ketatnya persaingan bisnis dalam dunia usaha sudah menjadi hal yang umum bagi pengusaha. Cepat dan tepat dalam menentukan tindakan adalah suatu hal yang diutamakan. Tata kelola yang baik pada suatu manajemen diperlukan demi memperlancar kinerja perusahaan. Untuk memenuhi semua kebutuhan perusahaan tersebut maka diperlukan sistem yaitu sistem informasi (Pahlevi et al., 2018). Sistem informasi merupakan suatu sistem dalam suatu organsasi yang mengkombinasikan pengolahan transaksi, operasional, manajerial, dan startegi dari organsasi tersebut untuk mendapatkan laporan-laporan yang diperlukan (Bagir & Putro, 2018). Sistem Informasi juga merupakan perpaduan yang terorganisir antara pengguna (user), perangkat keras (hardware), perangkat lunak (software), jaringan (network), data, serta langkah atau tindakan organsasi dalam mengambil, mengubah, dan menyebarkan informasi (Pahlevi et al., 2018).

Ada banyak sistem informasi dalam suatu perusahaan, antara lain adalah Sistem Informasi Inventori Gudang. Fungsi utama dari sistem ini adalah untuk mengetahui jumlah barang pada gudang (Pahlevi et al., 2018). Keberadaan Sistem Informasi Inventori Gudang dalam suatu kegiatan usaha sudah tidak bisa dihindari lagi. Mengapa begitu? Alasan utamanya adalah barang-barang tidak dapat diperoleh secara langsung(instan), namun perlu jangka waktu yang tidak sedikit untuk memperolehnya. Jangka waktu yang dimaksud adalah waktu dimulai dari saat pemesanan, produksi, pengiriman atau distribusi, sampai dengan waktu pemrosesan barang di gudang hingga siap digunakan oleh pelanggan (Bagir & Putro, 2018). Oleh karena itu suatu perusahaan harusnya memiliki sistem informasi inventori gudang yang dapat memenuhi kebutuhan-kebutuhan penyimpanan dan pelaporan pada stok persediaan barangnya. Di samping itu, penggunaan Sistem Informasi Inventori Gudang yang baik diharapkan akan mengurangi risiko hilangnya ataupun pencurian terhadap persediaan barang.

PT. Ladang Karya Husada merupakan perusahaan asal Surabaya yang bergerak dibidang perdagangan dan pengadaan barang/jasa di ranah pemerintahan. Berdiri sejak tahun 2010, beralamat di Jl. Kutisari 2 No. 8 Surabaya. Pelanggannya kebanyakan memang dari kelembagaan dan dinas-dinas, seperti dinas pendidikan, dinas kesehatan, lembaga satuan pendidikan dan sejenisnya, namun juga membuka akses untuk kerjasama antar sesama perusahaan. Dalam bisnis perdagangan tentunya banyak kerja sama dengan perusahaan lain, principal, produsen barang serta beberapa supplier. Belum lagi pelanggan dari beberapa daerah yang berbeda. Proses bisnis yang sudah berjalan sekarang adalah PT. Ladang Karya Husada mengumpulkan semua barang dari beberapa supplier dalam satu tempat / gudang, yang nantinya barang-barang tersebut akan didistribusikan ke pelanggan.

Saat ini proses inventori yang digunakan masih manual. Dimulai dari proses entri data, transaksi, dan pelaporannya, yaitu dengan menggunakan file Microsoft Excel. Kesalahan dalam penginputan data secara manual akan menghasilkan laporan yang tidak akurat. Hal ini dapat menimbulkan kesalahan koordinasi dan komunikasi antar unit berkaitan, sehingga bisnis perusahaan tidak berjalan efektif. Dengan adanya barang yang keluar – masuk pada gudang maka dibutuhkan suatu sistem atau teknologi yang dapat mengakomodir proses pendataan barang tersebut. Bentuk teknologi yang dapat dimanfaatkan untuk memenuhi kebutuhan tersebut yaitu Sistem Informasi Inventori Gudang. Dimana pengguna dapat mendata barang keluar-masuk dengan cepat, tepat, dan akurat.

Berdasarkan masalah tersebut dan kegunaan Sistem Informasi Inventori Gudang sangatl berkaitan. Keterkaitan yang dimaksud adalah dengan adanya Sistem ini maka permasalahan-permasalahan yang timbul selama ini dapat teratasi. Dengan adanya Sistem Informasi Inventori Gudang ini diharapkan PT Ladang Karya Husada dapat mengelola informasi gudang dengan lebih baik dan tertata.

Proses pengembangan Sistem Inventori Gudang ini akan menggunakan model SDLC (Systems Development Life Cycle/Siklus Hidup Pengembangan Sistem) Fountain. Model Fountain merupakan penyempurnaan logic dari model waterfall dengan urutan yang masih sama (Herdiana & Sudarma, 2021). Dalam Model Fountain proses pengembangan sistem dapat dilakukan secara tumpang tindih dan tidak tarpaku pada satu proses saja (Ernawati & Wati, 2021). Sehingga bisa mendahulukan langkah yang lain yang sudah siap untuk dilanjutkan ke proses selanjutnya, disamping itu juga bisa melanjutkan langkah sebelumnya ke tahapan selanjutnya (Aryadi et al., 2020). Namun, ada satu tahapan atau proses yang tidak bisa dilewati yaitu tahapan design yang diperlukan dahulu sebelum tahapan pengkodean (Herdiana & Sudarma, 2021).

Keunggulan model Fountain yaitu cocok untuk pengembangan perangkat lunak berorientasi objek. Model Fountain memiliki keuntungan bahwa perangkat lunak progresif diperbolehkan untuk ditambahkan pada setiap pengembangan berulang, dan lebih mudah untuk melengkapi yang hilang dalam fase pengembangan lainnya kapan saja selama satu fase pengembangan (Yu, 2018). Model ini dapat mendahulukan tahapan yang sudah siap untuk dikerjakan kemudian melanjutkan tahapan sebelumnya atau tahapan selanjutnya. Oleh karena itulah penulis memutuskan menggunakan model Fountain, karena dalam pengembangan aplikasi terhadap tahapan yang sering terjadi perulangan dan tumpang tindih fase pengembangan.

Sistem Informasi Inventori Gudang merupakan salah satu sistem yang dapat memenuhi kebutuhan dari PT. Ladang Karya Husada. Dengan adanya Sistem Informasi Inventori Gudang diharapkan dapat mempermudah proses bisnis dari PT. Ladang Karya Husada. Sehingga penelitian ini mengambil judul “Sistem Informasi Inventori Gudang Berbasis Website Menggunakan Model Fountain”

## Perumusan Masalah

Ditinjau dari latar belakang tersebut, dapat disimpulkan beberapa permasalahan yaitu:

1. Bagaimana merancang dan membangun Sistem Informasi Inventori Gudang di PT. Ladang Karya Husada?
2. Bagaimana menerapkan Fountain model pada pengembangan Sistem Informasi Inventori Gudang berbasis website di PT. Ladang Karya Husada?

## Tujuan

Tujuan penelitian yang didapatkan dari rumusan masalah yang ada, yaitu:

1. Proses pendataan barang keluar dan barang masuk lebih tertata.
2. Menyediakan informasi yang cepat, tepat dan akurat mengenai data maupun laporan.

## Batasan Masalah

Untuk menghindari penyusunan penelitian ini tidak keluar dari inti permasalahan yang telah dirumuskan, maka dibatasi pada :

1. Penelitian dilakukan di PT Ladang Karya Husada.
2. Sistem dibangun berbasis website dengan bahasa pemrograman PHP framework Laravel, dan Database PostgreSQL.
3. Sistem Informasi Inventori Gudang dapat mendata barang keluar-masuk di Gudang perusahaan.

## Metodologi

Dalam pembuatan skripsi, menggunakan metode penelitian dengan langkah sebagai berikut:

1. Studi Literatur

Mempelajari informasi dan literatur yang memiliki keterkaitan dengan rumusan masalah pada penelitian ini. Informasi yang akan digunakan bisa didapatkan melalui: internet, jurnal, e-book ataupun buku acuan.

1. Observasi

Untuk mendapatkan informasi tentang permasalahan yang ada dan untuk mengumpulkan data-data yang diperlukan dalam penelitian ini, maka akan dilakukan peninjauan langsung ke lokasi.

1. Analisis Permasalahan

Untuk mendapatkan batasan-batasan dari sistem yang akan dibangun, sehingga penulis dapat menentukan cara yang terbaik untuk menyelesaikan permasalahan yang ada.

1. Membangun Aplikasi

Setelah mendapatkan seluruh permasalahan, maka dapat langsung menuju langkah selanjutnya, yaitu membangun aplikasi sebagai solusi dari permasalahan yang ditemukan.

1. Implementasi Sistem

Sistem yang berhasil dibangun dapat diterapkan pada studi kasus, dimana bisa didapatkan hasil bahwa sistem dapat dioperasikan sesuai tujuan dibangunnya sistem tersebut.

1. Uji Coba Sistem

Melakukan pengujian pada sistem untuk menemukan kesalahan/bug yang masih ada, dan langsung melakukan pembenahan pada bug tersebut.

1. Dokumentasi
2. Menyusun laporan sebagai dokumentasi dari pelaksanaan, penelitian, dan pembangunan sistem tersebut.

## Sistem Penulisan

**BAB I : PENDAHULUAN**

Pada bab ini akan dijelaskan latar belakang penelitian, perumusan, masalah, batasan masalah, tujuan penelitian, dan metodologinya serta sistematika penulisan

**BAB II : LANDASAN TEORI**

Pada bab ini akan dijelaskan teori yang digunakan untuk menyelesaikan permasalahan dari penelitian yang akan diteliti, selain itu juga akan dijelaskan langkah-langkahnya.

**BAB III : PERANCANGAN SISTEM**

Pada bab ini akan dijelaskan proses perancangan yang dilakukan untuk menyelesaikan permasalahan pada penelitian. Hal ini meliputi metode penelitian yang digunakan, subjek penelitian dan perangkat apa saja yang digunakan pada penelitian.

**BAB IV : IMPLEMENTASI SISTEM**

Pada bab ini akan dijelaskan tentang penerapan rancangan pada penelitian pada sistem yang dibangun dan sudah beroperasi.

**BAB V : PENUTUP**

Pada bab ini akan memuat kesimpulan dan juga saran yang dapat membantu penelitian dalam mengembangkan sistem aplikasi untuk kedepannya.

# LANDASAN TEORI

## Sistem Informasi

Sistem Informasi adalah suatu sistem yang berada di dalam suatu organisasi yang mampu mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, dan kegiatan strategi dari suatu organisasi tersebut serta menyediakan laporan-laporan yang diperlukan.(Ernawati et al., 2021)

Sistem informasi, pada dasarnya sistem adalah kumpulan dari unsur-unsur yang berkaitan erat satu sama lain, yang berfungsi bersama untuk mencapai tujuan tertentu. (Efendi & Krisanty, 2020). Sistem Informasi terdiri dari beberapa elemen yaitu: orang, prosedur, perangkat keras, perangkat lunak, basis data, jaringan komputer dan komunikasi data. Semua elemen tersebit merupakan komponen fisik.(Bagir & Putro, 2018).

## Inventori

Inventori (Persediaan) adalah segala sesuatu atau sumber daya organisasi yang disimpan sebagai antisipasinya terhadap pemenuhan permintaan. Permintaan tersebut meliputi bahan mentah, barang dalam proses, barang jadi, ataupun produk final (produk jadi).(Jaharuddin et al., 2020)

Inventori (persediaan) adalah stok bahan yang digunakan untuk memudahkan produksi atau memuaskan pelanggan. Persediaan merupakan suatu aktiva yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam suatu periode usaha tertentu, atau persediaan barang-barang yang masih dalam pengerjaan atau proses produksi, ataupun persediaan bahan baku yang menunggu penggunaannya dalam suatu produksi.

## Gudang

Gudang dapat didefinisikan sebagai tempat yang dibebani tugas untuk menyimpan barang yang akan dipergunakan dalam produksi sampai barang diminta sesuai dengan jadwal produksi. Sejak dulu, gudang berfungsi sebagai buffer atau penyeimbang dan untuk menentukan langkah selanjutnya suatu perusahaan, apakah perusahaan akan menggunakan gudang untuk komersial atau lebih baik digunakan sendiri.(Fitri & Irsya Putri2, 2021)

## Website

Website dapat diartikan sebagai halaman-halaman yang berfungsi untuk menampilkan informasi. Berisikan teks, gambar, animasi, suara, ataupun gabungan dari semuanya. Membentuk satu rangkaian bangunan yang saling terkait, dan terhubung melalui jaringan-jaringan halaman. Perkembangan dunia website pada saat ini lebih menekankan pada pengelolaan content adalah sebuah website. Pengguna yang tidak bisa bahasa pemrograman website pada saat ini bisa membuat website dengan memanfaatkan CMS tersebut. (Elektro & Medan, 2012).

## Model Fountain

Model Fountain diusulkan sebagai cara untuk mewakili siklus hidup pengembangan perangkat lunak yang sangat berulang. Meskipun sebagian besar proses memiliki aliran sekuensial alami dalam siklus hidup pengembangannya, perangkat lunak adalah satu area di mana pengkodean dapat dimulai bahkan sebelum desain benar-benar matang. Ini karena inisiasi dan modifikasi tambahan untuk implementasi selama fase desain biasanya tidak terlalu mahal. Selanjutnya, programmer ahli memiliki kecenderungan untuk mengekspresikan desain mereka dalam bahasa pemrograman, bukan dalam bahasa alami. Dengan demikian, desain dan implementasi, tidak lagi terpisah dalam pengertian tradisional dan bahasa teknologi alat CASE saat ini, membantu mempercepat tumpang tindih antara fase analisis, desain, dan implementasi. Model Fountain merupakan penyempurnaan logis dari model waterfall yang langkah dan urutannya masih sama dengan model waterfall. Dalam model Fountain, suatu proses dapat didahulukan dalam merancang dan mengembangkan suatu sistem, namun ada urutan atau proses yang tidak dapat dilewati seperti melalui tahap desain proses hingga tahap pengkodean, tahap tersebut tidak dapat dilewati karena desain diperlukan terlebih dahulu sebelum proses ke tahap coding, jika di skip maka akan terjadi overlap pada siklus SDLC.(Herdiana & Sudarma, 2021). Ilustrasi Model Fountain dapat dilihat pada Gambar 2.1

Diagram

Description automatically generated

Gambar 2.5.1 Fountain Model

Metode fountain adalah perbaikan dari metode waterfall, di mana jenis tahapan masih sama. Namun beberapa jenis tahapan boleh didahulukan atau dilewati, tetapi ada tahapan yang tidak bisa dilewati, contohnya seperti kamu memerlukan design sebelum melakukan implementasi, jika hal tersebut dilewati maka akan ada tumpang tindih.(Setiawan, 2021)

Berikut adalah tahap-tahap pengembangan perangkat lunak menggunakan metode fountain:

1. Analysis

Menganalisis kebutuhan pengguna baik dengan cara yang mudah dan efisien penggunaan sistem yang dibutuhkan oleh pengguna.(Herdiana & Sudarma, 2021)

1. Specification Requirement
   1. User Specification Requirement

Peningkatan kebutuhan pengguna diupayakan secara khusus, untuk mengetahui kebutuhan secara detail dari pengguna pada sistem yang akan dikembangkan.(Herdiana & Sudarma, 2021)

* 1. Software Specification Requirements

Menganalisa kebutuhan yang meliputi ketersediaan perangkat lunak dari aplikasi yang akan dibangun.

1. Design
   1. Preliminary Design

Desain awal atau desain konseptual dari sistem yang dikembangkan dengan mempertimbangkan kebutuhan perangkat lunak dan perangkat keras sistem. (Herdiana & Sudarma, 2021).

Tahapan ini melakukan desain sistem sesuai dengan aplikasi yang akan dikembangkan. Sebelum membuat desain sistem terlebih dahulu peneliti membuat alur kerja dari aplikasi yang akan dikembangkan. Kemudian merancang desain sistem menggunakan UML dengan membuat usecase diagram dan activity diagram.(Ernawati et al., 2021).

* 1. Detailed Design

Membuat desain yang mendekati fisik sistem atau desain secara detail dari setiap bagian desain sistem yang dikembangkan. (Herdiana & Sudarma, 2021).

Sebelum memulai implementasi kode penulis menggunakan mockup untuk melihat gambaran aplikasi yang akan dibangun.(Ernawati et al., 2021).

1. Coding

Setelah tahapan physical design dilakukan, maka tahap selanjutnya adalah implementation/coding.(Ernawati et al., 2021). Dalam pengembangan aplikasi ini menggunakan framework Laravel dengan bahasa pemrograman PHP.

1. Sistem Testing

Pengujian sistem memastikan bahwa setiap bagian dalam desain atau pengembangan memenuhi persyaratan. Pengulangan mungkin diperlukan antara fase ini dan fase sebelumnya tergantung pada kelengkapan desain yang direncanakan.(Herdiana & Sudarma, 2021)

1. System Release

Pengguna dapat menguji sistem yang telah dihasilkan dan dapat diajarkan untuk menggunakan sistem yang telah dirancang dan dikembangkan. Umpan balik yang dihasilkan pengguna akan meningkatkan sistem yang dirancang karena melihat kembali melalui tahap sebelumnya.(Herdiana & Sudarma, 2021)

1. Maintenance and Evolution

Sistem yang dirancang dan dikembangkan perlu dilakukan perbaikan untuk menghindari beberapa cacat yang terjadi pada saat sistem dijalankan dan melakukan update untuk menghindari kesalahan pada sistem. Umpan balik dari pengguna diperlukan untuk melihat bagian-bagian yang harus diperbaiki dari sistem yang telah dibuat.(Herdiana & Sudarma, 2021)

## Bahasa Pemrograman

Bahasa Pemrograman adalah sebuah instruksi standar untuk memerintah komputer agar menjalankan fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

Berikut ini Penulis akan membahas mengenai bahasa pemrograman yang akan digunakan dalam merancang dan membuat Rancang Bangun Sistem Informasi Inventori Gudang di PT. Ladang Karya Husada berbasis website sebagaimana yang merupakan terpenting dalam membangun aplikasi, antara lain:

1. PHP

PHP Hypertext Preprocessor atau disingkat dengan PHP ini adalah suatu bahasa scripting khususnya yang digunakan untuk pengembangan web. Karena sifatnya yang server side scripting, maka untuk menjalankan PHP harus menggunakan web server. PHP juga dapat diintegrasikan dengan HTML, JavaScript, JQuery, Ajax. Namun, pada umumnya PHP lebih banyak digunakan secara bersamaan dengan file bertipe HTML.(Efendi & Krisanty, 2020).

1. Database

Database atau basis data adalah kumpulan data yang dikelola sedemikian rupa berdasarkan ketentuan tertentu yang saling berhubungan sehingga mudah dalam pengelolaannya. Melalui pengelolaan tersebut pengguna dapat memperoleh kemudahan dalam mencari informasi, menyimpan informasi dan membuang informasi. (Intern, 2020). Kumpulan file/table/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis. (Bagir & Putro, 2018).

1. PostgreSQL

PostgreSQL merupakan salah satu Object Relational Database Management System (ORDBMS) yang bersifat open source, yang berarti bahwa source code dari PostgreSQL dapat digunakan secara bebas. PostgreSQL mendukung Structured Query Language (SQL) yang memiliki kemampuan transactions, subqueries, triggers, dan lain-lain.(Ardian Dwi Praba, 2020)

1. HTML

HTML(Hyper Text Mark Up Language) merupakan bahasa yang digunakanuntuk mendeskripsikan struktur sebuah halaman web. HTML berfungsi untuk mempublikasi dokumen online. Statement dasar dari HTML disebut tags. Sebuah tag dinyatakan dalam sebuah kurung siku(<>). Tags yang ditujukan untuk sebuah dokumen atau bagian dari suatu dokumen haruslah dibuat berupa pasangan. Terdiri dari tag pembuka dan tag penutup. Dimana tag penutup menggunakan tambahan tanda garis miring(/) di awal nama tag.(Pahlevi et al., 2018)

1. CSS

Cascading style sheets (CSS) adalah file eksternal (mirip dengan format katalog) yang berisi pasangan label-nilai yang dapat digunakan untuk menggambarkan berbagai aspek keluaran, seperti font, warna (latar depan, latar belakang), dan perbatasan. Awalnya, CSS muncul sebagai cara untuk menerapkan informasi gaya ke output yang disajikan di W3, yaitu, HTML. Sebelum implementasi CSS, jenis informasi gaya (dan banyak lagi) dijelaskan sebelumnya disematkan bersama dengan konten yang disajikan.(Hadden & Hughes, 2019).

1. Javascript

Javascript merupakan suatu bahasa pemrograman yang paling banyak digunakan. Javascript ini menggunakan bahasa dengan jenis cripting yang di mana dapat digunakan di HTML, web, server, laptop, tablet, ponsel dan harus memasukkan kodenya secara lansung untuk dieksekusi, dan tidak dapat dikompilasi dengan file executable. Javascript juga bahasa pemrograman yang digunakan untuk membuat suatu program yang dapat digunakan agar dokumen HTML yang ditampilkan dalamb rowser menjadi lebih interaktif, tidak sekedar indah saja.(Wongso, 2020).

1. Laravel

Laravel merupakan web application framework berbasis PHP yang open source, menggunakan konsep Model–View–Controller (MVC). Laravel berada dibawah lisensi MIT License, dengan menggunakan GitHub sebagai tempat berbagi kode. Keunggulan Laravel dibandingkan dengan framework lain yaitu sebagai berikut:

1. Laravel merupakan framework PHP yang ekspresif, artinya sintaks pada Laravel menggunakan bahasa yang mudah dimengerti programmer pemula.
2. Laravel memiliki dokumentasi yang cukup lengkap, setiap versinya memiliki dokumentasi tersendiri mulai dari cara instalasi hingga penggunaan fiturnya.
3. Laravel didukung oleh Composer sehingga library-library Laravel dapat diperoleh dengan mudah dari internet menggunakan Composer. Composer sendiri merupakan Dependency Management PHP yang membantu kita mendapatkan library yang kita pakai dan menginstalnya dari internet.
4. Laravel memiliki template engine sendiri yang diberi nama blade yang memudahkan kita menampilkan data pada template HTML.

## UML

UML adalah bahasa yang digunakan untuk memvisualisasikan, membangun, dan mendokumentasikan item informasi. UML menawarkan standar untuk pemodelan sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan pada perangkat keras apapun, sistem operasi dan jaringan apapun, dan ditulis dalam bahasa pemrograman apapun (Kambivi et al., 2020).

Unified Modeling Language (UML) juga merupakan bahasa standar untuk merancang denah pengembangan suatu perangkat lunak. Contoh seperti arsitek bangunan perlu merancang gambar denah untuk sebuah perancangan pembangunan konstruksi, arsitek software juga perlu merancang diagram menggunakan UML untuk perancangan suatu pengembangan perangkat lunak (Pahlevi et al., 2018).

### **Use Case Diagram**

Diagram use case merupakan gambaran atau representasi dari interaksi yang terjadi antara sistem dan lingkungannya. Use case diagram bertujuan untuk menggambarkan/menjelaskan perilaku perangkat lunak yang akan dikembangkan. Konsep use case yaitu mendeskripsikan cara interaksi antara sistem dengan penggunanya melalui skenario cerita/rancangan tentang bagaimana sistem itu akan berjalan dan digunakan. Setiap skenario akan menjelaskan urutan kejadian sejak fungsi sebuah sistem digunakan hingga fungsi tersebut telah berhasil dieksekusi (Putra & Andriani, 2019). Diagram Use Case dapat dilihat pada table 1 dibawah ini:

Tabel 1 Diagram Use Case

|  |  |  |
| --- | --- | --- |
| **No.** | **Simbol** | **Keterangan** |
| 1 |  | Gambar yang mewakili pengguna aplikasi yang akan berinteraksi dengan sistem, dimana pengguna ini hanya akan menjalankan perintah dan menerima informasi dari sistem. |
| 2 |  | Gambar dari sebuah fungsi dari suatu aktivitas pada sistem yang akan dibangun. |
|  |  | Gambar yang menjelaskan bahwa ada suatu aktivitas yang harus terpenuhi terlebih dahulu agar sebuah event dapat dijalankan. |
|  |  | Gambar yang menjelaskan bahwa sebuah event yang dapat dijalnkan tanpa harus melewati aktivitas tertentu. |
|  | Association | Gambar yang menjelaskan bahwa adanya hubungan antar fungsi dengan fungsi lainnya, atau antara actor dengan fungsi yang ada. |

### **Activity Diagram**

Activity diagram merupakan diagram yang menggambarkkan workflow atau aktivitas dari sebuah sistem yang ada pada perangkat lunak (Putra & Andriani, 2019). Diagram Activity dapat dilihat pada tabel 3 di bawah ini:

Tabel 2   
Activity Diagram

|  |  |  |
| --- | --- | --- |
| **No.** | **Simbol** | **Keterangan** |
| 1 | Start | Menggambarkan bahwa simbol ini adalah awal dimulainya aktivitas yang akan dilakukan. |
| 2 | Control Flow | Menunjukkan urutan eksekusi. |
| 3 | Proses / Kegiatan | Menggambarkan suatu aktivitas yang akan dilakukan/terjadii pada sistem. |
| 4 | Decisions | Menggambarkan suatu keputusan dari aliran suatu aktivitas, Decisions dapat mengalir ke lebih dari satu jalur sesuai dengan yang diinginkan. |
| 5 | Final Activity | Menggambarkan bahwa simbol ini adalah berakhirnya suatu aktivitas yang dilakukan. |

### **Class Diagram**

Class Diagram adalah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Kelas menggambarkan keadaan (atribut/properti) sistem, sambil menawarkan layanan untuk memanipulasi situasi (metode/fungsi). Diagram kelas menggambarkan struktur dan deskripsi kelas, objek dan paketnya serta hubungan satu sama lain seperti penahanan, pewarisan, asosiasi, dan lain-lain. Class Diagram memiliki tiga area utama, yaitu nama, atribut, dan metode (Kambivi et al., 2020).

Atribut dan metode dapat memiliki salah satu properti dari berikut:

1. Private, tidak bisa dipanggil dari luar kelas yang bersangkutan.
2. Protected, Hanya dapat dipanggil oleh kelas terkait dan anak-anak yang mewarisinya.
3. Publik, Bisa ditelepon oleh siapa saja.

Class Diagram dapat dilihat pada tabel 4 berikut ini:

|  |  |  |
| --- | --- | --- |
| **No.** | **Simbol** | **Keterangan** |
| 1 | Class | Menggambarkan sebuah kelas pada struktur sistem. |
| 2 | Association | Menggambarkan adanya hubungan antara objek. |
| 3 | Agregations | Agregasi adalah hubungan antara dua class di mana salah satu class merupakan bagian dari class lain, tetapi dua class ini dapat berdiri masing-masing |
| 4 | Dependency | Menggambarkan adanya hubungan antara kelas, namun memiliki arti bahwa kelas tersebut memiliki ketergantungan dengan kelas yang lainnya. |
| 5 | Inheritance | Pewarisan atau inheritance dapat disebut juga generalization dalam class diagram adalah suatu kemampuan untuk mewarisi seluruh atribut dan metode dari class asalnya (superclass) ke class lain (subclass) |

## Tools yang digunakan

Dalam proses pengembangan sistem informasi ini, ada beberapa tools pengembang yang perlu dipersiapkan antara lain:

1. Visual Studio Code

Visual Studio Code adalah editor kode sumber gratis. Ini memiliki dukungan bawaan untuk bahasa pemrograman JavaScript dan TypeScript dan selanjutnya memungkinkan dukungan untuk bahasa lain melalui ekosistem ekstensi yang kaya. Visual Studio Code menggunakan folder atau sistem ruang kerja untuk berinteraksi dengan proyek dan sistem dokumen untuk menangani file kode sumber dalam proyek. Ini memungkinkan Kode VS menjadi agnostik bahasa dan mendelegasikan fungsionalitas khusus bahasa ke ekstensi (Fitzgerald et al., 2021).

1. Laragon

Laragon adalah lingkungan pengembangan universal yang portabel, terisolasi, cepat & kuat untuk PHP, Node.js, Python, Java, Go, Ruby. Ini cepat, ringan, mudah digunakan dan mudah diperpanjang. Laragon sangat bagus untuk membangun dan mengelola aplikasi web modern. Hal ini difokuskan pada kinerja - dirancang di sekitar stabilitas, kesederhanaan, fleksibilitas dan kebebasan. Laragon sangat ringan dan akan tetap ramping mungkin. Biner inti itu sendiri kurang dari 2MB dan menggunakan RAM kurang dari 4MB saat dijalankan. Laragon tidak menggunakan layanan Windows. Ini memiliki orkestrasi layanan sendiri yang mengelola layanan secara asinkron dan non-pemblokiran sehingga Anda akan menemukan segala sesuatunya berjalan cepat & lancar dengan Laragon .

## Skala Likert

Skala Likert adalah metode yang digunakan untuk mengukur tingkat kepuasan pengguna. Skala likert merupakan skala yang digunakan untuk mengukur suatu pendapat atau kepuasan pada seseorang atau kelompok tertentu mengenai hasil penelitian berdasarkan definisi yang telah ditetapkan oleh peneliti (Rachman et al., 2020). Kata Likert diambil dari pengembangnya yaitu *Rensis Likert* yang dikembangkan pada Tahun 1932. Skala likert memiliki empat atau lebih butir-butir pertanyaan yang dikombinasikan sehingga membentuk sebuah skor atau nilai yang mempresentasikan sifat individu, misalkan pengetahuan, sikap dan perilaku. Skala likert dapat juga dikatakan sebagai skala psikometrik yang umum digunakan dalam kuesioner dan merupakan skala yang paling banyak digunakan untuk penelitian (Setyawan & Atapukan, 2018).. Contoh Skala Likert dapat dilihat pada tabel 4 berikut:

Tabel 3   
Skala Likert

|  |  |
| --- | --- |
| **Skala** | **Kategori** |
| 1 | Tidak Setuju |
| 2 | Kurang Setuju |
| 3 | Cukup |
| 4 | Setuju |
| 5 | Sangat Setuju |

Untuk menetapkan peringkat dalam setiap skor dapat dilihat dari perbandingan antara skor aktual dan ideal. Skor aktual merupakan skor yang diperoleh melalui hasil penghitungan seluruh jawaban respoden, sedangkan skor ideal merupakan skor dari prediksi nilai yang tertinggi dikali jumlah pertanyaan dikali jumlah responden. Jadi skor aktual adalah jawaban seluruh responden terhadap pertanyaan dari kuesioner, sedangkan skor ideal adalah bobot tertinggi atau semua responden yang sudah memilih jawaban dari kuesioner. Berikut adalah nilai prosentase yang didapat dari skor ideal yang sudah dikali dengan skor bobot tabel 5

Tabel 4 Nilai Prosentase Skala Likert

|  |  |
| --- | --- |
| **Prosesntase(%)** | **Keterangan** |
| 0% - 19,99% | Tidak Setuju |
| 20% - 39,99% | Kurang Setuju |
| 40% - 59,99% | Cukup |
| 60% - 79,99% | Setuju |
| 80% - 100% | Sangat Setuju |

Nilai yang didapat dari tabel 5 merupakan prosentase nilai yang didapat dari skor ideal dikali dengan 100%, apabila mendapatkan nilai 80% artinya skor ideal yang didapat sangat baik sesuai kuesioner.

# PERANCANGAN SISTEM

Pada bab perancangan sistem ini sesuai yang sudah dijelaskan pada bab sebelumnya yaitu menggunakan model Fountain seperti pada gambar 2.5.1. Adapun tahapannya yaitu Analysis, Specification Requirement, Design, Coding, System Testing, System Release, Maintenance & Evolution. Yang mana masing – masing tahapan akan dijelaskan pada sub bab dan sub-sub bab di bawah ini.

## Analysis

Pada pengembangan perangkat lunak model Fountain, tahapan pertama yaitu Analysis dimana pada tahapan ini adalah proses untuk menganalisa kebutuhan pengguna sesuai yang terlihat pada gambar 2.5.1. Proses analisa terdiri dari dua aktivitas, yaitu wawancara dan observasi kepada pihak PT. Ladang Karya Husada.

### **Wawancara**

Penulis melakukan wawancara langsung pada Kepala Gudang dan Staff di PT Ladang Karya Husada. Wawancara ini bertujuan untuk mengumpulkan informasi tentang bagaimana proses inventori yang sudah berjalan, cara pengolahan data, dan fitur aplikasi yang dapat mempermudah kinerja tim Gudang. Berikut hasil wawancara yang dilakukan kepada Kepala Gudang:

1. Proses input data barang keluar masuk masih menggunakan file Microsoft Excel.
2. Untuk rekap data masih diperlukan banyak file beberapa versi.
3. Dalam satu gudang ini terdapat barang-barang dari beberapa perusahaan (Ladang Group).
4. Sering terjadi miskomunikasi tentang data stok, karena update data harus kirim file baru lagi.
5. Pelaporan data inventori tidak bisa realtime.

Wawancara yang dilakukan kepada Staff Gudang dapat beberapa poin informasi sebagai berikut

1. Menggunakan file Microsoft Excel sangat riskan hilang atau kerusakan file apalagi jika terjadi mati lampu dadakan, terkadang komputer tidak menyimpan recovery filenya.
2. Sering terjadi kesalahan input data, double data dan tidak sinkronnya data.
3. Perlu file tambahan khusus untuk data serial number serta riwayatnya.
4. Memerlukan idntifikasi khusus untuk perusahaan sebagai pemilik barang.
5. Sering ditemukan serial number yang duplikat data.

Dari wawancara ini akhirnya mendapatkan beberapa poin penting. Untuk pendataan barang keluar-masuk masih menggunakan Microsoft Excel dan harus dituliskan manual setiap ada produk yang ditransaksikan. Dokumen yang sudah dibuat kemudian disimpan pada penyimpan lokal komputer. Ketika data ingin diakses oleh staff gudang yang lain, harus lakukan penyalinan data. Tentu saja hal ini kurang efisien dan aman. Dikawatirkan dokumen yang disimpan pada penyimpanan lokal komputer dapat hilang saat perangkat keras penyimpanan rusak.

### **Observasi**

Setelah melalui proses wawancara penulis mencoba mengamati dengan terjun langsung di Gudang. Pengamatan ini bertujuan untuk melihat seperti apa detail dari sistem yang sudah berjalan selama ini. Dengan penulis dapat lebih jelas mengetahui permasalahan dan kendala yang dihadapi.

## Specification Requirement

### **User Specification Requirement**

Pada tahapan ini penulis akan menjabarkan kebutuhan pengguna secara spesifik agar proses pengembangan sistem ini sesuai dengan alur yang diinginkan oleh penggunanya. Kebutuhan ini dibagi menjadi dua yaitu, kebutuhan fungsional dan kebutuhan nonfungsional.

1. **Kebutuhan Fungsional**

Dalam pengembangan aplikasi ini kebutuhan fungsional dapat dijabarkan sebagai berikut:

1. Dapat melakukan login user,
2. Dapat mengelola data user.
3. Dapat mengelola kategori produk.
4. Dapat mengelola supplier / partner.
5. Dapat mengelola perusahaan pemilik barang.
6. Dapat mengelola data produk / barang.
7. Dapat Menampilkan data detail produk / barang.
8. Dapat mengelola transaksi (barang keluar-masuk).
9. Dapat Menampilkan data detail transaksi (barang keluar-masuk).
10. Dapat mengelola Serial Number produk.
11. **Kebutuhan Non Fungsional**

Dalam pengembangan aplikasi ini kebutuhan non fungsional dapat dijabarkan sebagai berikut:

* 1. Aplikasi dapat diakses user melalui browser yang mendukung javascript.
  2. Design aplikasi yang user friendly agar user merasa mudah dan nyaman menggunakannya.
  3. Keamanan dengan mewajibkan user login untuk mengakses fitur-fitur utama aplikasi.

### **Software Specification Requirements**

Pada tahapan ini penulis akan menjabarkan kebutuhan spesifik tentang software yang akan digunakan untuk proses pengembangan Sitem Informasi Inventori Gudang ini. Kebutuhan software tersebut dapat dilihat pada tabel berikut:

Tabel 5 Software Requirement

|  |  |
| --- | --- |
| **Jenis** | **Komponen** |
| Sistem Operasi | Windows 11 Pro (64 bit) |
| Browser | Google Chrome |
| Kode Editor | Visual Studio Code |
| Bahasa Pemrograman | Laravel (PHP), jQuery, Node.js, AdminLTE(Bootstrap) |
| Local Server Environment | Laragon |

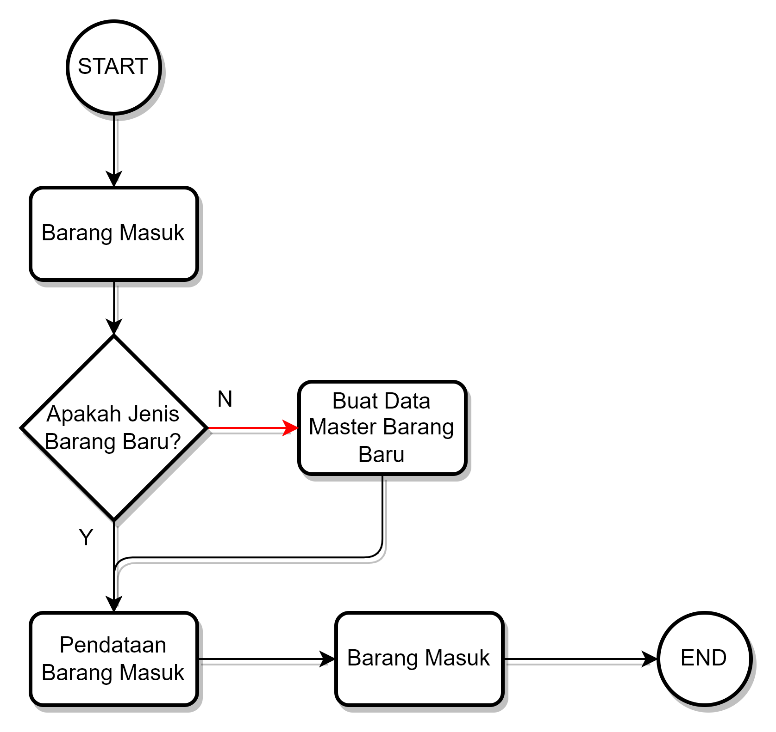
## Design

### **Preliminary Design**

Perancangan sistem sangat dibutuhkan sebelum membuat suatu sistem aplikasi. Rancangan tersebut meliputi perancangan input dan output. Untuk memahami dan merealisasikan sistem, Diperlukan suatu gambaran mengenai sistem dan alur data yang terjadi.

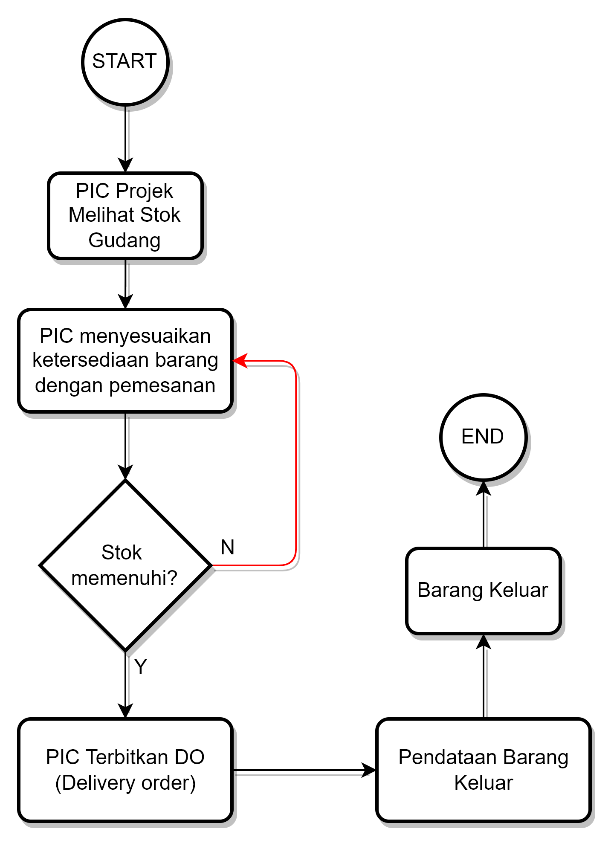
1. **Alur Kerja Sistem Informasi Inventori Gudang**

Alur kerja dibagi menjadi dua, yaitu alur kerja barang masuk dan alur kerja barang keluar. Untuk alur kerja barang masuk bisa dilihat pada gambar 3.3.1:



Gambar 3.3.1 Alur Barang Masuk

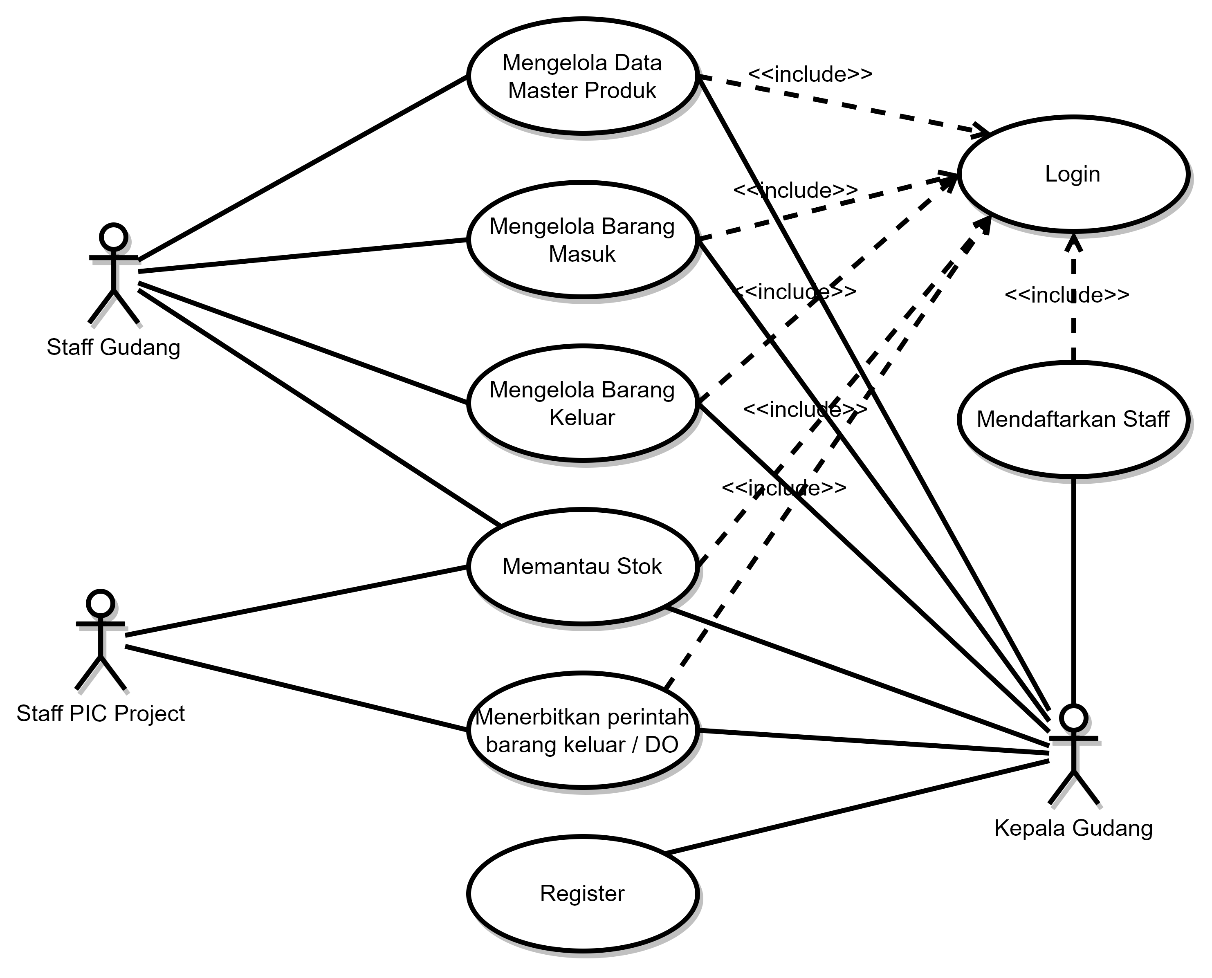
Untuk alur barang keluar bisa dilihat pada gambar 3.3.2:



Gambar 3.3.2 Alur Barang Keluar Gudang

1. **Use Case Diagram**

Use Case Diagram menggambarkan requirement fungsional dari suatu proses dalam sistem inventori. Use Case Diagram tidak menjelaskan secara rinci, hanya berupa gambaran global yang menunjukan hubungan antara aktor dengan sistem. Berikut ini adalah rancangan Use Case Diagram dari sistem yang akan dikembangkan:



Gambar 3.3.3 Use Case Diagram Sistem Inventori

Pada gambar 3.3.3 di atas terlihat ada 3 aktor yang akan menggunakan sistem ini, yaitu: Kepala Gudang, Staff Gudang, Staff PIC Projek. Kepala Gudang merupakan aktor yang mendaftarkan perusahaan yang dipegang pada sistem inventori. Kepala Gudang mempunyai hak akses khusus yaitu fitur Mendaftarkan Staff Gudang. Kepala Gudang juga memiliki akses penuh untuk semua fitur pada sistem inventori ini. Staf Gudang adalah user yang sudah di daftarkan oleh kepala gudang, dimana memiliki fitur yang sudah dikhususkan untuk mengelola inventori gudang. Staff PIC projek adalah user yang mendapatkan menu lebih terbatas. PIC Proyek merupakan user dari tim bisnis yang akan memonitor ketersediaan stok, yang kemudian akan menerbitkan DO *(Delivery Order)* atau surat perintah barang keluar.

1. **Use Case Naratif**

Use case naratif menjelaskan deskripsi dari setiap use case pada setiap aktor. Setiap skenario digambarkan dari perspektif aktor yang saling berinteraksi dengan sistem dalam berbagai cara. Skenario Use Case akan dijelaskan pada pembahasan di bawah ini:

1. Use Case Register

Use Case Register menjabarkan alur proses aktor mendaftar pada sistem. Pada proses ini juga dilengkapi dengan proses verifikasi email. Pembahasan juga menjelaskan tentang kemungkinan-kemungkinan yang akan terjadi pada proses tersebut. Use Case Register akan dijelaskan rinci pada tabel 6 berikut:

Tabel 6 Use Case Register

|  |  |  |
| --- | --- | --- |
| **Nama Use Case** | Register | |
| **Aktor** | Kepala Gudang | |
| **Deskripsi** | Use Case ini menggambarkan kegiatan aktor mendaftarkan diri dan perusahaan pada sistem. | |
| **Prekondisi** | Kepala Gudang akses halaman register | |
| **Kondisi Normal** | **Kegiatan Aktor** | **Respon Sistem** |
| 1. Kepala Gudang mengisikan form register | 1. Sistem memvalidasi apakah email sudah terdafar. 2. Sistem mengirimkan link verifikasi email 3. Simpan Data |
| **Kondisi Alternatif** | Langkah No. 2 : jika email sudah terdaftar maka sistem akan menginformasikan dan registrasi tidak dapat dilanjutkan. | |
| **Kesimpulan** | User baru sebagai Kepala Gudang berhasil terdaftarkan jika langkah No. 4 sudah berhasil. | |
| **Post Kondisi** | Dapat mengakses fitur yang sudah disediakan. | |

1. Use Case Mendaftarkan Staff

Use Case Mendaftarkan Staff menjabarkan alur proses user didaftarkan oleh Kepala Gudang. Pada proses ini juga dilengkapi dengan proses verifikasi email. Pembahasan juga menjelaskan tentang kemungkinan-kemungkinan yang akan terjadi pada proses tersebut. Use Case Mendaftarkan Staff akan dijelaskan rinci pada tabel 7 berikut:

Tabel 7 Narasi Use Case untuk Mendaftarkan Staff Gudang

|  |  |  |
| --- | --- | --- |
| **Nama Use Case** | Mendaftarkan Staff Gudang | |
| **Aktor** | Kepala Gudang | |
| **Deskripsi** | Use Case ini menggambarkan kegiatan mendaftarkan user baru oleh aktor (Kepala Gudang) | |
| **Prekondisi** | Kepala Gudang akses halaman register user baru | |
| **Kondisi Normal** | **Kegiatan Aktor** | **Respon Sistem** |
| 1. Kepala Gudang mengisikan form register staff baru. | 1. Sistem memvalidasi apakah email sudah terdafar. 2. Sistem mengirimkan link verifikasi email & password awal. 3. Simpan Data |
| **Kondisi Alternatif** | Langkah No. 2 : jika email sudah terdaftar maka sistem akan menginformasikan dan registrasi tidak dapat dilanjutkan. | |
| **Kesimpulan** | User baru berhasil terdaftarkan jika Langkah No. 4 sudah berhasil. | |
| **Post Kondisi** | User baru mendapatkan email verifikasi dan password sementara, kemudian berhasil melakukan login pada sistem yang sudah di sediakan. | |

1. Use Case Login

Use Case Login menjabarkan alur proses user login dan validasinya. Pembahasan juga menjelaskan tentang kemungkinan-kemungkinan yang akan terjadi pada proses tersebut. Use Case Login akan dijelaskan rinci pada tabel 8 berikut:

Tabel 8 Narasi Use Case untuk Login

|  |  |  |
| --- | --- | --- |
| **Nama Use Case** | Mendaftarkan Login | |
| **Aktor** | Kepala Gudang, Staff Gudang, Staff PIC Project | |
| **Deskripsi** | Use Case ini menggambarkan kegiatan login dari semua aktor | |
| **Prekondisi** | User pada halaman login | |
| **Kondisi Normal** | **Kegiatan Aktor** | **Respon Sistem** |
| 1. User mengisikan form login. | 1. Sistem memvalidasi apakah email dan password benar dan terdaftar. 2. Proses Login, kemudian diarahkan ke halaman dashboard. |
| **Kondisi Alternatif** | Langkah No. 2 : jika *email & password* salah & tidak terdaftar maka sistem akan menginformasikan kesalahan dan proses login tidak dapat dilanjutkan. | |
| **Kesimpulan** | User berhasil login jika *email & password* benar | |
| **Post Kondisi** | Dapat mengakses menu yang disediakan oleh sistem. | |

1. Use Case Mengelola Data Master Produk

Use Case Mengelola Data Master Produk menjabarkan proses user mengelola Data Produk. Mengelola di sini bermakna mulai insiasi data pertama produk, update stok, update spesifikasi, dan lainnya. Pembahasan lebih detail akan dijelaskan rinci pada tabel 9 berikut:

Tabel 9 Narasi Use Case untuk Mengelola Data Master Produk

|  |  |  |
| --- | --- | --- |
| **Nama Use Case** | Mengelola Data Master Produk | |
| **Aktor** | Kepala Gudang & Staff Gudang | |
| **Deskripsi** | Use Case ini menggambarkan kegiatan mengelola data master produk. | |
| **Prekondisi** | User sudah login | |
| **Kondisi Normal** | **Kegiatan Aktor** | **Respon Sistem** |
| 1. Masuk pada halaman produk. 2. Klik tombol “Tambah Produk Baru” 3. Masukan data produk 4. Klik simpan. | 1. Sistem memvalidasi inputan user. 2. Simpan data produk baru. |
|  | 1. Produk baru bisa dilihat di halaman daftar produk. 2. Klik tombol “Ubah” untuk mengedit data tentang produk 3. Edit data produk 4. Klik Simpan | 1. Sistem memvalidasi inputan user. 2. Simpan perubahan data produk. |
|  | 1. Klik tombol “Hapus” | 1. Memberikan konfirmasi untuk hapus data 2. Hapus Data |
| **Kondisi Alternatif** | Langkah No. 5 & 11: jika isian tidak sesuai rule yang dipinta sistem maka tidak dapat dilanjutkan. | |
| **Kesimpulan** | Mengelola data master produk ini berupa tambah, ubah, hapus data produk. | |
| **Post Kondisi** | Melihat daftar produk beserta informasi yang termuat. | |

1. Use Case Mengelola Barang Masuk

Use Case Mengelola Barang Masuk menjabarkan proses aktor mengelola Data Barang Masuk. Proses ini akan menjelaskan barang datang digudang kemudian menginputkan data barang masuk pada sistem inventori. Barang yang tiba digudang perlu didata agar stock dari masing-masing barang dapat terpantau. Pembahasan lebih detail akan dijelaskan rinci pada tabel 10 berikut:

Tabel 10 Narasi Use Case untuk Mengelola Barang Masuk

|  |  |  |
| --- | --- | --- |
| **Nama Use Case** | Mengelola Barang Masuk | |
| **Aktor** | Kepala Gudang & Staff Gudang | |
| **Deskripsi** | Use Case ini menggambarkan kegiatan mengelola data barang masuk. | |
| **Prekondisi** | User sudah login | |
| **Kondisi Normal** | **Kegiatan Aktor** | **Respon Sistem** |
| 1. Masuk pada halaman Barang Masuk. 2. Klik tombol “Tambah Barang Masuk” 3. Masukan data barang masuk 4. Klik simpan. | 1. Sistem memvalidasi inputan aktor. 2. Simpan data produk baru. |
|  | 1. Data Barang Masuk bisa dilihat di halaman daftar barang masuk. 2. Klik tombol “Ubah” untuk mengedit data barang masuk 3. Edit data 4. Klik Simpan | 1. Sistem memvalidasi inputan user. 2. Simpan perubahan data barang masuk. |
|  | 1. Klik tombol “Hapus” | 1. Memberikan konfirmasi untuk hapus data 2. Hapus Data |
| **Kondisi Alternatif** | Langkah No. 5 & 11: jika isian tidak sesuai rule yang dipinta sistem maka tidak dapat dilanjutkan. | |
| **Kesimpulan** | Mengelola data barang masuk ini berupa tambah, ubah, hapus. | |
| **Post Kondisi** | Melihat daftar barang masuk beserta informasi yang termuat. | |

1. Use Case Mengelola Barang Keluar

Use Case Mengelola Barang Keluar menjabarkan proses aktor mengelola Data Barang Keluar. Proses ini menjelaskan barang yang akan dikeluarkan dari Gudang. Barang yang keluar dari gudang perlu didata agar stock dari masing-masing barang dapat terpantau. Pembahasan lebih detail akan dijelaskan rinci pada tabel 11 berikut:

Tabel 11 Narasi Use Case untuk Mengelola Barang Keluar

|  |  |  |
| --- | --- | --- |
| **Nama Use Case** | Mengelola Barang Keluar | |
| **Aktor** | Kepala Gudang & Staff Gudang | |
| **Deskripsi** | Use Case ini menggambarkan kegiatan mengelola data barang keluar. | |
| **Prekondisi** | User sudah login dan mendapatkan perintah barang keluar melalui sistem dengan menerbitkan dokumen *Delivery Order (DO).* | |
| **Kondisi Normal** | **Kegiatan Aktor** | **Respon Sistem** |
| 1. Masuk pada halaman Barang Keluar. 2. Klik tombol “Tambah Barang Keluar” 3. Masukan data barang yang keluar. 4. Klik simpan. | 1. Sistem memvalidasi inputan aktor. 2. Simpan data barang keluar. |
|  | 1. Data barang keluar bisa dilihat di halaman daftar barang keluar. 2. Klik tombol “Ubah” untuk mengedit data barang keluar 3. Edit data 4. Klik Simpan | 1. Sistem memvalidasi inputan user. 2. Simpan perubahan data barang keluar. |
|  | 1. Klik tombol “Hapus” | 1. Memberikan konfirmasi untuk hapus data 2. Hapus Data |
| **Kondisi Alternatif** | Langkah No. 5 & 11: jika isian tidak sesuai rule yang dipinta sistem maka tidak dapat dilanjutkan. | |
| **Kesimpulan** | Mengelola data barang keluar ini berupa tambah, ubah, dan hapus. | |
| **Post Kondisi** | Melihat daftar barang keluar beserta informasi yang termuat. | |

1. Use Case Memantau Stok

Use Case Memantau Stok menjabarkan proses aktor memantau ketersediaan stok pada masing-masing barang. Pada tahapan ini aktor memantau ketersediaan stok yang mana akan dijadikan pelaporan dan acuan untuk proses transaksi dalam bisnis perusahaan. Pembahasan lebih detail akan dijelaskan rinci pada tabel 12 berikut:

Tabel 12 Narasi Use Case untuk Memantau Stok

|  |  |  |
| --- | --- | --- |
| **Nama Use Case** | Memantau Stok | |
| **Aktor** | Kepala Gudang, Staff Gudang, PIC Project | |
| **Deskripsi** | Use Case ini menggambarkan kegiatan memantau stok | |
| **Prekondisi** | User sudah login | |
| **Kondisi Normal** | **Kegiatan Aktor** | **Respon Sistem** |
| 1. Masuk pada halaman Daftar Barang. 2. Aktor langsung bisa melihat daftar barang serta stok terakhir. 3. Aktor mengetikan keyword untuk pencarian barang yang diinginkan. | 1. Sistem memproses pencarian data. 2. Sistem menampilkan data sesuai dengan keyword pencarian. |
| **Kondisi Alternatif** | Jika diperlukan data berupa file spreadsheet, sistem juga menyediakan fitur untuk eksport data. | |
| **Kesimpulan** | Aktor dapat langsung memantau ketersediaan stok dilengkapi dengan proses pencarian data. | |
| **Post Kondisi** | Melihat daftar barang dan stok beserta informasi yang termuat. | |

1. Use Case Menerbitkan Perintah Barang Keluar / *Delivery Order (DO)*

Use Case ini menjelaskan proses aktor memantau stok yang dilanjutkan untuk menerbitkan perintah barang keluar. Pada tahapan ini aktor memantau ketersediaan stok kemudian menyandingkan dengan kebutuhan barang yang harus dikeluarkan. Jika kebutuhan sudah sesuai, maka terbitlah perintah barang keluar. Pembahasan lebih detail akan dijelaskan rinci pada tabel 13 berikut

Tabel 13 Narasi Use Case untuk Menerbitkan Perintah Barang Keluar

|  |  |  |
| --- | --- | --- |
| **Nama Use Case** | Menerbitkan Perintah Barang Keluar | |
| **Aktor** | PIC Project | |
| **Deskripsi** | Use Case ini menggambarkan kegiatan memantau stok kemudian menerbitkan perintah barang keluar | |
| **Prekondisi** | User sudah login | |
| **Kondisi Normal** | **Kegiatan Aktor** | **Respon Sistem** |
| 1. Masuk pada halaman “Delivery Order”. 2. Klik Tombol “Buat Delivery Order Baru”. 3. Pilih barang dan kuantiti yang akan dikeluarkan. | 1. Sistem memvalidasi ketersediaan stok. 2. Sistem memproses dan menerbitkan perintah barang keluar / *Delivery Order*. |
| **Kondisi Alternatif** | Langkah No.4: jika validasi menunjukan stok pada gudang kurang dari kuantiti yang ditulis aktor, sistem tetap akan menyimpan Delivery Order tapi status masih sebagai *draft.* | |
| **Kesimpulan** | Proses ini berisi menyandingkan data order dari bisnis dengan ketersediaan stok digudang. | |
| **Post Kondisi** | Delivery Order berhasil diterbitkan, Staff Gudang bisa memproses barang keluar. | |

1. **Activity Diagram**

Activity Diagram menggambarkan alur kerja dari suatu sistem perangkat lunak yang dikembangkan. Aktifitas yang dilakukan oleh aktor dan sistem dapat digambarkan pada diagram ini. Biasanya activity diagram dibuat dari use case yang telah dibuat terlebih dahulu, kemudian dibuatlah activity diagram dari masing-masing use case. Berikut penjelasan dari setiap use case dalam bentuk activity diagram.

1. Activity Diagram Register

Diagram

Description automatically generated with medium confidence

Gambar 3.3.4 Activity Diagram Register

Gambar 3.3.4 menjelaskan alur aktor mendaftarkan akun dan perusahaan yang menggunakan sistem inventori. Aktor yang mendaftar akan diberikan role sebagai Kepala Gudang, dimana role ini yang bertindak selayaknya super admin atas akun di perusahaan pada sistem inventori.

1. Activity Diagram Login

Diagram

Description automatically generated

Gambar \* menjelaskan alur aktor melakukan proses login pada sistem inventori. Pertama-tama aktor akan mengakses halaman website, kemudian klik tombol login. Pada halaman login aktor akan memasukan email dan password. Sistem inventori akan memvalidasi inputan dan memeriksa apakah email dan password terdaftar pada sistem inventori. Jika validasi berhasil dan terdaftar maka proses login berjalan, aktor akan diarahkan ke menu dashboard sistem inventori.

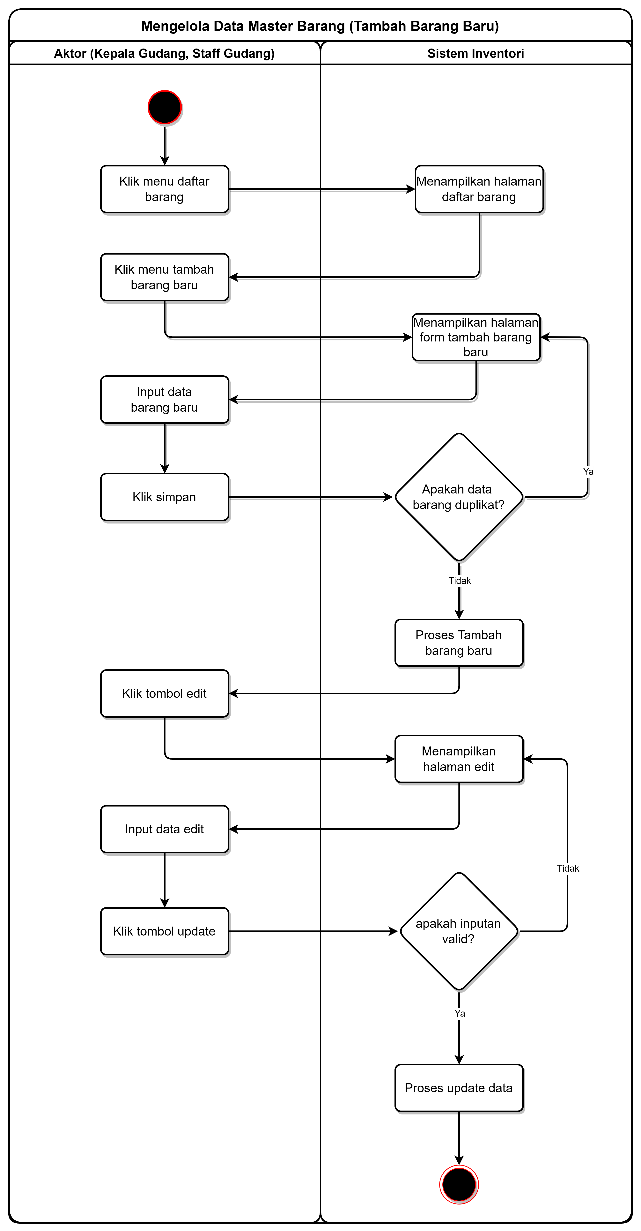
1. Activity Diagram Mendaftarkan Staff

Diagram

Description automatically generated with medium confidence

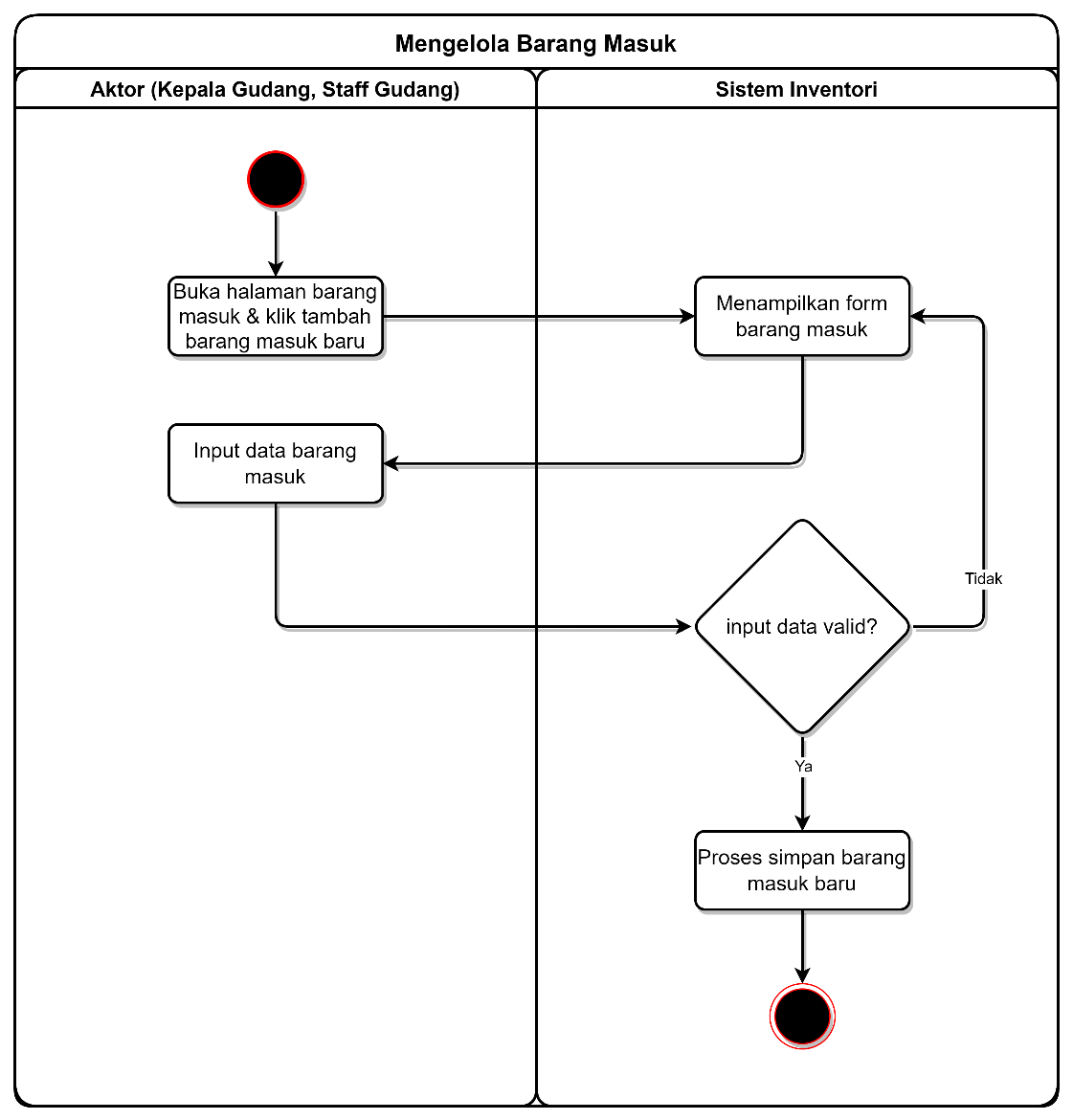
Gambar \* menjelaskan Kepala Gudang yang mendaftarkan staff lain untuk membantu proses pendataan di sistem inventori. Kepala Gudang mengakses halaman daftarkan staff, kemudian input email dan data sesuai form yang di sediakan. Sistem inventori akan memvalidasi email apakah sudah terdaftar? Jika belum maka proses daftar akan dilakukan.

1. Activity Diagram Mengelola Data Master Barang



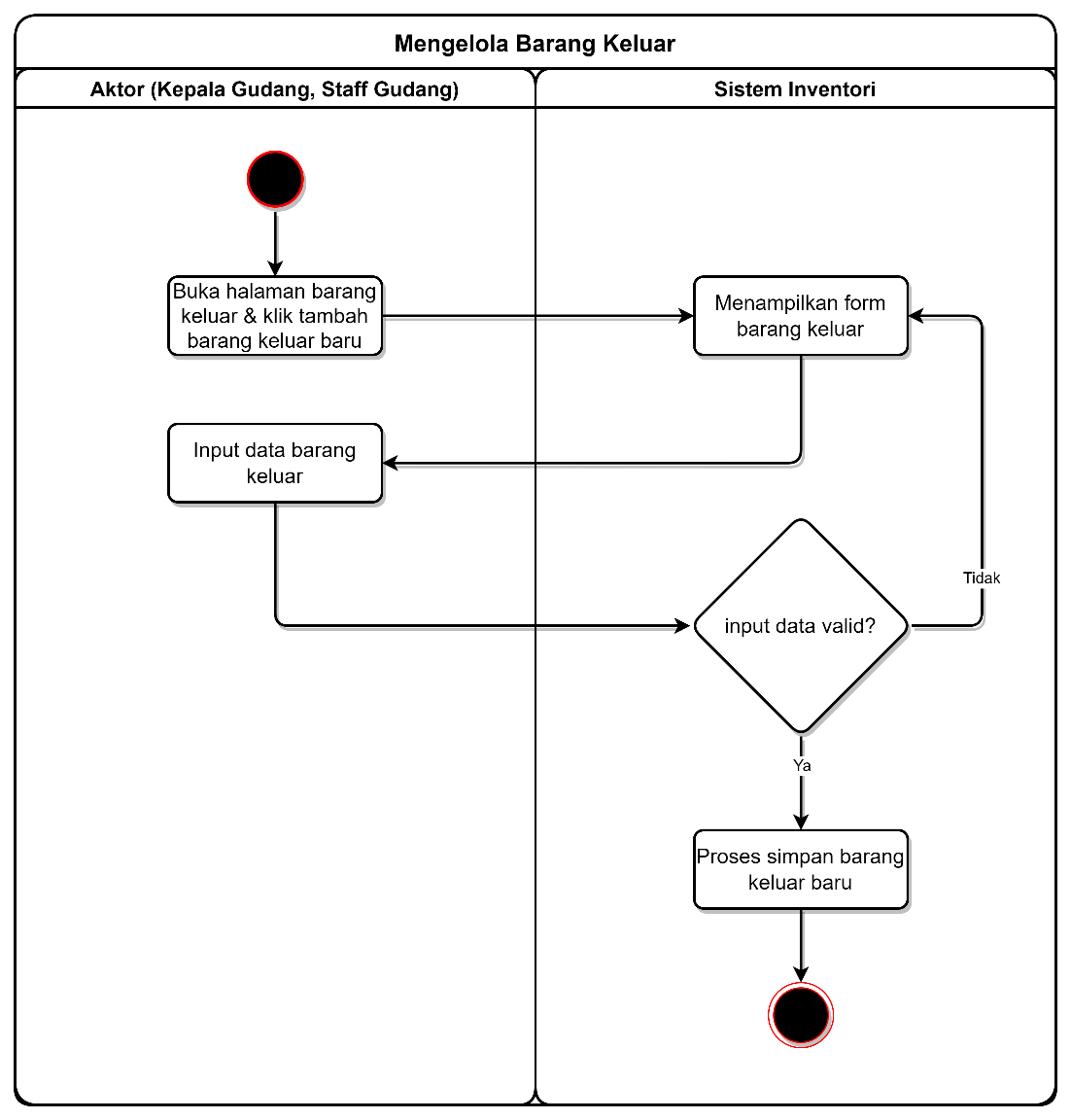
Gambar \* menjelaskan Kepala Gudang dan Staff Gudang pada saat akan melakukan pengelolaan data master barang di sistem inventori. Masing-masing aktor harus melakukan login terlebih dahulu agar dapat mengakses menu halaman data master barang. Aktor akan diberikan menu untuk menambahkan data barang baru. Aktor akan memasukan data-data spesifikasi barang dan data stok, kemudian klik simpan untuk menyimpan. Aktor juga dapat melakukan perubahan data pada barang yang sudah berhasil dimasukan. Aktor bisa menggunakan menu edit kemudian update data barang. Data update dari aktor akan divalidasi oleh sistem kemudian disimpan.

1. Activity Diagram Mengelola Barang Masuk



Gambar \* ini menjelaskan kepala gudang dan staff melakukan proses pengelolaan barang masuk. Aktor telah berhasil login pada sistem dan memilih menu tambah barang masuk baru. Aktor mengisikan data pada form barang masuk. Isian data akan divalidasi dan disimpan.

1. Activity Diagram Mengelola Barang Keluar



Gambar \* ini menjelaskan kepala gudang dan staff melakukan proses pengelolaan barang keluar. Aktor telah berhasil login pada sistem dan memilih menu tambah barang keluar baru. Aktor mengisikan data pada form barang keluar. Isian data akan divalidasi dan disimpan.

1. Activity Diagram Memantau Stok

Graphical user interface

Description automatically generated

Gambar \* ini menjelaskan semua aktor dapat memantau stok. Pertama-tama aktor harus berhasil login, kemudian akses menu daftar barang. Halaman daftar barang akan menampilkan data stok yang tersedia. Selain di halaman daftar barang juga tersedia pada dashboard tabel data 10 barang dengan stok tersedikit.

1. Activity Diagram Memantau Stok

Diagram

Description automatically generated

Pada gambar \* ini adalah activity diagram khusus aktor Staff PIC Project. Activity Diagram ini menjelaskan prose aktor akan mengeluarkan surat perintah barang keluar atau *Delivery Order (DO)*. Aktor harus sudah berhasil login, kemudian mengakses menu DO. Aktor akan diberikan form DO dimana sudah tersedia pilihan barang dan stok terakhir. Setelah memilih item barang dilanjutkan untuk mengisikan kuantiti barang yang diharuskan untuk keluar gudang. Jika semua form sudah terisi semua, klik simpan. Sistem akan memvalidasi apakah stok dalam gudang memenuhi kebutuh kuantiti barang yang akan dikeluarkan. Jika stok memenuhi maka DO berhasil diterbitkan, sebaliknya jika stok tidak dapat memenuhi kebutuhan maka DO akan simpan sebagai draft, untuk dapat dilanjutkan jika stok dalam gudang memenuhi kebutuhan.

1. **Class Diagram**

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. Gambar \* adalah *class diagram* dari sistem inventori gudang pada PT. Ladang Karya Husada.

A picture containing timeline

Description automatically generated

### **Detailed Design**

//TODO

# DAFTAR PUSTAKA

Ardian Dwi Praba, M. S. (2020). Studi Perbandingan Performansi Antara Mysql Dan Postgresql. *Jurnal Khatulistiwa Informatika*, *VIII*(2), 1–6.

Aryadi, R., Suyanto, & Widodo. (2020). Aplikasi Testing Interface Video Graphics Array Card Menggunakan Vb.Net. *Jurnal Sibernetika*, *5*(2), 209–215. http://jurnas.stmikmj.ac.id/index.php/sibernetika/article/view/99

Bagir, H., & Putro, B. E. (2018). Analisis Perancangan Sistem Informasi Pergudangan di CV. Karya Nugraha. *Jurnal Media Teknik Dan Sistem Industri*, *2*(1), 30. https://doi.org/10.35194/jmtsi.v2i1.274

Efendi, T. F., & Krisanty, M. (2020). Warehouse Data System Analysis PT. Kanaan Global Indonesia. *International Journal of Computer and Information System (IJCIS)*, *1*(3), 70–73. https://doi.org/10.29040/ijcis.v1i2.26

Ernawati, S., & Wati, R. (2021). ANDROID-BASED QURAN APPLICATION ON THE FLUTTER FRAMEWORK BY USING THE FOUNTAIN MODEL. *JURNAL RISET INFORMATIKA*, *3*(2). https://doi.org/10.34288/jri.v3i2.205

Ernawati, S., Wati, R., & Maulana, I. (2021). Penerapan Model Fountain Untuk Pengembangan Aplikasi Text Recognitiondan Text To Speechberbasis Android Menggunakan Flutter. *Prosiding Snast*, 178–186. https://journal.akprind.ac.id/index.php/prosidingsnast/article/view/3405/2472

Fitri, M., & Irsya Putri2, D. (2021). *Usulan Rancangan Tata Letak Gudang Penyimpanan Kantong Semen Menggunakan Metode Shared Storage*. Jurnal Teknologi Dan Sistem Informasi Bisnis. https://doi.org/10.47233/jteksis.v3i1.219

Fitzgerald, J., Oda, T., & Macedo, H. D. (2021). *Proceedings of the 18th International Overture Workshop*. http://arxiv.org/abs/2101.07261

Hadden, L. S., & Hughes, T. M. (2019). *DOMinate your ODS Output with PROC TEMPLATE , ODS Cascading Style Sheets ( CCS ), and the ODS Document Object Model ( DOM ) by Mastering ODS TRACE DOM , Parsing CSS with a PROC TEMPLATE Front End , and Overriding ODS Style Inheritance with Customized Styl*. 1–18.

Herdiana, G. A., & Sudarma, M. (2021). *Design of Internet Problem Report Management System in Diskominfos Bali Province*. *6*(2), 88–93.

Jaharuddin, Utama, R. E., Gani, N. A., & Priharta, A. (2020). *Buku Manajemen Operasi Full* (Issue November 2019).

Kambivi, H., Junirianto, E., & Fadhliyah, N. R. (2020). Development of Inventory Management Application Points of Sale Using Laravel. *Tepian*, *1*(1), 9–17.

Pahlevi, O., Mulyani, A., & Khoir, M. (2018). Sistem informasi inventori barang menggunakan metode object oriented di pt. Livaza teknologi indonesia jakarta. *Pt. Livaza Teknologi Indonesia Jakarta*, *5*(1Pahlevi, O., Mulyani, A., Khoir, M. (2018). Sistem Informasi Inventori Barang Menggunakan Metode Object Oriented Di Pt. Livaza Teknologi Indonesia Jakarta. Pt. Livaza Teknologi Indonesia Jakarta, 5(1), 27–35.), 27–35.

Putra, D. W. T., & Andriani, R. (2019). Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD. *Jurnal TeknoIf*, *7*(1), 32. https://doi.org/10.21063/jtif.2019.v7.1.32-39

Rachman, A., Andreansyah, & Rahmi. (2020). *Implementation of Incremental Models on Development of Web-Based Loan Cooperative Applications*. International Journal of Education, Science, Technology, and Engineering. https://doi.org/10.36079/lamintang.ijeste-0301.105

Setiawan, R. (2021). *Metode SDLC Dalam Pengembangan Software - Dicoding Blog*. Dicoding. https://www.dicoding.com/blog/metode-sdlc/

Setyawan, R. A., & Atapukan, W. F. (2018). Pengukuran Usability Website E-Commerce Sambal Nyoss Menggunakan Metode Skala Likert. *Compiler*, *7*(1), 54–61. https://doi.org/10.28989/compiler.v7i1.254

Wongso, D. (2020). Perancangan dan Implementasi Website Pariwisata di Desa Sembulang dengan Metode Extreme Programming. *Journal of Information System and Technology*, *01*(02), 142–162.

Yu, J. (2018). Research Process on Software Development Model. *IOP Conference Series: Materials Science and Engineering*, *394*(3). https://doi.org/10.1088/1757-899X/394/3/032045