**Implementasi Algoritma *K-Means Clustering* untuk Pengelompokan Pola Permintaan Barang**

**dalam Sistem Manajemen Inventori**

**PT Semen Padang**

**PROPOSAL TUGAS AKHIR**

*Diajukan Sebagai Salah Satu Syarat Memperoleh Gelar Sarjana (S1) Pada Program Studi Informatika Departemen Teknik Elektronika Fakultas Teknik Universitas Negeri Padang*



**IFDAL LISYUKRI**

**21346012**

**PROGRAM STUDI INFORMATIKA**

**DEPARTEMEN TEKNIK ELEKTRONIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS NEGERI PADANG**

**2025**

# DAFTAR ISI

Halaman

[DAFTAR ISI i](#_Toc191046082)

[DAFTAR TABEL iii](#_Toc191046083)

[DAFTAR GAMBAR iv](#_Toc191046084)

[BAB I PENDAHULUAN 1](#_Toc191046085)

[A. Latar Belakang 1](#_Toc191046086)

[B. Identifikasi Masalah 3](#_Toc191046087)

[C. Rumusan Masalah 4](#_Toc191046088)

[D. Tujuan Penelitian 4](#_Toc191046089)

[E. Manfaat Penelitian 5](#_Toc191046090)

[1. Bagi Perusahaan 5](#_Toc191046091)

[2. Bagi Karyawan 6](#_Toc191046092)

[3. Bagi Peneliti 7](#_Toc191046093)

[4. Bagi Dunia Akademik 7](#_Toc191046094)

[BAB II LANDASAN TEORI 9](#_Toc191046095)

[A. Pengertian Manajemen Inventori 9](#_Toc191046096)

[B. Sistem Informasi Berbasis *Web* 10](#_Toc191046097)

[C. Data *Clustering* dan Algoritma *K-Means* 11](#_Toc191046098)

[1. Pengertian *Clustering* 11](#_Toc191046099)

[2. Algoritma *K-Means* 11](#_Toc191046100)

[D. Penerapan *K-Means* pada Sistem Manajemen Inventori 14](#_Toc191046101)

[E. Metode Pengembangan 16](#_Toc191046102)

[F. Teknologi Pendukung 18](#_Toc191046103)

[G. Kerangka Teori Penelitian 20](#_Toc191046104)

[H. Penelitian Relevan 21](#_Toc191046105)

[BAB III ANALISIS DAN PERANCANGAN SISTEM 24](#_Toc191046106)

[A. Analisis Kebutuhan 24](#_Toc191046107)

[1. Kebutuhan Fungsional 24](#_Toc191046108)

[2. Kebutuhan Non-Fungsional 24](#_Toc191046109)

[3. Identifikasi Aktor 26](#_Toc191046110)

[4. Diagram Konteks 26](#_Toc191046111)

[B. Desain Sistem 27](#_Toc191046112)

[1. Arsitektur Sistem 27](#_Toc191046113)

[2. Diagram Alur Sistem 27](#_Toc191046114)

[3. *Activity Diagram* 30](#_Toc191046115)

[4. *Entity-Relationship Diagram (ERD)* 37](#_Toc191046116)

[5. *Class Diagram* 38](#_Toc191046117)

[6. *Database Design* 39](#_Toc191046118)

[7. Desain Antarmuka 43](#_Toc191046119)

[C. Implementasi Algoritma *K-Means* 48](#_Toc191046120)

[1. Proses *Clustering* dalam Sistem 48](#_Toc191046121)

[2. *Data Preprocessing* 50](#_Toc191046122)

[D. Teknologi dan *Tools* 50](#_Toc191046123)

[E. Use Case Diagram 51](#_Toc191046124)

[F. Pengelolaan Data 53](#_Toc191046125)

[1. Jenis Data 53](#_Toc191046126)

[2. Format Data 54](#_Toc191046127)

[3. Alur Pengolahan Data 54](#_Toc191046128)

[DAFTAR PUSTAKA 56](#_Toc191046129)

# DAFTAR TABEL

[Tabel. 1 Penelitian yang Relevan 21](#_Toc190995066)

[Tabel. 2 Tabel Barang Daerah 40](#_Toc190995067)

[Tabel. 3 Tabel Barang Gudang 41](#_Toc190995068)

[Tabel. 4 Tabel Requests 41](#_Toc190995069)

[Tabel. 5 Tabel Requests Gudang 42](#_Toc190995070)

[Tabel. 6 Tabel Users 43](#_Toc190995071)

# DAFTAR GAMBAR

[Gambar. 1 *Context Diagram* 26](#_Toc190995096)

[Gambar. 2 Arsitektur Sistem 27](#_Toc190995097)

[Gambar. 3 *Flowchart* 28](#_Toc190995098)

[Gambar. 4 *Activity Diagram* Registrasi 31](#_Toc190995099)

[Gambar. 5 *Activity Diagram Login* 31](#_Toc190995100)

[Gambar. 6 *Activity Diagram* Mencari Barang 32](#_Toc190995101)

[Gambar. 7 *Activity Diagram* Mengajukan Permintaan Barang 33](#_Toc190995102)

[Gambar. 8 *Activity Diagram* Tinjau Permintaan Barang 34](#_Toc190995103)

[Gambar. 9 *Activity Diagram* *Clustering* Permintaan Barang 35](#_Toc190995104)

[Gambar. 10 *Activity Diagram* Input Data Barang 36](#_Toc190995105)

[Gambar. 11 *Activity Diagram Logout* 37](#_Toc190995106)

[Gambar. 12 *Entity-Relationship Diagram (ERD)* 38](#_Toc190995107)

[Gambar. 13 *Class Diagram* 39](#_Toc190995108)

[Gambar. 14 Desain Halaman *Login* 44](#_Toc190995109)

[Gambar. 15 Desain Halaman *Login* 44](#_Toc190995110)

[Gambar. 16 Tampilan Tabel Barang 45](#_Toc190995111)

[Gambar. 17 Fitur Pencarian dan Filter Barang 45](#_Toc190995112)

[Gambar. 18 Form Permintaan Barang 46](#_Toc190995113)

[Gambar. 19 Fitur Input Data Barang 47](#_Toc190995114)

[Gambar. 20 Tabel Riwayat Permintaan Barang 47](#_Toc190995115)

[Gambar. 21 Hasil *Clustering* dengan *Pie Chart* 48](#_Toc190995116)

[Gambar. 22 *Use Case Diagram* 51](#_Toc190995117)

# BAB I PENDAHULUAN

1. Latar Belakang

Perkembangan teknologi industri telah membawa perubahan signifikan dalam berbagai sektor termasuk sektor manufaktur dan logistik. Dengan hadirnya revolusi industri 4.0. pemanfaatan teknologi digital, *Internet of Things (IoT)*, dan sistem otomatisasi telah menjadi faktor utama dalam meningkatkan efisiensi dan produktivitas industri (Hery et al., 2022). Teknologi tidak hanya berperan dalam mempercepat proses produksi tetapi juga dalam mengoptimalkan pengelolaan sumber daya termasuk sistem manajemen inventori.

Manajemen inventori merupakan sistem yang digunakan untuk mengelola persediaan barang. Mengelola bagaimana mengklarifikasi dan menjaga akurasi catatan persediaan barang (Nurcahyawati et al., 2023). Tanpa inventori yang terkelola dengan baik, proses operasional perusahaan dapat terhambat. Oleh karena itu dalam dunia industri modern, penerapan teknologi dalam manajemen inventori menjadi sangat krusial untuk memastikan kelancaran distribusi dan ketersediaan stok barang. Sistem manajemen inventori yang baik harus mampu menyediakan informasi *real-time* mengenai jumlah stok, lokasi penyimpanan, serta pola permintaan barang. Teknologi berbasis *web* dan algoritma *data mining* seperti *K-Means Clustering* telah terbukti dapat meningkatkan akurasi dan efisiensi dalam pengelolaan inventori, menggantikan metode manual yang rentan terhadap kesalahan pencatatan dan inkonsistensi data (Handoko et al., 2020).

Salah satu contoh perusahaan yang menghadapi tantangan dalam pengelolaan inventori adalah PT Semen Padang, perusahaan manufaktur semen terbesar di Indonesia dengan struktur operasional yang kompleks. Wilayah operasionalnya terbagi dalam beberapa daerah, yaitu Indarung 1 hingga 6, di mana saat ini yang masih beroperasi adalah Indarung 4, 5, dan 6. Setiap daerah ini terbagi lagi ke dalam beberapa area operasional diantaranya *Raw Mill*, *Kiln* dan *Finish Mill*, yang memiliki gudang penyimpanan barang yang dikategorikan menjadi gudang *electrical* dan *mechanical*.

Setelah melakukan wawancara dengan beberapa staf PT Semen Padang salah satu kendala utama dalam perusahaan ini adalah tidak adanya teknologi yang dapat menghubungkan pengelolaan barang antar area dan antar daerah secara terpusat. Pengelolaan yang ada di setiap gudang masih bersifat manual dengan pencatatan menggunakan *Microsoft excel*. Sistem manual ini memiliki berbagai keterbatasan seperti kesulitan dalam sinkronisasi data antar area, potensi kehilangan data, serta keterbatasan dalam pelacakan barang secara *real-time*. Hal ini menyebabkan inefisiensi dalam manajemen inventori dan berisiko menghambat operasional perusahaan (Lisa et al., 2025).

Selain itu sistem permintaan barang di PT Semen Padang juga masih dilakukan secara manual, dimana setiap permintaan barang harus diajukan melalui dokumen fisik atau komunikasi langsung dengan pihat terkait. Proses ini tidak hanya memakan waktu lebih lama, tetapi juga rentan terhadap kesalahan pencatatan da keterlambatan dalam pemenuhan permintaan barang. Ketiadaan sistem otomatisasi dalam pencatatan dan pengelolaan permintaan barang membuata proses ini menjadi kurang efektif dan tidak efisien.

Kurangnya analisis terhadap pola permintaan barang menjadi permasalahan tambahan dalam pengelolaan inventori PT Semen Padang. Ketiadaan sistem yang mengidentifikasi pola permintaan barang di setiap area menyebabkan kesulitan dalam menentukan strategi optimal dalam pengelolaan stok. Akibatnya perusahaan mengalami ketidakseimbangan antara persediaan dan permintaan, yang berujung pada kelebihan stok (*overstock*) atau kekurangan stok (*stockout*) yang menghambat kelancaran operasional (Andi Syahrul Ramdana et al., 2024).

Menurut (Calista et al., 2023), sistem informasi inventori berbasis *web* dapat meningkatkan efisiensi dalam pengelolaan stok barang serta meminimalkan kesalahan pencatatan yang sering terjadi dalam sistem manual. Dalam konteks perusahaan besar seperti PT Semen Padang, pengelolaan inventori memegang peranan penting dalam memastikan kelancaran produksi, distribusi barang, dan pemenuhan kebutuhan operasional di berbagai area perusahaan. Maka dari itu, sistem berbasis *web* dipilih sebagai platform utama dalam pengembangan sistem manajemen inventori karena memiliki fleksibilitas dan aksesibilitas yang tinggi. Dengan integrasi data secara *real-time*, semua area dan gudang dapat mengakses informasi yang sama tanpa adanya keterlambatan atau inkonsistensi data, sehingga proses operasional menjadi lebih efisien. Selain itu, sistem berbasis *web* juga lebih mudah dalam hal pemeliharaan dan pengembangan dibandingkan dengan sistem berbasis *desktop*, menjadikannya solusi yang lebih optimal untuk kebutuhan inventori perusahaan.

Namun, penerapan sistem berbasis *web* saja belum cukup untuk mengoptimalkan inventori. Penelitian sebelumnya yang dilakukan oleh (Nurdiyansyah & Akbar, 2021) menunjukkan bahwa implementasi algoritma *K-Means* dalam manajemen persediaan barang mampu meningkatkan efisiensi pengelolaan stok dengan cara mengelompokkan barang berdasarkan pola permintaan. Oleh karena itu, algoritma *K-Means* digunakan dalam proyek ini didasarkan pada kemampuannya dalam mengelompokkan data yang memiliki pola tertentu. Dengan menggunakan metode *clustering* ini perusahaan dapat melakukan analisis data permintaan barang secara lebih mendalam dan mengambil keputusan yang lebih akurat dalam pengelolaan inventori.

Dari keseluruhan pemaparan diatas maka penulis memutuskan untuk memilih judul “Implementasi Algoritma *K-Means Clustering* untuk mengelompokkan Pola Permintaan Barang dalam Sistem Manajemen Inventori PT Semen Padang”. Judul ini dipilih untuk menggambarkan permasalahan utama yang dihadapi oleh PT Semen Padang serta solusi yang ditawarkan melalui penerapan teknologi berbasis *web* dan algoritma *K-Means Clustering*.

1. Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan, berikut adalah poin-poin utama perrmasalahan yang dihadapi PT Semen Padang dalam sistem manajemen inventori:

1. Pengelolaan inventori dan permintaan barang masih dilakukkan secara manual
2. Sistem manajemen inventori belum terintegrasi antar area dan daerah
3. Belum diterapkannya sistem berbasis *web* dalam manajemen inventori
4. Kurangnya sistem analisis dalam pengelolaan stok dan pola permintaan barang
5. Rumusan Masalah

Berdasarkan identifikasi masalah yang telah disusun, penelitian ini bertujuan untuk menjawab beberapa pertanyaan berikut:

1. Bagaimana mengembangkan sistem manajemen inventori berbasis *web* yang dapat mengintegrasikan data dari berbagai area dan gudang di PT Semen Padang?
2. Bagaimana meningkatkan efisiensi pencatatan dan pelacakan stok barang dengan sistem berbasis *web* yang lebih akurat dan *real-time*?
3. Bagaimana sistem berbasis *web* dapat mengotomatisasi proses permintaan barang agar lebih cepat dan mengurangi kesalahan perncatatan?
4. Bagaimana menerapkan algoritma *K-Means Clustering* untuk menganalisis dan mengelompokkan pola permintaan barang guna mengoptimalkan manajemen stok?
5. Tujuan Penelitian

Tujuan penelitian ini disusun berdasarkan rumusan masalah di atas:

1. **Mengembangkan sistem manajemen inventori berbasis *web* yang memungkinkan integrasi data secara *real-time* di seluruh area dan gudang PT Semen Padang.**
2. Meningkatkan efisiensi pencatatan dan pemantauan stok barang dengan sistem pencatatan yang lebih akurat, terdokumentasi dengan baik, serta dapat diakses secara *real-time*.
3. Mengotomatisasi proses permintaan barang antar area dan daerah untuk mempercepat waktu pemrosesan serta meminimalisir kesalahan pencatatan dan keterlambatan pemenuhan permintaan barang.
4. Menerapkan algoritma *K-Means Clustering* untuk mengelompokkan pola permintaan barang guna membantu pengambilan keputusan dalam pengelolaan inventori.
5. Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah:

1. Bagi Perusahaan
2. Mempermudah pendataan stok barang

Sistem berbasis *web* yang terintegrasi memungkinkan pengelolaan data stok barang di seluruh gudang secara otomatis dan *real-time*. Dengan demikian, proses pencatatan stok menjadi lebih cepat, akurat, dan minim kesalahan.

1. Mempermudah pemantauan permintaan barang

Proses permintaan barang yang sebelumnya dilakukan secara manual kini dapat dilakukan secara otomatis melalui aplikasi, sehingga mempermudah pengguna dalam mengajukan permintaan barang dan memantau status permintaan tanpa perlu dokumen fisik.

1. Mengurangi beban administratif manual di gudang

Dengan adanya sistem otomatisasi, tugas administratif seperti pencatatan stok, pemrosesan barang, dan pelaporan data dapat dilakukan lebih efisien. Hal ini mengurangi beban kerja karyawan gudang yang sebelumnya dihabiskan untuk mengelola dokumen fisik atau file *Excel*.

1. Meningkatkan efisiensi kerja karyawan gudang

Sistem ini membantu karyawan gudang dalam melakukan tugas operasional sehari-hari, seperti memantau ketersediaan barang dan menyetujui permintaan, dengan lebih cepat dan mudah. Efisiensi kerja meningkat karena sistem dapat memberikan informasi stok barang secara instan dan memproses permintaan secara otomatis.

1. Mengurangi resiko kesalahan pencatatan

Dengan pengelolaan data secara digital, resiko kehilangan atau keesalahan pencatatan yang sering terjadi pada sistem manual dapat diminimalkan. Data yang tersimpan secara terpusat juga dapat diakses oleh semua pihak yang berkepentingan, sehingga memastikan konsistensi informasi.

1. Meningkatkan aksesibilitas dan kecepatan informasi

Sistem berbasis *web* memungkinkan informasi terkait stok barang, permintaan, dan laporan ketersediaan dapat diakses kapan saja dan dimana saja melalui perangkat yang terhubung internet. Hal ini mempermudah pengambilan keputusan berdasarkan data yang selalu terbarui.

1. Mendukung pengambilan keputusan strategis

Dengan penerapan algoritma *K-Means Clustering*, sistem dapat memberikan wawasan terkait pola permintaan barang. Hasil analisis ini membantu perusahaan dalam menentukan prioritas pengadaan barang, mengoptimalkan persediaan, dan meningkatkan efisiensi operasional secara keseluruhan.

1. Bagi Karyawan
2. Menyederhanakan proses operasional

Karyawan gudang tidak lagi perlu memperbarui data stok barang secara manual untuk mencatat permintaan barang menggunakan dokumen fisik. Semua proses kini dapat dilakukan melalui aplikasi yang telah terintegrasi.

1. Mempermudah pengelolaan dan pemantauan stok

Informasi stok barang dapat diakses secara *real-time*, sehingga karyawan dapat langsung mengetahui ketersediaan barang tanpa harus memeriksa secara manual ke gudang.

1. Mengurangi beban kerja

Tugas administratif seperti memproses permintaan barang atau menyusun laporan bulanan dapat dilakukan secara otomatis melalui sistem. Hal ini memberikan lebih banyak waktu bagi karyawan untuk fokus pada tugas-tugas starategis lainnya.

1. Meningkatkan akurasi dalam pemrosesan data

Dengan fitur otomatisasi, data yang diinputkan karyawan lebih minim kesalahan, terutana dalam proses pencatatan stok atau pemenuhan permintaan barang.

1. Meningkatkan produktivitas

Sistem yang cepat dan efisien memungkinkan karyawan untuk menyelesaikan tugas-tugas harian dengan waktu lebih singkat, sehingga produktivitas kerja meningkat.

1. Bagi Peneliti
2. Memberikan pengalaman dalam mengembangkan aplikasi berbasis *web* dengan implementasi algoritma *clustering*

Penelitian ini memberikan kesempatan bagi peneliti untuk mengembangkan keterampilan teknis dalam membangun sistem berbasis *web* yang menggunakan algoritma *K-Means Clustering*.

1. Memperluas wawasan terkait penerapan teknologi modern dalam manajemen inventori

Proyek ini memberikan pemahaman mendalam mengenai integrasi teknologi dalam pengelolaan inventori, termasuk cara mengatasi permasalahan operasional dengan solusi digital.

1. Memberikan kontribusi langsung kepada perusahaan tempat magang

Hasil penelitian ini tidak hanya menjadi pengalaman akademis tetapi juga berkontribusi nyata dalam memecahkan permasalahan di PT Semen Padang.

1. Bagi Dunia Akademik
2. Menambah referensi terkait pengembangan sistem berbasis *web* dengan penerapan algoritma *K-Means Clustering*

Penelitian ini dapat menjadi acuan bagi mahasiswa, akademisi, atau peneliti lain yang ingin mengembangkan sistem serupa dalam bidang manajemen inventori.

1. Mendorong penelitian lebih lanjut terkait penerapan teknologi dalam pengelolaan inventori

Penelitian ini membuka peluang untuk pengembangan lebih lanjut, seperti penerapan teknologi tambahan atau algoritma lain untuk meningkatkan efisiensi dan akurasi sistem manajemen inventori.

# BAB II LANDASAN TEORI

1. Pengertian Manajemen Inventori

Manajemen inventori adalah sebuah sistem yang digunakan untuk mengelola persediaan barang. Mengelola bagaimana mengklarifikasi dan menjaga akurasi catatan persediaan barang (Nurcahyawati et al., 2023). Menurut (Ray Silaen et al., 2024), inventori adaIah suatu sistem yang bertujuan untuk mengelola semua asset yang tersedia dalam jumlah yang optimal untuk menghindari kekurangan atau kelebihan. Inventori yang dikelola dengan baik memungkinkan perusahaan untuk menghindari risiko kelebihan stok (*overstock*) atau kekurangan stok (*stockout*), sehingga operasi perusahaan dapat berjalan dengan efisien dan lancar.

Dalam konteks industri besar seperti PT Semen Padang, pengelolaan inventori yang efisien sangat penting untuk memastikan kelancaran produksi dan distribusi barang di area operasional perusahaan. Sistem manual yang digunakan saat ini, seperti pencatatan menggunakan *Microsoft excel*, memiliki berbagai keterbatasan yang berdampak pada akurasi data dan efisiensi kerja. Untuk meningkatkan efisiensi dalam manajemen inventori, banyak perusahaan beralih ke sistem informasi berbasis *web* dan *mobile*. Misalnya, (Ramadhan & Valentino, 2024) mengembangkan sistem manajemen persediaan berbasis *web* yang membantu perusahaan mengoptimalkan pengelolaan stok, mengurangi kesalahan pencatatan, dan meningkatkan efisiensi operasional. Sistem ini memungkinkan pemantauan stok secara *real-time*, sehingga pengambilan keputusan dapat dilakukan cepat dan tepat. Selain itu, (Saputri & Hirzan, 2024) merancang aplikasi manajemen inventori berbasis *mobile* menggunakan *Flutter* dan *Firebase Realtime Database*, yang dapat memangkas waktu pengecekan stok barang dan meningkatkan mutu pelayanan.

Dengan demikian, penerapan teknologi informasi dalam manajemen inventori dapat meningkatkan efisiensi operasional dan akurasi data, yang sangat penting bagi perusahaan seperti PT Semen Padang dalam menghadapi tantangan operasional sehari-hari.

Proses manual menggunakan *Excel* sering kali menyebabkan beberapa masalah berikut:

1. Kurangnya akurasi data

Ketika karyawan memperbarui data stok barang secara manual, risiko kesalahan input data meningkat, terutama jika jumlah barang dalam skala besar.

1. Ketidaksesuaian informasi antar pengguna

Ketika file *Excel* diperbarui oleh satu pengguna, perubahan tersebut tidak otomatis terlihat oleh pengguna lain yang menggunakan file serupa. Hal ini menyebabkan ketidaksesuaian data stok antar departemen.

1. Keterbatasan aksesibilitas

File *Excel* harus diakses pada perangkat tertentu, sehingga karyawan tidak dapat memantau data stok secara *real-time* atau dari lokasi yang berbeda. Hal ini memperlambat pengambilan keputusan, terutama ketika terjadi permintaan barang mendesak.

Untuk mengatasi permasalahan ini, sistem manajemen inventori berbasis teknologi perlu diterapkan. Sistem tersebut tidak hanya dapat mengintergasikan data inventori secara otomatis, tetapi juga menyediakan akses data *real-time* yang dapat digunakan oleh berbagai departemen secara bersamaan. Dengan demikian, perusahaan dapat meningkatkan akurasi data, mempercepat proses pengambilan keputusan, dan mengoptimalkan ketersediaan barang di seluruh gudang,

1. Sistem Informasi Berbasis *Web*

Sistem inventori berbasis *web* adalah aplikasi yang dirancang untuk di jalankan melalui *browser* *web*, memungkinkan akses data secara *real-time* dari berbagai lokasi dan perangkat. Menurut (Fortuna Alfarisi et al., 2023) pada Jurnal JITT, sistem ini menawarkan keunggulan berupa aksesibilitas, skalibilitas, dan kemudahan integrasi dengan teknologi lainnya.

Keunggulan sistem berbasis *web* meliputi:

1. *Real-time* *Monitoring*: Data dapat diperbarui dan diakses secara langsung.
2. *Multi-user Access*: Mendukung penggunaan oleh banyak pengguna secara simultan.
3. Efisiensi Proses: Mengurangi kebutuhan dokumentasi manual.

(Samsudin et al., 2024) juga menyatakan bahwa sistem berbasis *web* dapat mengurangi waktu proses manual hingga 25% meningkatkan akurasi data, serta memperbaiki pengambilan keputusan opersional.

1. Data *Clustering* dan Algoritma *K-Means*
2. Pengertian *Clustering*

*Clustering* adalah suatu cara menganalisa dengan mengelompokkan ke dalam kelompok kesamaan tertentu. *Clustering* mengacu pada pengelompokan *record*, mengamati dan membuat kelas objek yang memiliki kesamaan (Sari et al., 2023). Menurut (Putri & Budayawan, 2024), *clustering* digunakan untuk menganalisis data dalam jumlah besar dengan cara menemukan pola-pola tersembunyi.

1. Algoritma *K-Means*

Algoritma *K-Means* adalah salah satu metode *clustering* yang paling popular dan digunakan dalam analisis data. Menurut (Ahmed et al., 2020), Algoritma *K-Means* dianggap sebagai salah satu algoritma *data mining* yang paling kuat dan popular dalam komunitas penelitian.

Agoritma ini digunakan untuk mengelompokkan data berdasarkan kesamaan sehingga dapat membantu mengidentifikasi pola dan karakteristik tertentu yang mungkin tersembunyi dalam dataset (Permadi & Wiyaja, 2023). Dalam sistem manajemen inventori, algoritma *K-Means* dapat diterapkan untuk mengelompokkan pola permintaan barang berdasarkan data historis, seperti jumlah barang yang diminta, frekuensi permintaan, atau waktu permintaan.

Berikut adalah langkah-langkah algoritma *K-Means* yang telah disesuaikan untuk data barang di PT Semen Padang:

1. Menentukan jumlah *cluster* (k).

Jumlah *cluster* ditentukan berdasarkan kebutuhan analisis. Dalam konteks permintaan barang, *cluster* dapat diatur menjadi tiga kategori:

1. *Cluster* 1 untuk barang dengan permintaan rendah dengan kuantitas permintaan kecil dari lima.
2. *Cluster* 2 untuk barang dengan permintaan sedang dengan kuantitas permintaan antara lima sampai dengan sepuluh.
3. *Cluster* 3 untuk barang dengan permintaan tinggi dengan kuantitas permintaan diatas sepuluh.
4. Memillih *centroid* awal secara acak.

*Centroid* awal dipilih dari data permintaan barang yang sudah ada. Sebagai contoh, *centroid* awal untuk setiap *cluster* dapat berupa rata-rata permintaan dari historis sebelumnya.

1. Mengelompokkan data berdasarkan jarak terdekat ke *centroid*.

Setiap data barang akan dihitung jaraknya ke *centroid* menggunakan rumus *Euclidean Distance*. Data kemudian dimasukkan ke *cluster* dengan *centroid* terdekat.

1. Memperbarui posisi *centroid* berdasarkan rata-rata data dalam *cluster*.

*Centroid* baru dihitung sebagai rata-rata dari semua dalam satu *cluster*. Misalnya, jika *cluster* permintaan rendah berisi data dengan kuantiras [3, 4, 2], maka *centroid* baru dihitung sebagai rata-rata:

**Centroid Baru**

1. Mengulangi proses hingga *centroid* tidak berubah.

Langkah 3 dan 4 diulang hingga posisi *centroid* stabil atau perubahannya sangat kecil (*konvergen*). Pada titik ini, *cluster* telah terbentuk secara optimal.

1. Menganalisis hasil *clustering*

Hasil *clustering* digunakan untuk mengidentifikasi barang-barang dengan permintaan tinggi, sedang, dan rendah. Informasi ini dapat dimanfaatkan untuk merencanakan stok prioritas dan strategi pengadaan.

Penelitian oleh (Andi Syahrul Ramdana et al., 2024) menunjukkan bahwa algoritma *K-Means* dapat mencapai akurasi hingga 70% dalam pengelompokkan pola permintaan barang.

Keunggulan *K-Means* dibandingkan dengan algoritma *clustering* lainnya, seperti:

1. *DBSCAN (Destiny-Based Spatial Clustering of Application with Noise)*

Kelebihan *DBSCAN* yaitu mampu mendeteksi *cluster* dengan bentuk arbiter dan mengabaikan data *outliner*.Sementara ekurangan *DBSCAN* dalam hal ini adalah tidak cocok untuk data inventori yang memiliki *cluster* berukuran serupa, karena algoritma ini lebih focus pada kepadatan data. Alasan mengapa algoritma ini tidak dipilih dalam kasus PT Semen Padang karena data permintaan barang cenderung berbentuk terstruktur dan berukuran serupa, sehingga *K-Means* lebih sesuai untuk mengelompokkan data berdasarkan kuantitas.

1. *Hierarchial Clustering*

Agoritma ini memiliki kelebihan dapat menghasilkan dendogram yang memberikan pandangan visual terkait hierarki *cluster*. Sementara itu kekurangannya adalah tidak efisien untuk data berukuran besar karena memiliki kompleksitas komputasi tinggi. Alasan algoritma ini tidak dipilih karena data permintaan barang di PT Semen Padang yang besar akan terus berkembang membuat *K-Means* lebih efisien dibandingkan pendekatan hierarki.

Jadi alasan mengapa *K-Means* lebih relevan untuk digunakan dalam pengembangan ini diantaranya adalah :

1. *K-Means* memiliki kompleksitas komputasi rendah, sehingga dapat menangani data permintaan barang yang besar dalam waktu singkat.
2. Algoritma ini mudah diimplementasikan dalam sistem berbasis *web* dengan bahasa pemrogramann modern seperti *Javascript* atau *Python*.
3. Data barang yang memiliki pola permintaan dapat dengan mudah dikelompokan berdasarkan kategori yang relevan, seperti permintaan rendah, sedang, dan tinggi.
4. *K-Means* sangat cocok untuk data inventori yang memiliki pola distribusi sederhana dan jumlah *cluster* telah ditentukan sebelumnya.
5. Penerapan *K-Means* pada Sistem Manajemen Inventori

Algoritma *K-Means Clustering* adalah sebuah teknik yang digunakan untuk mengelompokkan data ke dalam beberapa kelompok berdasarkan jarak, kriteria, kondisi, atau karakteristik tertentu (Muhamad Rizki, 2023). Dalam konteks manajemen inventori, algoritma ini dapat membantu mengelompokkan barang berdasarkan pola permintaan historis, sehingga dapat mendukung stategi pengadaan barang yang lebih efisien dan tepat sasaran.

Beberapa penelitian sebelumnya menunjukkan bahwa penerapan algoritma *K-Means* dapat meningkatkan efisiensi dalam pengelolaan inventori. Misalnya penelitian oleh (Andi Syahrul Ramdana et al., 2024) menunjukkan bahwa penggunaan algoritma *K-Means* dalam manajemen persediaan perpustakaan dapat meningkatkan akurasi klasifikasi barang hingga 70%, sehingga mempermudah pengelolaan stok berdasarkan kategori penggunaan.

Penelitian lain oleh (Nurdiyansyah & Akbar, 2021) menemukan bahwa algoritma *K-Means* mampu mengoptimalkan manajemen inventori pada *Poultry Shop* dengan mengelompokkan barang berdasarkan tingkat permintaan, yang berujung pada efisiensi pengadaan stok dan pengurangan risiko *overstock* maupun *understock*.Namun, penerapan algoritma *K-Means* dalam sistem manajemen inventori di PT Semen Padang memiliki tantangan dan karakteristik yang berbeda dibandingkan studi-studi sebelumnya. PT Semen Padang mengelola ribuan jenis barang di berbagai gudang yang tersebar di beberapa area operasional, seperti Indarung 4, 5, dan 6. Pola permintaan barang di setiap gudang juga bervariasi tergantung pada kebutuhan operasional di masing-masing area. Misalnya, gudang di *Raw Mill* memiliki permintaan tinggi untuk suku cadang mesin penggiling, sedangkan gudang di *Cement Mill* lebih banyak membutuhkan komponen kelistrikan.

Dengan menerapkan *K-Means Clustering*, sistem ini dapat mengelompokkan barang berdasarkan pola permintaan historis yang telah tercatat dalam *database* requests. Misalnya, jika suatu barang memiliki tingkat permintaan yang tinggi dalam periode tertentu, sistem dapat secara otomatis mengklasifikasikannya ke dalam kategori *High Demand*. Sebaliknya, barang dengan permintaan jarang atau tidak konsisten dapat dimasukkan ke dalam kategori *Low Demand*.

Dalam implementasi sistem ini di PT Semen Padang, algoritma *K-Means* akan bekerja dengan tahapan berikut:

1. Pengumpulan Data

Data permintaan barang dikumpulkan dari tabel *requests* yang mencatat transaksi selama beberapa bulan terakhir. Data ini mencakup atribut seperti kode barang, jumlah permintaan, waktu permintaan, dan lokasi gudang.

1. *Preprocessing* Data

Membersihkan data dari anomaly (permintaan duplikat atau permintaan yang dibatalkan), kemudian menormalisasikan data agar skala permintaan lebih seragam.

1. Penentuan Jumlah Cluster (k)

Berdasarkan analisis eksploratif data historis, jumlah cluster akan ditentukan dengan metode *Elbow* *Method*. Misalnya, barang dapat diklasifikasikan ke dalam tiga cluster utama:

1. *Cluster* 1 (*Low Demand*), yang merupakan permintaan kurang dari 5 unit per bulan.
2. *Cluster* 2 *(Medium Demand*), yang merupakan permintaan antara 5-10 unit per bulan.
3. *Cluster* 3 (*High Demand*), yang merupakan permintaan lebih dari 10 unit per bulan.
4. Penerapan Algoritma *K-Means*

Data permintaan barang diproses menggunakan *K-Means* untuk menemukan pola kelompok barang berdasarkan tingkat permintaan di setiap area.

1. Visualisasi Hasil *Clustering*

Hasil pengelompokan barang akan divisualisasikan dalam bentuk grafik *pie* atau *bar* *chart* menggunakan *ApexCharts*, kemudian *dashboard* admin akan menampilkan daftar barang beserta *cluster*-nya untuk membantu dalam proses pengambilan keputusan pengadaan barang.

Dengan adanya penerapan algoritma *K-Means* ini, PT Semen Padang dapat memperoleh manfaat sebagai berikut:

1. Optimalisasi pengadaan barang

Barang dengan permintaan tinggi dapat segera diprioritaskan dalam perencanaan stok, sehingga mengurangi risiko kekurangan stok pada saat dibutuhkan. Efisiensi penyimpanan gudang Barang dengan kategori permintaan rendah dapat disimpan dalam jumlah lebih sedikit untuk menghindari *overstock* yang memakan kapasitas gudang.

1. Dukungan keputusan berbasis data

Manajemen dapat menggunakan hasil *clustering* untuk menentukan strategi inventori, seperti kapan harus melakukan pengadaan atau kapan harus mengurangi stok barang tertentu.

1. Peningkatan efisiensi operasional

Dengan mnengetahui pola permintaan barang berdasarkan data historis, proses persetujuan dan dsitribusi barang dapat lebih cepat dan tepat sasaran.

Dengan demikian, penerapan *K-Means Clustering* dalam sistem manajemen inventori PT Semen Padang dapat membantu perusahaan dalam mengelola stok barang secara lebih efisien, mengurangi risiko kesalahan pengadaan, serta meningkatkan efektivitas pengelolaan gudang di setiap area operasional.

1. Metode Pengembangan

Proyek ini menggunakan metodologi pengembangan *Agile* karena fleksibilitas dan iterasi yang memungkinkan proyek berkembang sesuai kebutuhan. Metode pengembangan *agile* adalah pendekatan dalam pengembangan perangkat lunak yang menekankan kolaborasi tim, fleksibilitas, dan respon cepat terhadap perubahan kebutuhan. Pendekatan ini melibatkan iterasi dan inkrementasi, dimana produk dikembangkan dalam siklus pendek yang memungkinkan penyesuaian berkelanjutan sesuai umpan balik pengguna dan perubahan kondisi pasar.

Dalam jurnal “*Agility is responsiveness to change: An essential defenition”*, oleh (Gren & Lenberg, 2020) menyatakan bahwa inti dari pendekatan *Agile* adalah kemampuan untuk merespons perubahan secara cepat dan efektif. Mereka menyoroti bahwa prinsip-prinsip *Agile*, seperti kolaborasi tim dan adaptasi terhadap perubahan, berfokus pada peningkatan responsivitas terhadap perubahan kebutuhan dan kondisi.

Berikut adalah alasan utama memilih metode *Agile* untuk proyek ini:

1. Fleksibilitas dalam Menangani Perubahan Kebutuhan Dalam proyek ini, kebutuhan pengguna dan fitur sistem dapat berkembang berdasarkan umpan balik. *Agile* memungkinkan perubahan tersebut diakomodasi pada setiap iterasi tanpa memengaruhi keseluruhan proyek.
2. Pendekatan Iteratif dan Inkremental Dengan *Agile*, pengembangan dilakukan dalam siklus pendek yang disebut sprint, sehingga bagian-bagian sistem dapat selesai dan diuji secara bertahap. Hal ini meminimalkan risiko kesalahan besar.
3. Umpan Balik Pengguna yang Cepat *Agile* mendorong keterlibatan aktif pengguna selama proses pengembangan. Dalam proyek ini, staf PT Semen Padang dapat memberikan masukan langsung pada setiap iterasi untuk memastikan solusi yang dihasilkan sesuai kebutuhan operasional mereka.
4. Pengujian Berkelanjutan Pengujian dilakukan di setiap tahap iterasi, memastikan bahwa fitur yang dikembangkan dapat langsung berfungsi dengan baik sebelum melanjutkan ke tahap berikutnya.
5. Kolaborasi Tim yang Efisien *Agile* memfasilitasi komunikasi dan koordinasi antar anggota tim, termasuk pengembang *frontend*, *backend*, dan analis sistem, untuk memastikan semua komponen sistem terintegrasi dengan baik.

Dengan keunggulan-keunggulan ini, *Agile* menjadi pilihan yang tepat untuk proyek pengembangan sistem manajemen inventori yang kompleks, dinamis, dan membutuhkan penyesuaian berkelanjutan.

1. Teknologi Pendukung

Sistem yang dirancang untuk PT Semen Padang menggunakan berbagai teknologi modern untuk memastikan efisiensi, kemudahan penggunaan, serta skalabilitas dalam pengelolaan inventori. Berikut adalah teknologi yang digunakan dan alasan pemilihannya:

1. *Frontend*
2. *HTML, CSS, dan Javascript*

Digunakan untuk membangun tampilan antarmuka sistem agar responsif dan mudah digunakan oleh berbagai pengguna.

1. *SweetAlert2*

*Library* *Javascript* yang digunakan untuk menampilkan *pop-up* interaktif saat pengguna melakukan aksi penting, seperti permintaan barang atau konfirmasi tindakan. Alasan pemilihannya adalah dibandingkan *alert* bawaan *browser* atau *library* lain seperti *Toastr*, *SweetAlert2* lebih fleksibel, memiliki desain modern, serta mendukung berbagai jenis notifikasi yang interaktif dan dapat dikontumisasi sesuai kebutuhan sistem.

1. *Backend*
2. *Node.js dan Express*

Digunakan sebagai *backend* untuk menangani permintaan *API* dengan cepat dan efisien.

1. *MySQL*

*Database* relasional yang digunakan untuk menyimpan data inventori dan transaksi permintaan barang secara terstruktur. *MySQL* dipilih karena skalabilitasnya yang tinggi dan kompabilitasnya dengan sistem berbasis *web*

1. Visualisasi Data

Terdapat salah satu *library* *Javascript* yaitu *ApexCharts* antuk menampilkan data dalam bentuk grafik interaktif, seperti diagram batang dan *pie chart* untuk memvisualisasikan pola permintaan barang berdasarkan hasil *clustering*. Beberapa alasan memilih *library* ini diantaranya :

1. Dibandingkan dengan *Chart.js* atau *D3.js*, *ApexCharts* lebih ringan dan memiliki fitur bawaan yang lebih mendukung visualisasi realtime tanpa perlu banyak konfigurasi tambahan.
2. Mendukung animasi dinamis dan responsif, sehingga pengguna dapat memahami tren permintaan barang secara lebih intuitif.
3. Memiliki fitur *tooltip* interaktif yang mempermudah pengguna dalam membaca data tanpa perlu melakukan klik tambahan.
4. Pengolahan dan Ekspor Data
5. *XLSX.js*

Digunakan untuk membaca dan mengekspor data inventori dalam format *spreadsheet* (*Excel*), memungkinkan pengguna mengunduh dan mengelola data dengan lebih fleksibel.

1. *jsPDF*

Digunakan untuk menghasilkan laporan dalam bentuk *PDF* secara langsung dari sistem. Dibandingkan dengan metode ekspor manual menggunakan *browser*, *jsPDF* memungkinkan pengguna mengunduh laporan tanpa memerlukan *software* tambahan, serta memiliki kontrol penuh atas format dokumen yang dihasilkan.

1. *Version Control* dan *API Testing*
2. *Git dan GitHub*

Digunakan untuk *version control* agar tim pengembang dapat mengelola perubahan kode dengan baik dan mencegah kesalahan dalam pengembangan sistem.

1. *Postman*

Digunakan untuk menguji *API* sebelum diimplementasikan ke dalam sistem *frontend*.

Berdasarkan penelitian oleh (Fachrul Rezy & Ikasari, 2023), pada jurnal BIIKMA, mencatat bahwa integrasi teknologi ini dapat meningkatkan aksesibilitas dan efisiensi pengelolaan inventori di perusahaan besar.

1. Kerangka Teori Penelitian

Kerangka teori penelitian ini menggambarkan hubungan antara manajjemen inventori, sistem berbasis *web*, dan algoritma *K-Means*. Diagram berikut menunjukkan alur implementasi:

1. Data historis permintaan barang, yaitu data yang dikumpulkan dari laporan permintaan sebelumnya.
2. Proses *clustering*, yang mana algoritma *K-Means* diterapkan untuk mengelompokkan pola permintaan barang.
3. Pengambilan keputusan, yang nantinya hasil dari *clustering* digunakan untuk merancang startegi pengelolaan stok dan pengadaan barang.

Pendekatan ini di dukung oleh penelitian oleh (Andi Syahrul Ramdana et al., 2024), yang menunjukkan bahwa penerapan algoritma *K-Means* dalam manajemen persediaan di perpustakaan dapat meningkatkan efisiensi pengelolaan dengan mengelompokkan item berdasarkan pola permintaan. Implementasi ini juga memberikan manfaat berupa penghematan waktu dan biaya.

1. Penelitian Relevan

Berikut adalah penelitian yang relevan:

Tabel. 1 Penelitian yang Relevan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Judul** | **Peneliti** | **Permasalahan** | **Metode** | **Hasil** |
| 1 | OPTIMALISASI STOK BARANG MELALUI ALGORITMA *K-MEANS CLUSTERING* ANALISIS UNTUK MANAJEMEN PERSEDIAAN DALAM KONTEKS BISNIS MODERN | Martanto, M. (2024) | Bagaimana mengoptimalkan stok barang dengan menggunakan algoritma *clustering* agar lebih efisien dalam manajemen persediaan. | Algoritma *K-Means Clustering* diterapkan pada data persediaan barang untuk mengelompokkan pola permintaan. | Penerapan *K-Means Clustering* dapat meningkatkan efisiensi dalam pengelolaan stok, mengurangi kelebihan persediaan, dan memastikan ketersediaan barang sesuai kebutuhan. |
| 2 | PENERAPAN ALGORITMA *K-MEANS* UNTUK MANAJEMEN PERSEDIAAN DI PERPUSTAKAAN | Ramdana, A. S., & Pramono, E. (2024) | Manajemen persediaan di perpustakaan sering mengalami kendala dalam menentukan kategori buku yang memiliki permintaan tinggi. | Algoritma *K-Means Clustering* digunakan untuk mengelompokkan buku berdasarkan tingkat peminjaman. | Algoritma *K-Means* membantu menentukan buku dengan permintaan tinggi, sehingga manajemen perpustakaan dapat lebih efisien dalam pengadaan buku. |
| 3 | Implementasi Algoritma *K-Means* untuk Menentukan Persediaan Barang pada *Poultry Shop* | Nurdiyansyah, F., & Akbar, I. (2021) | Kesulitan dalam menentukan jumlah persediaan barang di toko unggas (*poultry shop*) karena pola permintaan yang bervariasi. | Algoritma *K-Means Clustering* diterapkan pada data historis penjualan. | Dengan i, pemilik toko dapat mengoptimalkan stok barang berdasarkan kategori permintaan tinggi, sedang, dan rendah. |
| 4 | *Systematic Literature Review:* Sistem Informasi Manajemen Inventori Barang Berbasis *Web* | Rezy, A. F., & Ikasari, I. H. (2023) | Kurangnya kajian literatur terkait penerapan sistem informasi inventori berbasis *web* yang optimal. | Metode *Systematic Literature Review (SLR)* untuk menganalisis berbagai penelitian terkait sistem inventori. | Studi menunjukkan bahwa sistem informasi berbasis *web* dapat meningkatkan efisiensi dan akurasi dalam manajemen persediaan. |
| 5 | Implementasi Algoritma *K-Means* Menggunakan Metode *Clustering* Untuk Menentukan Penjualan Produk Laris Dan Tidak Laris Di Grosir Chintiya | Putri, A. Y., Syafrijon, & Budayawan, K. (2024) | Pemilik toko grosir mengalami kesulitan dalam mengidentifikasi produk dengan tingkat penjualan tinggi dan rendah. | Algoritma *K-Means Clustering* diterapkan untuk mengelompokkan produk berdasarkan data historis penjualan. | Hasil *clustering* membantu pemilik toko membuat strategi pemasaran dan pengadaan stok yang lebih efektif. |

# BAB III ANALISIS DAN PERANCANGAN SISTEM

1. Analisis Kebutuhan

Analisis kebutuhan meliputi kebutuhan fungsional dan non-fungsional dari sistem yang dikembangkan, termasuk juga identifikasi aktor, dan diagram konteks dari sistem.

1. Kebutuhan Fungsional

Sistem yang dikembangkan bertujuan untuk memenuhi kebutuhan berikut :

1. Pengelolaan data barang, admin dapat mengelola data barang di gudang setiap area secara terpusat.
2. Impor data barang melalui file *excel* oleh admin, dan juga melakukan ekspor laporan permintaan barang bulanan.
3. Persetujuan permintaan barang gudang di area secara otomatis, sistem yang memungkinkan pengguna untuk mengajukan permintaan barang secara otomatis.
4. *Clustering* pola permintaan, algoritma *K-Means* diterapkan untuk mengelompokokan pola permintaan barang berdasarkan data historis.
5. Pemantauan *real-time*, jadi pengguna dapat memantau stok barang dan proses permintaan secara *real-time*.
6. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional dalam sistem ini mencakup aspek performa, skalabilitas, keamanan, dan kompabilitas guna memastika sistem dapat berjalan dengan optimal di lingkungan PT Semen Padang. Berikut adalah rincian kebutuhan non-fungsional:

1. Waktu respon maksimal (<5 detik)

Sistem harus mampu menampilkan data inventori dan hasil pencarian barang dalam waktu kurang dari 5 detik untuk meningkatkan efisiensi kerja pengguna. Kemudian sistem permintaan data, baik pencarian stok barang, permintaan barang, atau hasil *clustering*, harus diproses dalam waktu yang optimal agar tidak menghambat operasional gudang.

1. Kompabilitas perangkat dan *browser*

Sistem harus dapat berjalan pada berbagai perangkat, termasuk *desktop*, laptop, tablet, dan *smartphone* tanpa mengalami kendala tampilan atau fungsionalitas. Sistem juga harus kompatibel dengan *browser* popular seperti *Google Chrome, Mozila Firefox, Microsoft Edge, dan Safari.*

1. Keamanan sistem dan data

Data pengguna dan inventori harus disimpan dengan enkripsi untuk mencegah kebocoran atau akses tidak sah. Sistem juga harus memiliki mekanisme autentikasi berbasis sesi untuk memastikan hanya pengguna yang memiliki akses yang dapat menggunakan fitur-fitur sistem. Kemudian, setiap transaksi yang dilakukan dalam sistem harus tercatat dalam log aktivitas guna memudahkan audit dan pemantauan aktivitas pengguna.

1. Ketersediaan dan keandalan sistem

Sistem harus dapat memastikan ketersediaan layanan selama jam operasional perusahaan

1. Efisiensi pengolahan data

Algoaritma *K-Means Clustering* yang diterapkan harus dapat mengelompokkan pola permintaan barang dalam waktu kurang dari 10 detik, bahkan untuk dataset besar, dan juga proses ekspor data ke *Excel* atau *PDF* harus dapat dilakukan dalam waktu maksimal 5 detik per 1000 data barang.

Dengan adanya spesifikasi ini, sistem diharapkan dapat berjalan dengan efisien, aman dan dapat diakses dengan baik oleh seluruh pengguna yang berperan dalam pengelolaan inventori di PT Semen Padang.

1. Identifikasi Aktor
2. *User*

Melihat ketersediaan barang, melakukan pencarian barang yang dibutuhkan, dan melakukan permintaan ke gudang daerah, kemudian jika barang di gudang daerah tidak tersedia, maka permintaan barang dialihkan ke gudang pusat.

1. Admin Daerah

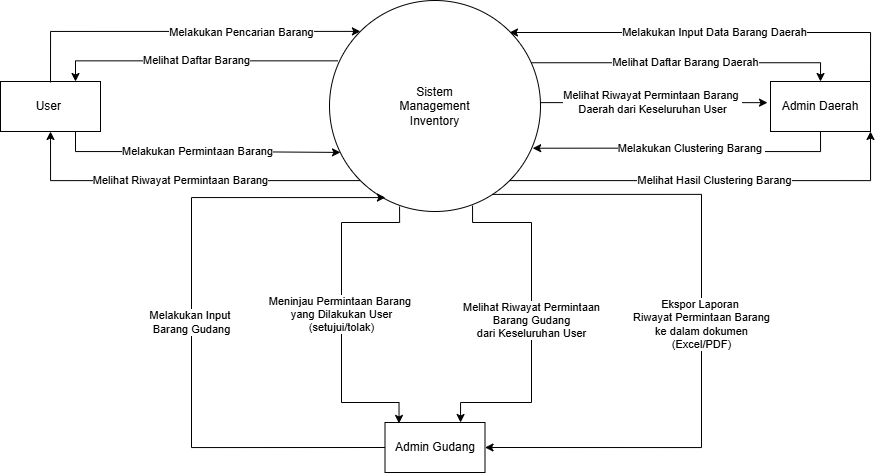
Bertanggung jawab untuk mengelola barang pada gudang di setiap area yang ada di setiap daerah, memantau permintaan dari *user* dan juga memantau persediaan barang.

1. Admin Gudang

Bertanggung jawab untuk memantaau dan mengelola persediaan barang di gudang pusat yang berada di luar kawasan pabrik, dan juga menangani permintaan *user* ke gudang (melakukan persetujuan atau penolakan untuk permintaan tersebut)

1. Diagram Konteks

Berdasarkan dari identifikasi aktor sebelumnya berikut adalah diagram konteks dari sistem ini:

a

Gambar. *Context Diagram*

1. Desain Sistem
2. Arsitektur Sistem

Sistem ini dirancang menggunakan arsitektur *client-server* berbasis *web*, komponen utama yang terdapat dalam sistem ini diantaranya:

1. *Frontend*

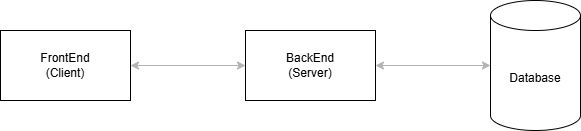
Dibangun dengan *HTML, CSS, Javascript* (*Apexcharts* untuk visualisasi data).

1. *Backend*

Menggunakan *Node.js* dan *express* sebagai framework.

1. *Database*

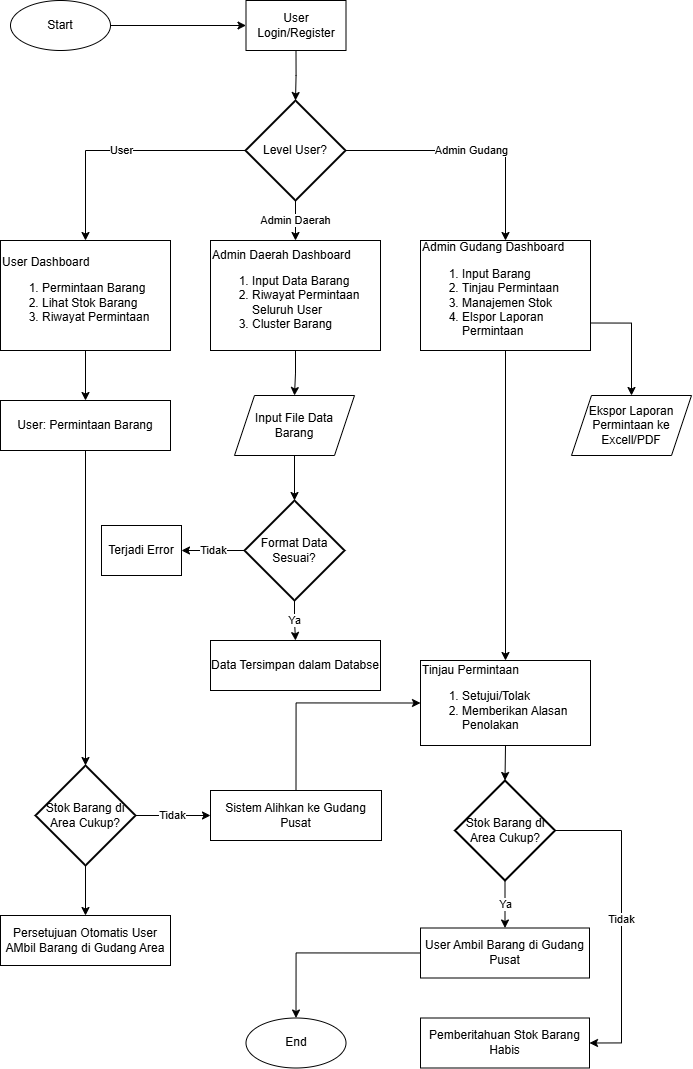
*MySQL* sebagai sistem penyimpanan data.



Gambar. Arsitektur Sistem

1. Diagram Alur Sistem

Diagram alur sistem menggambarkan bagaimana pengguna berinteraksi dengan sistem dalam melakukan permintaan barang, bagaimana sistem memproses permintaan tersebut, serta bagaimana sistem mengelola data inventori. Berikut adalah tahapan utama dalam *flowchart*:



Gambar. 3 *Flowchart*

Berikut adalah penjelasan dari proses yang terdapat pada *flowchart* sebelumnya:

1. Pengguna melakukan *login*

Pengguna harus masuk ke dalam sistem dengan memasukkan kredensial berupa *email* dan *password*, selanjutnya sistem akan memverifikasi kredensial yang dimasukkan dengan data yang tersimpan di *database*, jika autentikasi berhasil, sistem akan mengenali peran pengguna berdasarkan level aksesnya.

1. Sistem mengidentifikasi level pengguna

Setelah *login* berhasil, sistem akan mengarahkan pengguna ke halaman utama sesuai dengan peran mereka, yang pertama itu ada *user* (pengguna biasa) yang memiliki akses untuk melihat stok barang dan mengajukan permintaan barang. Kemudian yang kedua ada admin daerah yang memiliki akses untuk mengelola stok barang di gudang daerah, dan yang terakhir ada admin gudang yang bertanggung jawab atas stok barang di gudang pusat dan menangani permintaan barang yang dialihkan dari gudang daerah.

1. *User* mengajukan permintaan barang.

*User* dapat mencari barang yang dibutuhkan menggunakan fitur pencarian atau filter barang. Setelah menemukan barang yang diinginkan, *user* mengisi jumlah barang yang akan diminta dan mengirimkan permintaan ke sistem.

1. Sistem mengecek ketersediaan stok di gudang daerah

Jika barang tersedia di gudang daerah, permintaan akan langsung di proses oleh sistem dan otomatis disetujui. Namun, jika barang tidak tersedia, permintaan akan dialihkan ke gudang pusat untuk ditangani oleh admin gudang.

1. Admin menyetujui atau menolah permintaan

Admin gudang akan menerima notifikasi terkait permintaan barang. Admin dapat menyetujui permintaan jika stok barang mencukupi atau menolak permintaan jika barang tidak tersedia. Jika permintaan disetujui, stok barang akan diperbarui secara otomatis dalam sistem, dan jika ditolak, sistem akan memberikan notifikasi kepada *user* terkait penolakan.

1. Sistem melakukan *clustering* pola permintaan barang

Sistem akan menganalisis data permintaan barang ke dalam beberapa *cluster* (permintaan rendah, sedang, dan tinggi). Hasil cluster ini akan divisualisasikan dalam bentuk grafik di *dashboard* untuk membantu pengambilan keputusan terkait pengadaan barang.

1. Sistem menampilkan hasil *clustering* dan laporan stok

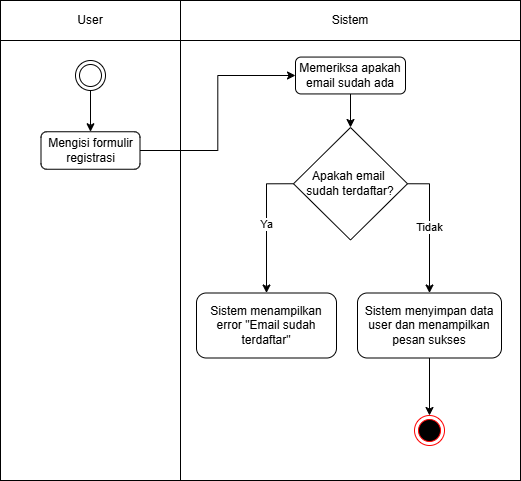
Admin dapat melihat laporan stok barang, riwayat permintaan, serta hasil *clustering* dalam bentuk grafik yang divisualisasikan menggunakan *ApexCharts*.

1. *Activity Diagram*

Dalam sistem manajemen inventori ini terdapat beberapa *Activity Diagram* sesuai dengan proses yang berjalan. Berikut adalah *Activity Diagram* dari sistem ini:

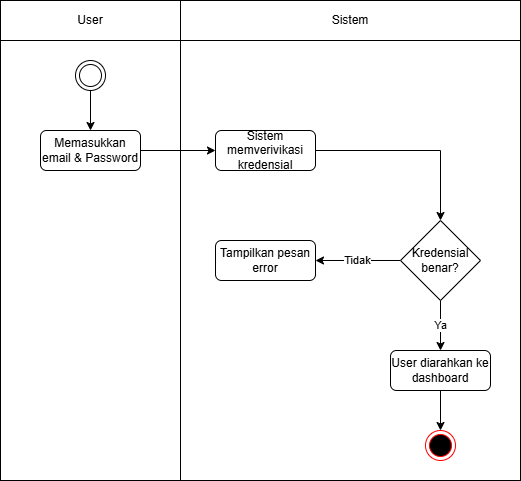
1. *Activity Diagram* Registrasi

Proses registrasi memungkinkan *user* baru untuk membuat akun dalam sistem manajemen inventori. *User* diminta untuk mengisi formulir pendaftaran yang berisi nama, email, dan *password*. Setelah data dikirim, sistem akan melakukan pengecekan apakah email yang dimasukkan sudah terdaftar dalam *database*. Jika email belum digunakan, maka sistem akan menyimpan data *user* dan menampilkan pesan sukses, Namun jika email sudah terdaftar, sistem akan menampilkan pesan *error* yang menyatakan bahwa email sudah digunakan.



Gambar. *Activity Diagram* Registrasi

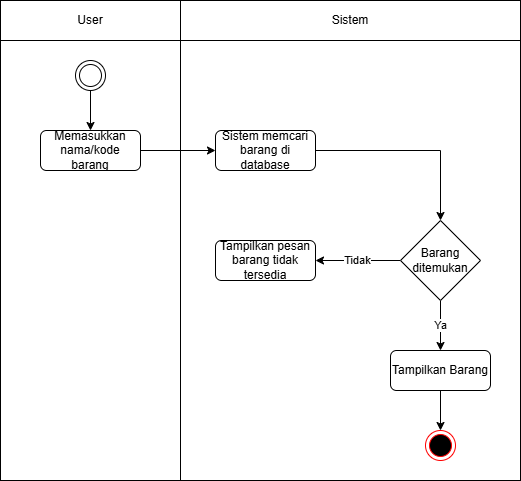
1. *Activity Diagram* *Login*



Gambar. *Activity Diagram Login*

Proses *login* dalam sistem manajemen inventori bertujuan untuk memastikan bahwa setiap pengguna memiliki hak akses yang sesuai. *User* memasukkan email dan *password* ke dalam sistem, kemudian sistem akan melakukan verifikasi terhadap kredensial yang diberikan. Jika email dan *password* yang dimasukkan sesuai dengan data yang tersimpan dalam *database*, maka *user* akan diarahkan ke *dashboard* sesuai dengan perannya (*User*, Admin Daerah, atau Admin Gudang). Namun, jika kredensial yang dimasukkan salah, sistem akan menampilkan pesan *error* dan meminta pengguna untuk mencoba kembali. Proses ini memastikan bahwa hanya pengguna yang terdaftar yang dapat mengakses sistem.

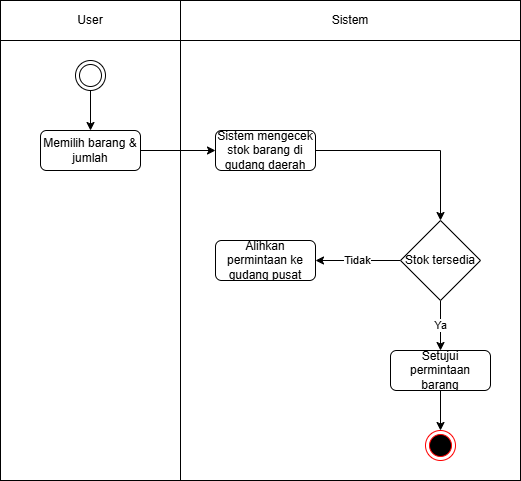
1. *Activity Diagram* Mencari Barang



Gambar. *Activity Diagram* Mencari Barang

Fitur pencarian barang memungkinkan *user* untuk menemukan barang yang tersedis dalam sistem berdasarkan nama atau kode barang. *User* memasukkan kata kunci pencarian, kemudian sistem akan melakukan pencocokan data dengan informasi barang yang tersimpan dalam *database*. Jika barang ditemukan, sistem akan menampilkan daftar barang yang sesuai dengan kata kunci pencarian. Namun, jika barang tidak ditemukan, sistem akan memberikan notifikasi bahwa barang tidak tersedia. Proses pencarian ini membantu *user* dalam menemukan barang yang dibutuhkan dengan cepat dan efisien.

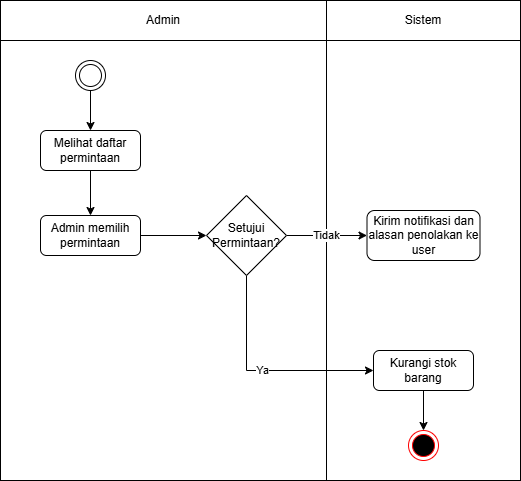
1. *Activity Diagram* Mengajukan Permintaan Barang



Gambar. *Activity Diagram* Mengajukan Permintaan Barang

Pada proses pengajuan permintaan barang, *user* memilih barang yang diinginkan dan memasukkan jumlah yang dibutuhkan, Sistem akan terlebih dahulu mengecek ketersediaan stok barang di gudang daerah. Jika sistem akan secara otomatis menyetujui permintaan barang. Namun, jika stok barang tidak tersedia, sistem akan secara otomatis mengalihkan permintaan ke gudang pusat agar dapat di proses lebih lanjut. Proses ini memungkinkan *user* untuk mengajukan permintaan barang dengan fleksibel, baik daru gudang daerah maupun gudang pusat.

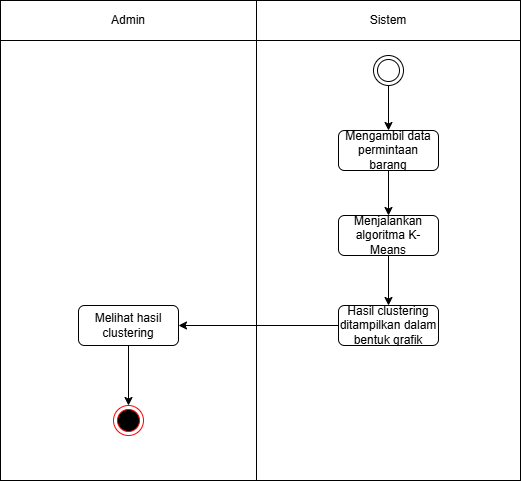
1. *Activity Diagram* Menyetujui atau Menolak Permintaan



Gambar. *Activity Diagram* Tinjau Permintaan Barang

Setelah *user* mengajukan permintaan barang, admin gudang bertanggung jawab untuk meninjau permintaan tersebut. Admin dapat melihat daftar permintaan yang masuk, kemudian memilih permintaan tertentu untuk di proses. Admin dapar memiliki dua opsi, yaitu menyetujui atau menolak permintaan, Jika permintaan barang disetujui, sistem akan memperbarui stok barang di gudang dan *user* akan menerima notifikasi bahwa permintaan telah disetujui. Jika permintaan ditolak, sistem akan memberikan notifikasi kepada *user* beserta alas an penolakan. Proses ini memastikan bahwa setiap permintaan barang ditinjau secara menyeluruh sebelum diproses lebih lanjut.

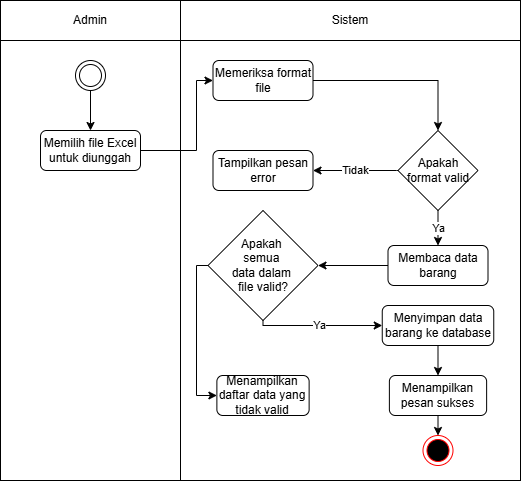
1. *Activity Diagram* *Clustering* Permintaan Barang



Gambar. *Activity Diagram* *Clustering* Permintaan Barang

Untuk mengoptimalkan pengelolaan inventori, sistem menggunakan algoritma *K-Means Clustering* untuk mengelompokkan pola permintaan barang berdasarkan jumlah dan frekuensi permintaan. Sistem mengambil data permintaan barang yang telah diajukan oleh *user*, kemudian menerapkan algoritma *clustering* untuk mengkategorikan barang dalam beberapa kelompok berdasarkan karakteristik tertentu. Hasil *clustering* akan divisualisasikan dalam bentuk grafik pada *dashboard* admin, sehingga dapat membantu pengambilan keputusan dalam pengelolaan stok barang. Dengan adanya proses *clustering* ini, admin dapat lebih mudah mengidentifikasi barang yang memiliki permintaan tinggi, sedang, dan rendah.

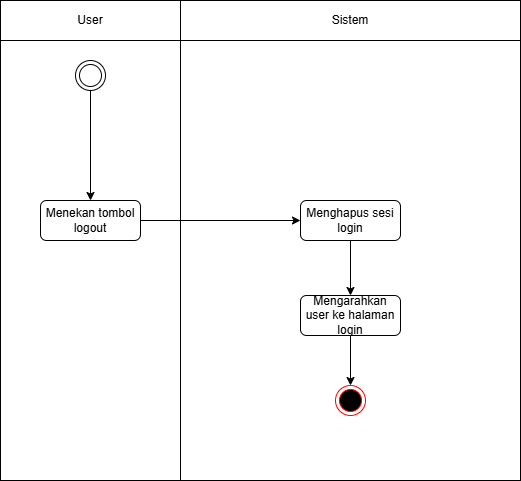
1. *Activity Diagram* Input Data Barang



Gambar. *Activity Diagram* Input Data Barang

Fitur input data barang melalui file *Excel* digunakan untuk mempercepat proses pengelolaan stok barang dalam jumlah besar. Admin mengunggah file *Excel* yang berisi daftar barang, kemudian sistem akan memeriksa apakah format file yang diunggah valid (misalnya .*xlsx* atau .*csv*). Jika format file salah, sistem akan menampilkan pesan *error*. Jika format benar, sistem akan membaca data dalam file dan melakukan validasi terhadap isi file. Jika ada kesalahan dalam data, sistem akan menampilkan daftar data yang tidak valid agar dapat diperbaiki oleh admin. Jika semua data valid, sistem akan menyimpan data barang ke dalam *database*, dan menampilkan pesan sukses.

1. *Activity Diagram* *Logout*



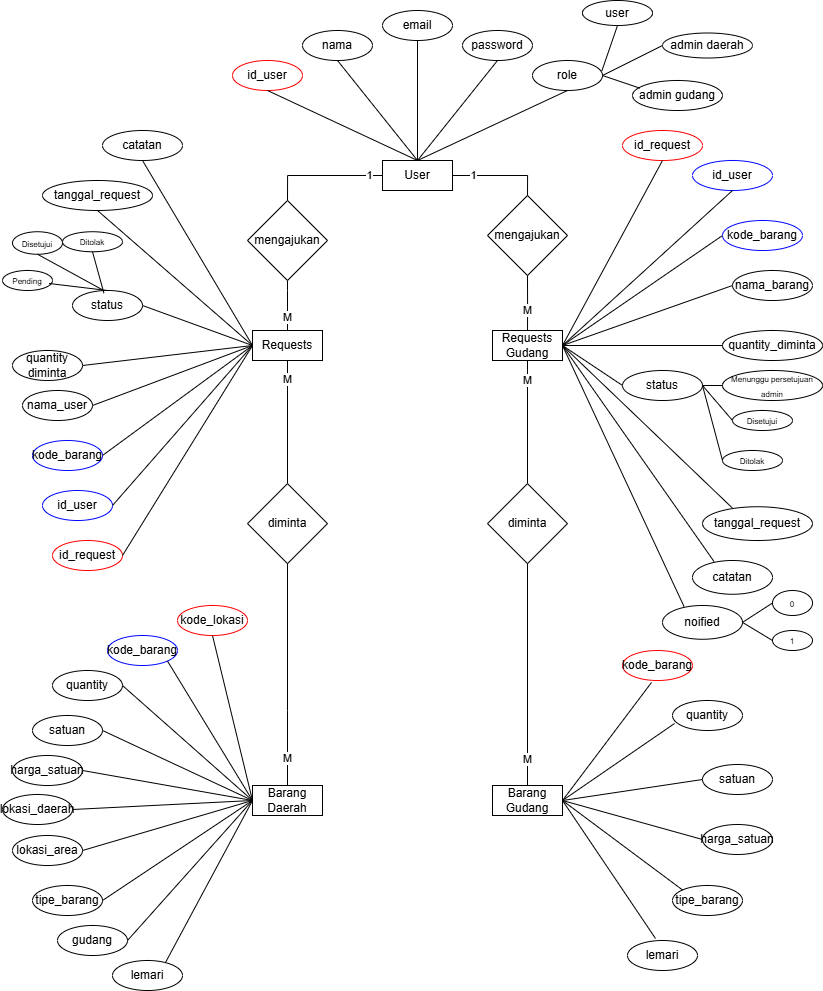
Gambar. *Activity Diagram Logout*

Fitur *logout* digunakan untuk mengahiri sesi *login* dalam sistem, Ketika *user* memilih untuk *logout*, sistem akan menghapus sesi autentikasi dan mengarahkan *user* kembali ke halaman *login*.

1. *Entity-Relationship Diagram (ERD)*

Sistem ini melibatkan beberapa entitas utama, yaitu *User*, Requests, Requests Gudang, Barang Daerah, dan Barang Gudang. Setiap entitas memiliki keterikatan dengan entitas lainnya. *User* dapat mengajukan banyak permintaan barang, baik ke gudang daerah maupun gudang pusat, sehingga hubungan antara *User* dan Requests bersifat *One to Many*. Selain itu, satu barang yang terdapat di gudang daerah maupun gudang pusat dapat diminta dalam berbagai requests, sehingga hubungan antara Barang dan Requests bersifat *Many to Many*.

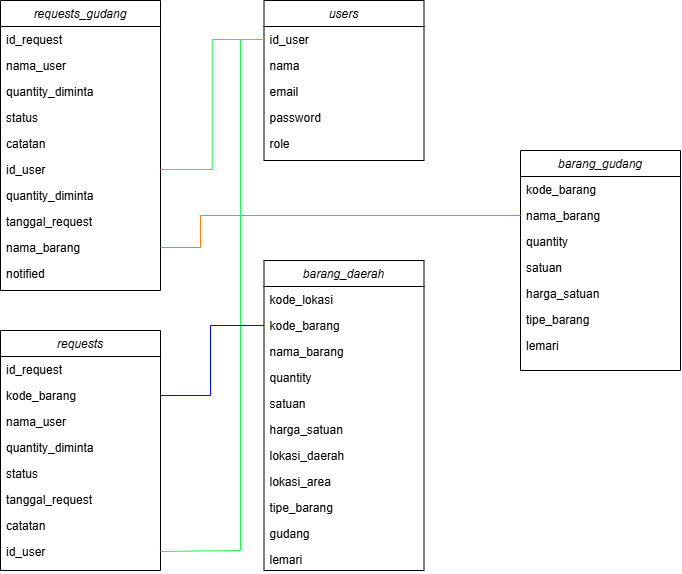
Berikut adalah *Entiry-Relationship Diagram(ERD)* dari sistem ini:



Gambar. 12 *Entity-Relationship Diagram (ERD)*

1. *Class Diagram*

Pada diagram ini terdapat relasi antar kelas pada masing masing table yang terhubung dengan *foreign key.*



Gambar. *Class Diagram*

Seperti pada tabel requests dan tabel requests gudang, terdapat id\_user yang merupakan foreign key yang berasal dari user\_id pada table user, kemudian juga kode\_barang yang juga merupakan foreign key yang berasal dari kode\_barang pada tabel barang\_daerah. Begitu juga nama\_barang pada tabel requests gudang yang merupakan foreign key yang berasal dari nama\_barang pada tabel barang\_gudang.

1. *Database Design*

Sistem ini menggunakan *database* relasional untuk mengelola data terkait inventori dan permintaan barang. Tabel utama yang digunakan dalam sistem ini meliputi barang\_daerah, barang\_gudang, Requests, requests\_gudang, dan *user*s.

1. Tabel barang\_daerah

Tabel ini menyimpan seluruh informasi mengenai barang yang tersimpan di gudang daerah. Data yang dicatat mencakup kode unik barang, nama barang, jumlah stok yang tersedia, satuan barang, serta informasi harga barang. Selain itu, table ini juga menyimpan lokasi spesifik barang dalam gudang, termasuk daerah penyimpanan, area tertentu dalam gudang, nomor gudang, dan nomor lemari penyimpanan untuk mempermudah pencarian barang.

Tabel. 2 Tabel Barang Daerah

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Atribut | Tipe Data | Ukuran | Keterangan |
| kode\_lokasi | *Varchar* | 255 | *Primary key* |
| kode\_barang | *Varchar* | 50 |  |
| nama\_barang | *Varchar* | 100 |  |
| quantity | *Integer* | 11 |  |
| satuan | *Varchar* | 20 |  |
| harga\_satuan | *Decimal* | 10 |  |
| lokasi\_daerah | *Varchar* | 50 |  |
| lokasi\_area | *Varchar* | 50 |  |
| tipe\_barang | *Varchar* | 50 |  |
| gudang | *Varchar* | 10 |  |
| lemari | *Varchar* | 10 |  |

1. Tabel barang\_gudang

Table ini menyimpan informasi barang yang tersedia di gudang. Data yang dicatat mencakup kode barang, serta lokasi penyimpanan di dalam gudang pusat. Informasi ini digunakan untuk mengelola barang secara efektif dan memastikan permintaan barang dapat dipenuhi dengan cepat.

Tabel. 3 Tabel Barang Gudang

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Atribut | Tipe Data | Ukuran | Keterangan |
| kode\_barang | *Varchar* | 50 | *Primary key* |
| nama\_barang | *Varchar* | 100 |  |
| quantity | *Integer* | 11 |  |
| satuan | *Varchar* | 20 |  |
| harga\_satuan | *Decimal* | 10 |  |
| tipe\_barang | *Varchar* | 50 |  |
| lemari | *Varchar* | 50 |  |

1. Tabel requests

Table ini mencatat seluruh permintaan barang yang diajukan oleh pengguna. Data yang disimpan mencakup informasi pengguna yang mengajukan permintaan, jumlah barang yang diminta, status permintaan (pending, disetujui, atau ditolak), serta catatan tambahan yang diberikan oleh pengguna atau admin terkait dengan permintaan tersebut. Informasi dalam table ini sangat penting untuk melacak permintaan barang dan memastikan barang dapat disediakan dengan cepat.

Tabel. 4 Tabel Requests

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Atribut | Tipe Data | Ukuran | Keterangan |
| id\_request | *Integer* | 11 | *Primary key* |
| kode\_barang | *Varchar* | 50 |  |
| nama\_*user* | *Varchar* | 100 |  |
| quantity\_diminta | *Integer* | 11 |  |
| status | *Enum* | Pending, Disetujui, Ditolak |  |
| tanggal\_request | *Date* |  |  |
| catatan | *Text* |  |  |
| id\_*user* | *Integer* | 11 | *Foreign Key* |

1. Tabel requests\_gudang

Tabel ini hampir sama dengan table requests sebelumnya, data yang disimpan adalah data permintaan yang dialihkan dari gudang daerah apabila stok pada gudang daerah tidak tersedia barang yang diminta sehingga harus dialihkan ke gudang pusat. Tabel ini mencatat permintaan yang baru saja dialihkan ke gudang dengan status menunggu persetujuan admin, kemudian admin akan melakukan putusan terkait permintaan tersebut disetujui atau ditolak sehingga juga akan mengubah status permintaan dari pengguna.

Tabel. 5 Tabel Requests Gudang

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Atribut | Tipe Data | Ukuran | Keterangan |
| id\_request | *Integer* | 11 | *Primary key* |
| nama\_*user* | *Varchar* | 255 |  |
| quantity\_diminta | *Integer* | 11 |  |
| Status | *Enum* | Menunggu Persetujuan Admin, Disetujui, Ditolak |  |
| catatan | *Text* |  |  |
| id\_*user* | *Integer* | 11 | *Foreign Key* |
| tanggal\_request | *Timestamp* |  |  |
| nama\_barang | *Varchar* | 255 | *Foreign key* |
| notified | *Tinyinteger* | 1 |  |

1. Tabel *user*s

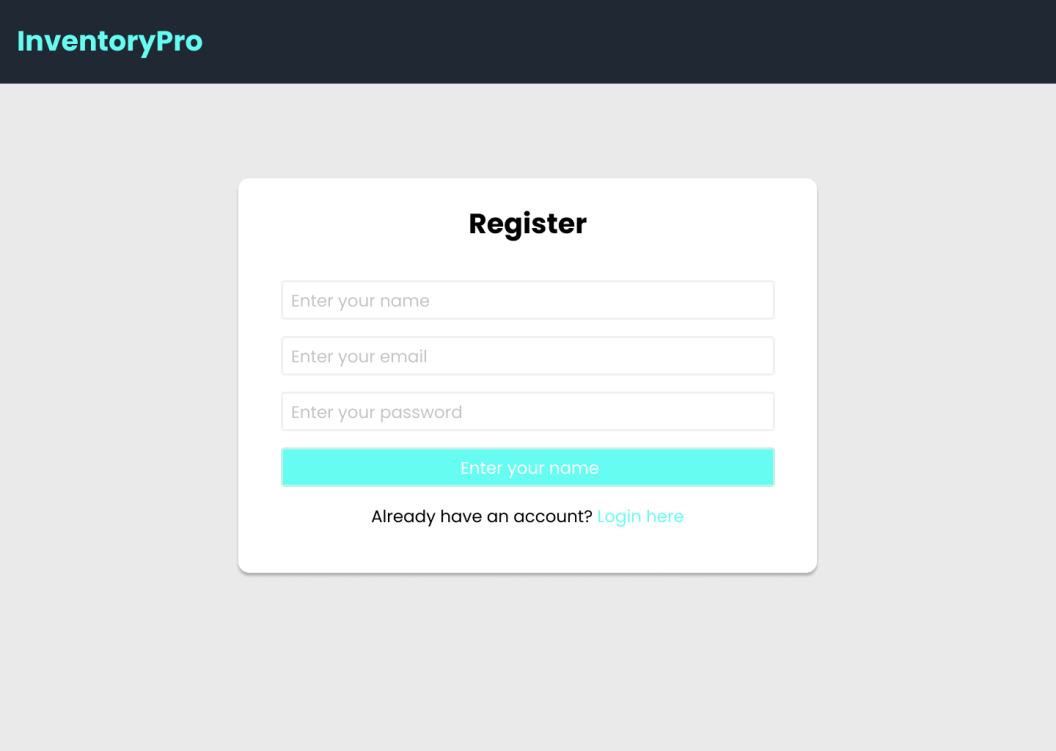
Tabel ini mencatat data keseluruhan *user* yang telah terdaftar, menyimpan nama *user*, email, dan juga level dari *user* tersebut, apakah hanya *user* biasa, admin daerah, atau admin gudang.

Tabel. 6 Tabel Users

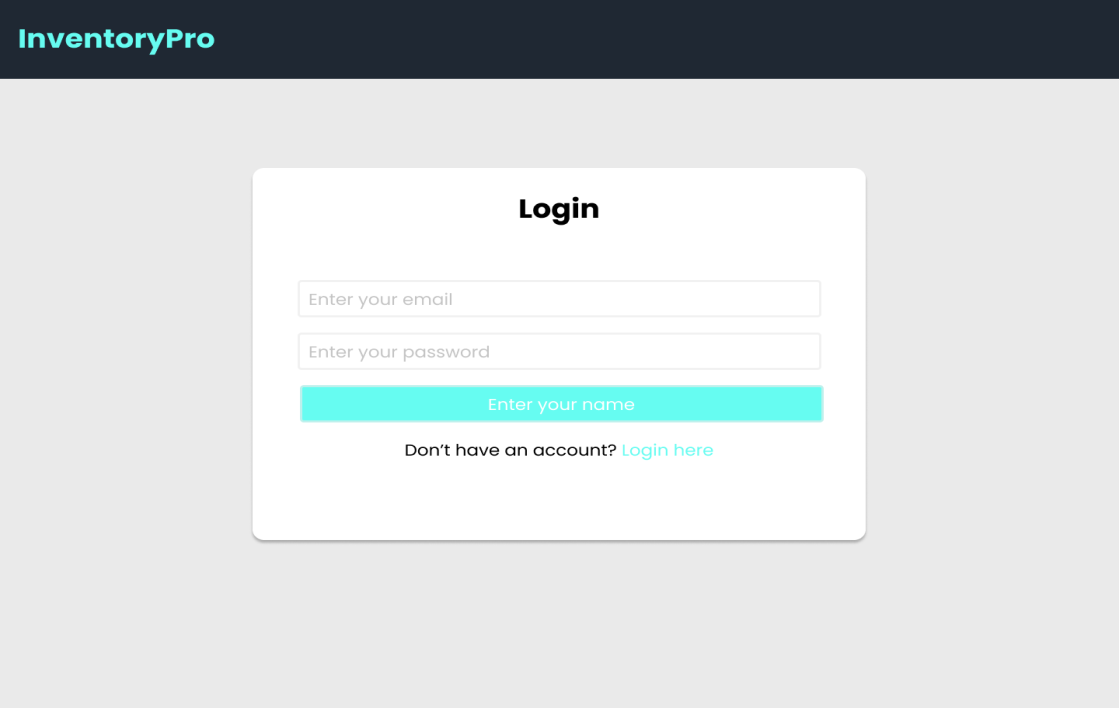
|  |  |  |  |
| --- | --- | --- | --- |
| Nama Atribut | Tipe Data | Ukuran | Keterangan |
| id\_*user* | *Integer* | 11 | *Primary key* |
| nama | *Varchar* | 100 |  |
| email | *Varchar* | 100 |  |
| *password* | *Varchar* | 255 |  |
| role | *enum* | *User*, Admin Daerah, Admin Gudang |  |

1. Desain Antarmuka
2. Halaman register & *login*

Berikut adalah tampilan dari halaman *login* dan halaman registrasi untuk *user* yang belum memiliki akun. Seperti yang terlihat pada gambar pada form registrasi pengguna dapat menginputkan data berupa nama pengguna, email, dan juga *password*.



Gambar. Desain Halaman *Login*

Bagi pengguna yang sudah memiliki akun, dapat langsung melakukan *login* dengan memasukkan email, dan juga *password* yang telah terdaftar. Setelah berhasil *login*, sistem akan secara otomatis mengarahkan pengguna ke halaman *dashboard*, dimana tampilan konten akan disesuaikan dengan level akses pengguna.

Gambar. Desain Halaman *Login*

Jika pengguna memiliki level *user* biasa (*user*), maka sistem secara otomatis akan menampilkan konten yang hanya dapat diakses oleh *user*, kemudian jika level *user* adalah admin daerah maka sistem juga akan menampilkan konten yang hanya dapat diakses oleh admin daerah tersebut, begitu juga dengan admin gudang, jika level *user* adalah admin gudang, maka sistem juga secara otomatis akan menampilkan konten yang hanya dapat diakses oleh admin gudang.

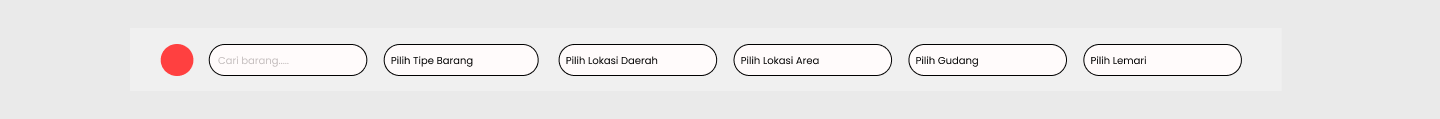
1. Tampilan tabel barang, form pencarian barang, dan filter pencarian barang

Pada halaman *dashboard*, baik itu *dashboard* *user* maupun admin, terdapat tabel barang yang menampilkan daftar lengkap barang yang tersimpan di gudang. Dalam tabel ini, pengguna dapat melihat nama barang, kode barang, serta spesifikasi lokasi penyimpanan barang tersebut.



Gambar. Tampilan Tabel Barang

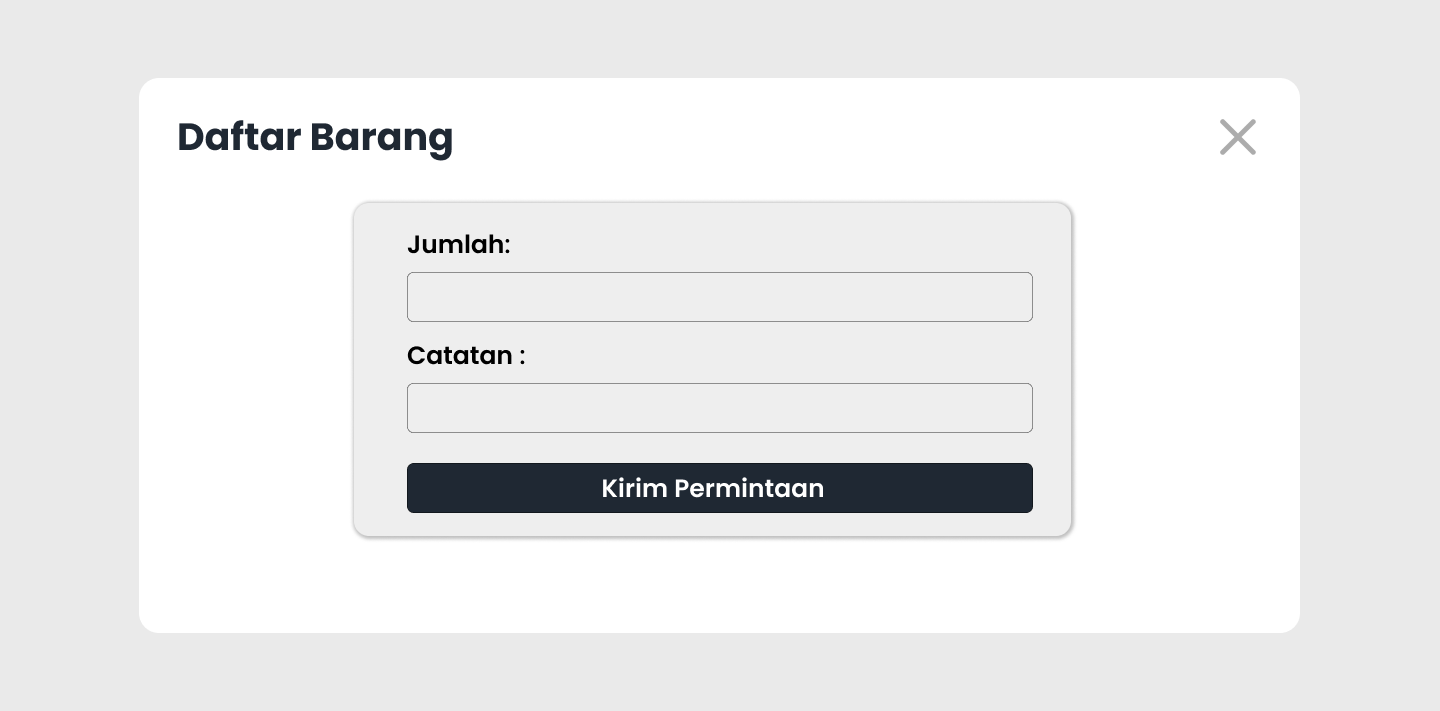
Di bagian atas tabel barang tersebut terdapat fitur pencarian dan filter barang yang memungkinkan pengguna untuk mencari barang berdasarkan nama atau kode barang. Selain itu, fitur filter juga mempermudah pengguna dalam menyaring daftar barang yang ingin ditampilkan. Penyaringan dapat dilakukan berdasarkan beberapa kategori seperti tipe barang, lokasi daerah, area, gudang, maupun lemari penyimpanan.



Gambar. Fitur Pencarian dan Filter Barang

1. Form permintaan barang untuk *user*

Bagi pengguna yang ingin mengajukan permintaan barang, cukup melakukan pencarian barang yang dibutuhkan. Untuk mempermudah pencarian, tersedia fitur pencarian dan filter, sehingga pengguna dapat menemukan barang dengan lebih cepat. Setelah menemukan barang yang diinginkan, pengguna hanya perlu mengklik tombol “Minta Barang”, dan sistem akan menampilkan form permintaan barang dalam bentuk *pop-up* seperti pada gambar berikut:



Gambar. Form Permintaan Barang

Pada form tersebut, pengguna dapat menginputkan jumlah barang yang dibutuhkan serta menambahkan catatan tambahan jika diperlukan. Jika semua data telah sesuai, pengguna cukup mengklik tombol “Kirim Permintaan”, dan permintaan akan langsung di proses oleh sistem.

1. Fungsi input data barang untuk admin

Selanjutnya untuk membantu admin dalam mengelola data barang yang terdapat di gudang, terdapat fitur untuk melakukan input data barang, dan untuk lebih mempermudah lagi, admin dapat menginputkan data barang dalam jumlah yang besar sekaligus dengan menginputkan file dalam format *Excel* yang berisikan data barang-barang yang akan diinputkan ke gudang.

Fitur ini terdapat pada *dashboard* admin daerah dan juga pada *dashboard* admin gudang, berikut adalah tampilan dari form input file data barang:



Gambar. Fitur Input Data Barang

Admin dapat memilih file *Excel* yang berisi data barang dengan mengklik tombol “Pilih File *Excel*”. Setelah memilih file yang sesuai, admin cukup mengklik tombol “Unggah”, dan sistem akan membaca serta menyimpan data barang ke dalam *database*.

Untuk menghindari kesalahan format, sistem juga menyediakan opsi untuk mengunduh template file *Excel* yang sudah sesuai dengan format yang diperlukan. Hal ini memastikan bahwa data yang diunggah dapat di proses dengan baik tanpa ada kesalahan struktur atau format.

1. Fungsi input data barang untuk admin

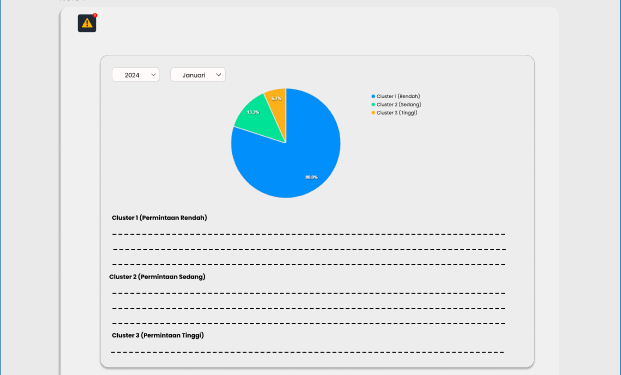
Pada *dashboard* *user*, terdapat tabel riwayat permintaan yang menampilkan daftar permintaan barang yang pernah diajukan oleh pengguna tersebut. Sementara itu, pada *dashboard* admin, tabel ini menampilkan seluruh riwayat permintaan barang dari semua pengguna yang mengajukan permintaan. Dengan tabel ini pengguna atau admin dapat dengan mudah melacak status permintaan barang.



Gambar. Tabel Riwayat Permintaan Barang

1. Tampilan hasil *clustering* pada dashboard admin

Di dalam *dashboard* admin, terdapat fitur hasil *clustering* pola permintaan barang yang dihasilkan berdasarkan data historis permintaan barang dari pengguna. Hasil *clustering* ini divisualisasikan dalam bentuk *pie chart*, yang menampilkan pola pengelompokan barang berdasarkan frekuensi permintaan.



Gambar. Hasil *Clustering* dengan *Pie Chart*

Di bawah grafik, terdapat daftar barang yang telah dikelompokkan menggunakan algoritma *K-Means*, sehingga admin dapat melihat pola permintaan barang dengan lebih jelas. Selain itu, hasil *clustering* dapat di filter berdasarkan periode waktu tertentu, seperti bulan dan tahun, sehingga admin dapat menganalisis tren permintaan barang dalam periode yang diinginkan.

1. Implementasi Algoritma *K-Means*
2. Proses *Clustering* dalam Sistem

Proses *clustering* pada sistem ini dilakukan dalam dua tahap utama, yaitu pemrosesan data di *backend* dan visualisasi hasil *clustering* di *frontend*.

1. *Backend* (Pemrosesan Data *Clustering*)

*Backend* bertanggung jawab untuk mengambil dan mengolah data permintaan barang dari table *requests* yang telah disetujui. Data yang diperoleh akan dikelompokkan ke dalam beberapa *cluster* berdasarkan pola permintaan barang. Langkah-langkah yang dilakukan adalah:

1. Pengambilan data historis

Data permintaan barang yang memiliki status Disetujui dikumpulkan dari *database*. Setiap permintaan barang dicatat berdasarkan kode barang, jumlah permintaan, dan tanggal permintaan.

1. Normalisasi data

Data dinormalisasi untuk memastikan tidak ada atribut yang mendominasi proses *clustering*. Penghapusan data duplikat serta penyusunan format tanggal dan kuantitas dilakukan sebelum *clustering*.

1. Penerapan algoritma *K-Means*

Algoritma *K-Means* diterapkan untuk mengelompokkan barang berdasarjan pola permintaan menggunakan tiga *cluster* (permintaan rendah, sedang, dan tinggi)

1. Penyimpanan dan pengiriman hasil *clustering* ke *frontend*

Hasil *clustering* dikirimkan dalam format *JSON* melalui *API*. Setiap barang diberi label sesuai dengan *cluster* yang telah di tentukan.

1. *Frontend*

*Frontend* bertanggung jawab untuk menampilkan hasil *clustering* dalam bentuk grafik dan table yang dapat digunakan oleh admin gudang untuk menganalisis permintaan barang. Tahapan utama dalam *frontend* adalah:

1. Pengambilan data *clustering* dari *backend*

Data *clustering* yang telah di proses di *backend* ditampilkan dalam bentuk tabel dan grafik interaktif. Data dapat difilter berdasarkan periode waktu tertentu (bulan dan tahun).

1. Pengelompokan barang berdasarkan pola permintaan

Sistem menampilkan informasi stok dan permintaan barang dalam bentuk *pie chart* atau bar chart menggunakan *Apexcharts*. Barang dengan permintaan tinggi diberikan warna khusus untuk membedakan dari barang dengan permintaan rendah.

1. *Data Preprocessing*

Untuk memastikan hasil *clustering* yang optimal, data permintaan barang diproses melalui tahap-tahap berikut:

1. Pengumpulan Data

Data diambil dalam tabel permintaan barang dengan status “Disetujui”.

1. Pembersihan Data

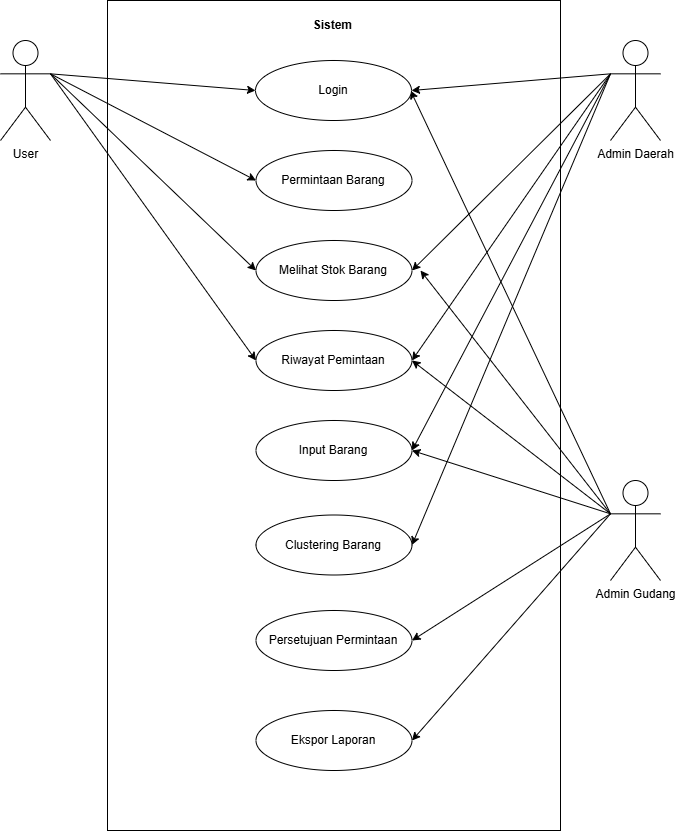
Data duplikat dihapus, dan format data diperiksa agar konsisten, seperti format tanggal dan kuantitas.

1. Normalisasi Data

Data dinormalisasi untuk memastikan tidak ada atribut yang mendominasi proses *clustering*.

1. Teknologi dan *Tools*
2. *Frontend* menggunakan *HTML, CSS, Javascript.*
3. *Backend* menggunakan *Node.js, Express*.
4. *Database* menggunakan *MySQL*.
5. Visualisasi data hasil *clustering* menggunakan *ApexCharts*.
6. *Tools* pendukung lainnya seperti *Git* dan *GitHub* untuk version control, Postman untuk pengujian *API, XLSX* dan *jsPDF* untuk membaca dan melakukan ekspor data.
7. Use Case Diagram

*Use case diagram* menggambarkan interaksi antara aktor dan sistem, serta fitur-fitur yang ada pada sistem manajemen inventori barang.……

.…..

Gambar. 22 *Use Case Diagram*

Diagram ini membantu memvisualisasikan fungsi utama yang tersedia serta peran masing-masing aktor dalam sistem, berikut ini adalah interaksi aktor dan sistem:

1. Admin Mengelola Data Barang

Admin memiliki peran utama untuk memastikan data barang dalam sistem selalu terbarui dan akurat. Aktivitas yang dilakukan admin meliputi:

1. Input data barang baru ke dalam sistem.
2. Memeriksa dan memberikan keputusan terhadap permintaan barang yang dilakukan *user* dengan melakukan persetujuan ataupun penolakan terhadap permintaan tersebut.
3. Melakukan analisis *clustering* berdasarkan data permintaan untuk mengelompokkan barang sesuai pola yang ditentukan.
4. Ekspor laporan terkait stok barang, permintaan, atau hasil *clustering* dalam format dokumen tertentu seperti *Excel* atau *PDF*.
5. Pengguna Mengajukan Permintaan Barang

Pengguna (*user*) dapat berinteraksi dengan sistem untuk melakukan permintaan barang yang dibutuhkan, serta memantau status dan riwayat permintaan mereka. Aktivitas pengguna meliputi:

1. *Login* dan mengakses sistem dengan autentikasi.
2. Mengajukan permintaan barang dengan menentukan nama barang, jumlah yang akan diminta, dan keterangan tambahan jika diperlukan.
3. Mengecek ketersediaan barang dalam sistem.
4. Memantau daftar permintaan barang sebelumnya, termasuk status persetujuan oleh admin.
5. Sistem Melakukan *Clustering* dan Menampilkan Hasilnya

Sistem menyediakan fitur *clustering* untuk membantu admin menganalisis pola permintaaan barang. Proses *clustering* ini bertujuan untuk mengelompokkan barang berdasarkan karakteristik terttentu, seperti:

1. Frekuensi permintaan.
2. Jumlah barang yang diminta.
3. Waktu permintaan.

Hasil *clustering* ditampilkan dalam bentuk visualisasi yang memudahkan admin dalam pengambilan keputusan, seperti menentukan stok prioritas atau mengelompokkan gudang barang.

1. Pengelolaan Data

Pengelolaan data dalam sistem ini mencakup beberapa hal utama yang memastikan data dapat tersimpan, diproses, dan digunakan dengan optimal dalam mendukung operasional sistem.

1. Jenis Data

Sistem ini mengelola beberapa jenis utama yang mendukung berbagai proses dalam sistem.

1. Data Barang

Data barang mencakup informasi lengkap mengenai barang yang tersimpan di gudang. Data ini meliputi kode barang, nama barang, jumlah stok, harga satuan, tipe barang, serta lokasi penyimpanan. Data ini sangat penting dalam mendukung fitur pencarian barang, pemantauan stok, serta pengelolaan permintaan barang yang dilakukan oleh pengguna dan admin. Dengan adanya data barang yang terstruktur, sistem dapat memastikan bahwa stok barang selalu tercatat dengan akurat serta dapat ditelusuri dengan mudah melalui *dashboard*.

1. Data Permintaan Barang

Data permintaan barang berisi catatan mengenai setiap barang yang diminta oleh pengguna. Data ini mencakup ID request, kode barang, nama *user*, jumlah barang yang diminta, status permintaan, serta tanggal permintaan. Setiap permintaan yang masuk akan diproses dan di verifikasi oleh admin gudang maupun admin daerah. Data ini tidak hanya berguna dalam mencatat riwayat transaksi permintaan, tetapi juga dalam membantu analisis pola permintaan barang guna meningkatkan efisiensi pengelolaan stok serta memastikan ketersedian barang yang sering dibutuhkan.

1. Data *Clustering*

Data *clustering* digunakan dalam proses analisis permintan barang berdasarkan riwayat permintaan sebelumnya. Data ini meliputi jumlah permintaan, kategori barang, serta hasil *clustering* berdasarkan algoritma *K-Means*. Proses *clustering* ini bertujuan untuk mengelompokkan barang berdasarkan tingkat permintaan, sehingga admin dapat memahami pola kebutuhan barang dari waktu ke waktu. Hasil dari *clustering* ini divisualisasikan dalam bentuk grafik di *dashboard* admin untuk membantu pengambilan keputusan terkait pengelolaan inventaris dan perencanaan pengadaan barang di masa mendatang.

1. Format Data

Data dalam sistem ini disimpan dalam tabel-tabel terstruktur dalam *database* *MySQL*, dengan relasi yang telah dioptimalkan untuk mendukung efisiensi pencarian dan pemrosesan data. Hubungan antar tabel dijaga dengan relasi kunci primer dan kunci asing guna memastikan integritas data tetap terjaga. Data barang disimpan dalam tabel barang daerah dan barang gudang, di mana setiap entri memiliki kode unik barang serta atribut lain yang relevan seperti jumlah stok, harga satuan, lokasi penyimpanan dan tipe barang.

Untuk data permintaan barang, sistem menyimpannya dalam tabel requests yang mencatat seluruh transaksi permintaan dari pengguna. Setiap entri dalam tabel ini mencakup id request, kode barang, nama *user*, jumlah barang yang diminta, serta status permintaan. Sedangkan data *clustering* yang dihasilkan malalui analisis pola permintaan disimpan dalam tabel khusus atau dianalisis langsung dari data permintaan yang telah tersimpan di dalam sistem.

1. Alur Pengolahan Data

Proses pengolahan data dalam sistem terdiri dari berbagai tahapan utama yang memastikan data dapat digunakan secara efisien dan optimal. Tahap pertama adalah input data, dimana barang dapat diunggah dalam format file *Excel*. Pengguna juga dapat melakukan input data permintaan barang melalui form yang tersedia dalam sistem. Untuk mempermudah pengelolaan barang dalam jumlah besar, admin dapat mengunggah file *Excel* yang berisi data barang dalam format yang telah ditentukan.

Setelah data diinputkan, sistem akan menjalankan tahap kedua, yaitu proses penyimpanan data. Pada tahap ini, data yang telah dimasukkan akan divalidasi terlebih dahulu untuk memastikan kesesuaian format serta kelengkapan data. Jika data sudah sesuai, maka akan disimpan dalam *database* *MySQL* dengan struktur tabel yang telah ditentukan agar dapat diakses dan diproses dengan optimal.

Selanjutnya, sistem akan menjalankan tahap ketiga yaitu pengolahan dan analisis data. Dalam tahap ini, sistem akan mengelola data permintaan barang dan menerapkan algoritma *K-Means* untuk mengidentifikasi pola permintaan. Proses *clustering* ini akan mengelompokkan barang berdasarkan pola permintaan rendah, sedang, atau tinggi. Hasil dari proses *clustering* ini akan disimpan dan divisualisasikan dalam bentuk grafik interaktif pada *dashboard* admin, sehingga admin dapat dengan mudah memahami pola permintaan barang berdasarkan data historis.

Terakhir, sistem akan menjalankan tahap penyajian data, di mana informasi mengenai barang, riwayat permintaan, serta hasil *clustering* ditampilkan dalam *dashboard* *user* dan admin, Admin memiliki akses untuk melihat laporan secara lebih mendetail serta dapat mengunduh data dalam format *Excel* atau *PDF* untuk keperluan analisis lebih lanjut dan pengambilan keputusan strategis terkait pengelolaan stok barang.

# DAFTAR PUSTAKA

Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The *K-Means* algorithm: A comprehensive survey and performance evaluation. *Electronics (Switzerland)*, *9*(8), 1–12. https://doi.org/10.3390/electronics9081295

Andi Syahrul Ramdana, Kusrini, & Pramono, E. (2024). Penerapan Algoritma *K-Means* Untuk Manajemen Persediaan Di Perpustakaan. *Jurnal Informatika Teknologi Dan Sains (Jinteks)*, *6*(1), 109–114. https://doi.org/10.51401/jinteks.v6i1.3911

Calista, S., Husaein, A., & Gunardi. (2023). Perancangan Sistem Informasi Inventory Barang Berbasis *Web* pada Toko Laris Furniture Jambi. *Jurnal Manajemen Teknologi Dan Sistem Informasi (JMS)*, *3*(2), 437–449. https://doi.org/10.33998/jms.2023.3.2.788

Fachrul Rezy, A., & Ikasari, I. H. (2023). Systematic Literature Review: Sistem Informasi Manajemen Inventory Barang Berbasis *Web*. *BIIKMA : Buletin Ilmiah Ilmu Komputer Dan Multimedia*, *1*(1), 121–125. https://jurnalmahasiswa.com/index.php/biikma

Fortuna Alfarisi, A., Agita Rindri, Y., Josi, A., & Manufaktur Negeri Bangka Belitung, P. (2023). *Sistem Informasi Manajemen Inventaris Berbasis Web di SDIT Alam Biruni* (Vol. 01, Issue 1).

Gren, L., & Lenberg, P. (2020). Agility is responsiveness to change: An essential definition. *ACM International Conference Proceeding Series*, 348–353. https://doi.org/10.1145/3383219.3383265

Handoko, S., Fauziah, F., & Handayani, E. T. E. (2020). Implementasi *Data mining* Untuk Menentukan Tingkat Penjualan Paket Data Telkomsel Menggunakan Metode *K-Means Clustering*. *Jurnal Ilmiah Teknologi Dan Rekayasa*, *25*(1), 76–88. https://doi.org/10.35760/tr.2020.v25i1.2677

Hery, Renaldo Luih, J., Alencia Haryani, C., & E. Widjaja, A. (2022). Penerapan Teknologi Qr Code Berbasis *Web* pada Sistem Manajemen Inventaris di Gudang PT XYZ. *Technomedia Journal*, *7*(2), 202–215. https://doi.org/10.33050/tmj.v7i2.1903

Lisa, L., Sutejo, H., & Kungkung, A. Y. (2025). *Sistem Informasi Manajemen Persediaan Barang Berbasis Website di PT . Rejo Mulyo Solution*. *6*(3), 1742–1753.

Muhamad Rizki, M. (2023). Penerapan Metode *K-Means* Pada Data Penjualan Untuk. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *5*(1), 228–236.

Nurcahyawati, V., Riyondha Aprilian Brahmantyo, & Januar Wibowo. (2023). Manajemen Persediaan Menggunakan Metode Safety Stock dan Reorder Point. *Jurnal Sains Dan Informatika*, *9*(April), 89–99. https://doi.org/10.34128/jsi.v9i1.431

Nurdiyansyah, F., & Akbar, I. (2021). Implementasi Algoritma *K-Means* untuk Menentukan Persediaan Barang pada Poultry Shop. *Jurnal Teknologi Dan Manajemen Informatika*, *7*(2), 86–94. https://doi.org/10.26905/jtmi.v7i2.6377

Permadi, A., & Wiyaja, Y. A. (2023). Pengelompokan Terbaik Menggunakan Algoritma *K-Means* Pada Dataset Bus Biskita Bogor. *INTERNAL (Information System Journal)*, *6*(1), 88–100. https://doi.org/10.32627/internal.v6i1.689

Putri, A. Y., & Budayawan, K. (2024). *Gudang Jurnal Multidisiplin Ilmu Implementasi Algoritma K-Means Menggunakan Metode Clustering Untuk Menentukan Penjualan Produk Laris Dan Tidak Laris Di Grosir Chintiya*. *2*, 1–13.

Ramadhan, A. R., & Valentino, M. (2024). Implementasi Sistem Manajemen Persediaan Berbasis *Web* Untuk Efisiensi Stok Barang. *Buletin Ilmiah Ilmu Komputer …*, *2*(1), 96–107. http://jurnalmahasiswa.com/index.php/biikma/article/view/1136%0Ahttps://jurnalmahasiswa.com/index.php/biikma/article/download/1136/701

Ray Silaen, B., Nasution, M., & Muti’ah, R. (2024). Implementation of the ABC Analysis to the Inventory Management. *International Journal of Science, Technology & Management*, *5*(4), 816–825. https://doi.org/10.46729/ijstm.v5i4.1144

Samsudin, R., Martanto, M., & Hayati, U. (2024). Optimalisasi Stok Barang Melalui Algoritma *K-Means Clustering* Analisis Untuk Manajemen Persediaan Dalam Konteks Bisnis Modern. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(3), 3572–3580. https://doi.org/10.36040/jati.v8i3.9742

Saputri, A., & Hirzan, A. M. (2024). Aplikasi Manajemen Inventori Berbasis Mobile Menggunakan Flutter Dan Firebase Realtime Database. *Jurnal Informatika Dan Teknik Elektro Terapan*, *12*(3), 1586–1592. https://doi.org/10.23960/jitet.v12i3.4324

Sari, N., Handayani, H. H., & Siregar, A. M. (2023). Implementasi *Clustering* Data Kasus Covid 19 Di Indonesia Menggunakan Algoritma *K-Means*. *Bianglala Informatika*, *11*(1), 7–12. https://doi.org/10.31294/bi.v11i1.14762