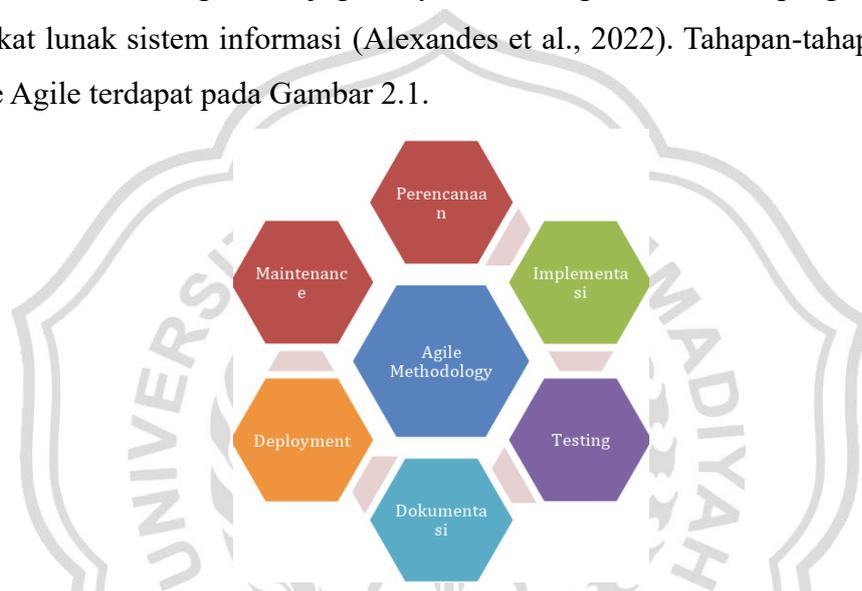


BAB 2

LANDASAN TEORI

2.1 Teori Pengembangan Sistem

Pengembangan Sistem atau Software Development adalah aktivitas seperti mendefinisikan, mengembangkan, menguji, mengirim, mengoperasikan, dan memelihara perangkat lunak atau sebuah sistem informasi. Salah satu metode yang sering digunakan dalam SDLC yaitu metode Agile. Metode Agile memiliki kelebihan dan kekurangan, dan juga banyak sekali digunakan dalam pengembangan perangkat lunak sistem informasi (Alexandes et al., 2022). Tahapan-tahapan pada metode Agile terdapat pada Gambar 2.1.



Gambar 2.1 Metode Agile

1. Perencanaan, tahap ini adalah tahap membuat rencana sistem yang akan dikembangkan dengan mengumpulkan data pengguna dalam bentuk kuesioner untuk mendapatkan kebutuhan yang diinginkan oleh pengguna (Saputra et al., 2022).
2. Implementasi, pada tahapan ini perencanaan yang telah dibuat sebelumnya dan akan diimplementasikan menjadi sebuah sistem informasi atau aplikasi.
3. Testing, pada tahapan ini pengujian dilakukan pada sistem informasi atau aplikasi yang telah dikembangkan untuk menemukan kesalahan yang ada di dalam sistem.

4. Dokumentasi, pada tahapan ini dilakukan dokumentasi modul dan fungsi sebagai catatan selama pengembangan dan untuk memudahkan tim dalam pengembangan selanjutnya (Saputra et al., 2022).
5. Deployment, pada tahapan ini sistem yang telah dilakukan testing akan di unggah ke penyedia layanan web supaya sistem dapat diakses oleh publik.
6. Pemeliharaan, tahapan ini dilakukan secara berkala agar aman dari kesalahan sistem atau celah sistem.

Pengelompokkan aktivitas pengembangan sistem informasi inventori dilakukan pada tahap perencanaan dengan melakukan pengamatan pada fitur-fitur aplikasi yang sudah ada dan fitur-fitur aplikasi yang digunakan pada penelitian terdahulu untuk mendapatkan fitur aplikasi yang relevan. Selain menentukan fitur yang akan digunakan, dilakukan penentuan beban kerja untuk masing-masing fitur untuk mendapatkan estimasi waktu yang akan digunakan dalam implementasi fitur ke dalam aplikasi.

2.2 Aktivitas Pengembangan

Aktivitas pengembangan sistem informasi inventori adalah proses yang melibatkan pembuatan dan implementasi sistem berbasis komputer untuk mengelola inventori dengan lebih efisien. Pengelompokkan aktivitas pengembangan sistem informasi inventori adalah proses membagi keseluruhan proyek pengembangan sistem informasi inventori menjadi beberapa aktivitas atau kegiatan yang terstruktur, dan setiap aktivitas memiliki tujuan spesifik yang membantu dalam membangun sistem informasi inventori yang efektif dan efisien (Oladele et al., 2021).

2.3 Fitur-fitur aplikasi inventori

Fitur merupakan sebuah fungsi yang dimiliki oleh sebuah aplikasi. Fitur-fitur dirancang dan diimplementasikan kedalam aplikasi untuk memenuhi kebutuhan pengguna atau user dalam menggunakan aplikasi tersebut. Fitur-fitur dalam aplikasi inventory meliputi manajemen barang atau stok barang, manajemen penjualan dan pembelian atau restock barang, manajemen pemasukan bulanan atau tahunan, dan manajemen vendor atau pemasok barang.

Fitur-fitur yang digunakan dalam aplikasi sistem inventaris adalah sebagai berikut:

1. Fitur manajemen barang

Fitur ini berfungsi untuk menambahkan data barang atau stok barang kedalam database aplikasi. Selain menambahkan data, terdapat fitur untuk mengubah atau edit data barang, dan fitur hapus data barang (Fatimah, 2022).

Untuk daftar fitur manajemen barang terdapat pada tabel 2.1.

Tabel 2.1 Fitur manajemen barang

No	Fitur
1	Tambah barang
2	Edit Barang
3	Hapus Barang
4	List Barang

2. Fitur manajemen penjualan

Fitur ini berfungsi untuk menambahkan data penjualan atau transaksi barang kedalam database aplikasi. Selain menambahkan data penjualan atau transaksi, terdapat fitur untuk mengubah data penjualan dan fitur menghapus data penjualan (Fatimah, 2022; Maulana et al., 2023). Untuk daftar fitur manajemen penjualan terdapat pada tabel 2.2.

Tabel 2.2 Fitur manajemen penjualan

No	Fitur
1	Tambah Penjualan
2	Edit Penjualan
3	Hapus Penjualan
4	List Penjualan

3. Fitur manajemen pembelian

Fitur ini berfungsi untuk menambahkan data pembelian atau restok barang dari pemasok kedalam database aplikasi. Selain fitur tambah data, ada fitur ubah data dan hapus data pembelian atau restok barang (Maulana et al.,

2023; Sodikin et al., 2021). Untuk daftar fitur manajemen pembelian terdapat pada tabel 2.3.

Tabel 2.3 Fitur manajemen pembelian

No	Fitur
1	Tambah Pembelian
2	Edit Pembelian
3	Hapus Pembelian
4	List Pembelian

4. Fitur manajemen vendor atau pemasok barang

Fitur ini berfungsi untuk menambahkan data vendor atau pemasok barang kedalam database aplikasi. Selain fitur untuk menambah data, terdapat fitur untuk mengubah data dan menghapus data vendor atau pemasok barang (Sodikin et al., 2021). Untuk daftar fitur manajemen vendor terdapat pada tabel 2.4.

Tabel 2.4 Fitur manajemen vendor

No	Fitur
1	Tambah Vendor
2	Edit Vendor
3	Hapus Vendor
4	List Vendor

5. Fitur hubungi owner atau vendor

Fitur ini berfungsi untuk menghubungi pemilik atau owner, atau pemasok barang atau vendor, untuk mempermudah komunikasi dalam melakukan transaksi pembelian barang atau restok barang (Sodikin et al., 2021). Untuk daftar fitur hubungi owner dan vendor terdapat pada tabel 2.5.

Tabel 2.5 Fitur hubungi owner dan vendor

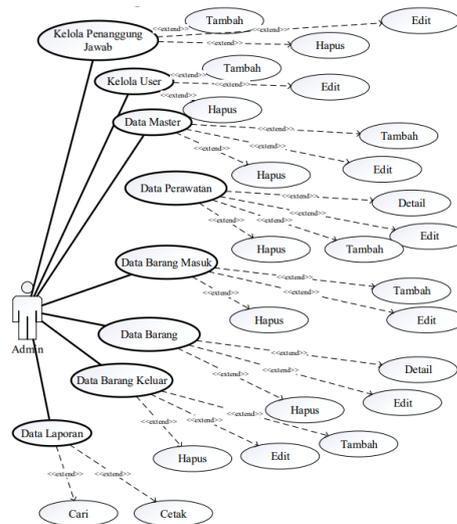
No	Fitur
1	Hubungi Vendor

No	Fitur
2	Hubungi Owner

2.4 Aplikasi inventori

Inventori merupakan sebuah kata yang berasal dari bahasa Inggris, Echols dan Shadily merumuskan dalam Kamus Besar Bahasa Indonesia sebagai daftar barang disertai dengan nilainya masing-masing yang dimiliki perusahaan atau organisasi dalam kurun waktu tertentu yang digunakan dalam kegiatan usaha perusahaan atau organisasi. Dalam praktek, inventory disebut juga sebagai persediaan barang yang artinya barang-barang biasanya dapat dijumpai digudang tertutup, lapangan, gudang terbuka, atau tempat-tempat penyimpanan lain, baik berupa bahan baku, barang setengah jadi, barang jadi, barang-barang untuk keperluan operasi atau barang-barang untuk keperluan suatu proyek (Hay's et al., 2018).

Aplikasi inventori, adalah sebuah sistem informasi yang dibuat dengan tujuan untuk memudahkan dalam kegiatan manajemen barang atau stok barang, pembelian atau restock barang dan penjualan barang, dan manajemen pemasok barang. Tujuan dari aplikasi inventori adalah untuk memudahkan pengguna dalam mengelola ketersediaan barang agar tidak terjadi kekurangan atau kelebihan barang, serta dapat mengoptimalkan efisiensi dan keuntungan suatu perusahaan atau organisasi. Gambaran *use case diagram*nya terdapat pada Gambar 2.3.



Gambar 2.2 Use Case Diagram (Khalim et al., 2020)

2.5 Metode Moscow (*Must have, Should have, Could have, Won't have*)

Metode Moscow merupakan salah satu metodologi pembangunan perangkat lunak berbasis agile-modelling yaitu DSDM (*Dynamic Software Development Method*). Kelebihan dari metode Moscow adalah mudah dan cepat dalam mengurutkan daftar kebutuhan serta menghasilkan urutan yang akurat. Teknik ini mengelompokkan seluruh kebutuhan ke dalam 4 (empat) kelas berdasarkan tingkat prioritasnya. (Marthasari & Hayatin, 2020)

Agile-modelling adalah model pengembangan perangkat lunak dalam jangka pendek untuk kemudian diadaptasi secara cepat dalam mengatasi setiap perubahan. Nilai terpenting dari *Agile-modelling* ini adalah memungkinkan sebuah tim dalam mengambil keputusan dengan cepat, kualitas dan prediksi yang baik, serta memiliki potensi yang baik dalam menangani setiap perubahan. (Hikmah et al., 2021)

Terdapat 4 (empat) tingkat prioritas yang digunakan dalam metode Moscow, yaitu *Must have, Should have, Could have, dan Won't have*. Pada prioritas *Must have*, merupakan tingkatan tertinggi, yang artinya jika terdapat fitur yang memiliki prioritas *Must have* didalam aplikasi, maka fitur tersebut harus digunakan karena tanpa adanya fitur tersebut, aplikasi yang dikembangkan akan kurang atau bahkan tidak cocok untuk digunakan dalam kehidupan sehari-hari. Prioritas *Should have*, merupakan prioritas yang harus digunakan atau diterapkan kedalam aplikasi. Jika dalam pengembangan aplikasi terdapat fitur dengan prioritas *Should have*, maka

pengembang sebaiknya menggunakan fitur tersebut, karena prioritas ini bersifat sebagai pendukung dalam pengembangan aplikasi, beda dengan prioritas *Could have*. Pada prioritas *Could have* adalah prioritas yang bersifat opsional, karena tanpa adanya fitur yang berprioritas ini, aplikasi tersebut masih bisa berjalan dengan baik. Prioritas *Won't have* merupakan prioritas yang harus tidak dikerjakan sesegera mungkin atau tidak perlu digunakan. Karena dengan ada atau tidaknya fitur dengan prioritas ini aplikasi yang dikembangkan masih tetap bisa berjalan dengan normal. Terdapat tahapan-tahapan yang harus dilakukan dalam menggunakan metode Moscow, seperti identifikasi kebutuhan, klasifikasi kebutuhan, penilaian, kuesioner, dan analisis hasil. Untuk penjelasan dari tahapan-tahapan metode Moscow adalah sebagai berikut:

1. Identifikasi Kebutuhan: Mengidentifikasi kebutuhan untuk proyek yang mencakup kebutuhan fungsional (fitur yang akan digunakan atau keinginan pengguna) dan non-fungsional (persyaratan performa, keamanan, dll). Setiap kebutuhan didokumentasikan secara jelas untuk memastikan semua orang yang terlibat dalam proyek dapat memahami tentang aplikasi yang akan dibangun.
2. Klasifikasi Kebutuhan: Mengklasifikasikan kebutuhan fitur ke dalam salah satu kategori Moscow (Must, Should, Could, dan Won't). Pengelompokan kategori dilakukan sesuai dengan tingkat kebutuhan untuk mencapai tujuan proyek. Kebutuhan yang kritis dan wajib ada masuk ke dalam kategori Must have, sedangkan kebutuhan yang kurang penting bisa dimasukkan kedalam kategori Could have atau Won't have.
3. Penilaian: Setelah melakukan pengelompokan berdasarkan kebutuhan, langkah selanjutnya adalah melakukan penilaian terhadap kebutuhan-kebutuhan tersebut berdasarkan tingkat kepentingan (Must, Should, Could, atau Won't) secara individu.
4. Perbandingan: Setelah melakukan penilaian secara individu, dilakukan perbandingan penilaian dengan membuat kuesioner yang akan digunakan sebagai pembanding antara penilaian pribadi terhadap

prioritas kebutuhan dengan penilaian kolektif dari sejumlah orang yang memiliki pengetahuan dan pengalaman yang relevan dengan proyek. Tujuan dari perbandingan penilaian pribadi dengan penilaian kolektif adalah untuk mengidentifikasi dan memprioritaskan kebutuhan berdasarkan urgensi dan kepentingan.

5. Analisis hasil: Setelah pengumpulan data selesai, dilakukan analisis terhadap hasil dari perbandingan pengelompokan pribadi dan kuesioner untuk mendapatkan prioritas kebutuhan yang kritis atau penting, serta memahami perbedaan persepsi atau prioritas kebutuhan. Hasil analisis menjadi landasan untuk menentukan prioritas akhir dari setiap kebutuhan, memastikan bahwa kebutuhan yang paling kritis dan wajib ada diprioritaskan dengan benar.

Berikut adalah tabel contoh klasifikasi kebutuhan menggunakan metode Moscow dengan menggunakan tabel fitur yang telah dibuat dan diubah menjadi tabel aktivitas pengembangan untuk sistem informasi inventori. Tabel aktivitas pengembangan dengan prioritas moscow terdapat pada Tabel 2.6.

Tabel 2.6 Aktivitas pengembangan dengan prioritas moscow

No	Aktivitas Pengembangan	Prioritas
1	Tambah barang	<i>Must have</i>
2	Edit Barang	<i>Must have</i>
3	Hapus Barang	<i>Must have</i>
4	List Barang	<i>Must have</i>
5	Tambah Penjualan	<i>Must have</i>
6	Edit Penjualan	<i>Must have</i>
7	Hapus Penjualan	<i>Could have</i>
8	List Penjualan	<i>Must have</i>
9	Tambah Pembelian	<i>Should have</i>
10	Edit Pembelian	<i>Should have</i>
11	Hapus Pembelian	<i>Could have</i>
12	List Pembelian	<i>Should have</i>

No	Aktivitas Pengembangan	Prioritas
13	Tambah Vendor	<i>Should have</i>
14	Edit Vendor	<i>Should have</i>
15	Hapus Vendor	<i>Won't have</i>
16	List Vendor	<i>Should have</i>
17	Hubungi Vendor	<i>Could have</i>
18	Hubungi Owner	<i>Should have</i>

2.6 Analisis Korelasi Pearson

Analisis korelasi adalah metode statistika yang digunakan untuk menentukan suatu besaran yang menyatakan bagaimana kuat hubungan suatu variabel dengan variabel lain dengan tidak mempersoalkan apakah suatu variabel tertentu tergantung kepada variabel lain. Semakin nyata hubungan garis lurus atau linier, maka semakin kuat atau tinggi derajat hubungan garis lurus antara kedua variabel atau lebih (Mustafidah & Giarto, 2021). Apabila terdapat angka korelasi mendekati nilai satu, maka korelasi dari dua variabel akan semakin kuat. Sebaliknya, jika angka korelasi mendekati nol maka korelasi dua variabel semakin lemah (Altman, 2020).

Terdapat tiga teknik korelasi yang digunakan dalam menganalisa korelasi hubungan antara variabel terikat (dependent) dengan variabel bebas (independent). Diantaranya adalah Korelasi Pearson, Korelasi Spearman, dan Korelasi Kendall. Korelasi Pearson merupakan korelasi sederhana yang hanya melibatkan satu variabel terikat dan satu variabel bebas. Korelasi Pearson menghasilkan koefisien korelasi yang berfungsi untuk mengukur kekuatan hubungan linier antara dua variabel. Jika hubungan dua variabel tidak linier, maka nilai koefisien korelasi Pearson tersebut tidak mencerminkan kekuatan hubungan dua variabel yang sedang diteliti, meski kedua variabel mempunyai hubungan kuat. Koefisien korelasi ini disebut korelasi Pearson karena diperkenalkan pertama kali oleh Karl Pearson pada tahun 1990 (Safitri, 2016).

Analisis korelasi pearson digunakan untuk mengetahui hubungan masing-masing variabel dari metode Moscow (*Must have, Should have, Could have, Won't*

have) dengan variabel pengamatan yang telah dilakukan pada aplikasi-aplikasi yang sudah ada dan dari penelitian terdahulu. Setelah mendapatkan hasil uji korelasi pearson, dilakukan proses eliminasi pada hasil uji korelasi pearson yaitu dengan menghilangkan variabel yang memiliki nilai signifikan terendah diantara metode Moscow. Tahapan ini dilakukan untuk mendapatkan tingkat presisi atau akurasi dalam proses k-means *clustering*. Rumus perhitungan uji korelasi pearson dapat dilihat pada Pers 2.1.

$$r = \frac{n\sum XY - (\sum X)(\sum Y)}{\sqrt{(n\sum(X)^2 - (\sum X)^2)(n\sum(Y)^2 - (\sum Y)^2)}} \quad \text{Pers 2.1 Uji korelasi pearson}$$

Keterangan

r =	Koefisien korelasi pearson
Σ =	Jumlah data
n =	Jumlah sampel
x =	Variabel bebas pertama
y =	Variabel bebas kedua

2.7 K-Means Clustering

K-Means merupakan salah satu algoritma dalam data mining yang bisa digunakan untuk melakukan pengelompokan atau *clustering* suatu data. Ada banyak pendekatan untuk membuat *cluster*, diantaranya adalah membuat aturan yang mendikte keanggotaan dalam grup yang sama berdasarkan tingkat persamaan diantara anggota-anggotanya. Pendekatan lainnya adalah dengan menggunakan sekumpulan fungsi yang mengukur beberapa properti dari pengelompokan tersebut sebagai fungsi dari beberapa parameter dari sebuah *clustering* (Windha Mega Pradnya Duhita, 2016). Prinsip dasar algoritma K-Means yaitu menentukan *centroid*, dan menentukan anggota *cluster*. *Centroid* adalah pusat dari *cluster*, dimana tiap-tiap *cluster* memiliki *centroid*. Untuk menentukan *centroid* awal, dilakukan secara acak atau random. Ketika menentukan anggota *cluster*, pengelompokan tersebut dihitung berdasarkan jarak antar data/objek ke *centroid* masing-masing *cluster* yang sudah ditentukan sejak awal. Data/objek yang paling

dekat dengan centroid maka akan dikelompokkan menjadi satu *cluster* dengan *centroid* tersebut (Purnama, 2019).

Dalam penyelesaiannya, algoritma k-means menghasilkan titik *centroid* yang akan dijadikan sebagai tujuan dari algoritma ini. Saat iterasi k-means berhenti, maka setiap objek dalam dataset menjadi anggota dari suatu *cluster* atau kelompok. Nilai *cluster* ditentukan dengan mencari seluruh objek untuk menemukan *cluster* dengan jarak terdekat ke objek. Sehingga algoritma k-means akan mengelompokkan item dalam suatu dataset ke suatu *cluster* berdasarkan jarak terdekat (Bangoria et al., 2013). Nilai *centroid* awal dipilih secara acak yang akan digunakan sebagai titik pusat awal dan akan dihitung menggunakan rumus euclidean distance (Jollyta et al., 2021).

Euclidean distance merupakan salah satu metode perhitungan jarak yang digunakan untuk mengukur jarak dari 2 (dua) buah titik dalam euclidean space (meliputi bidang Euclidean dua dimensi, tiga dimensi atau bahkan lebih)(Nishom, 2019). Rumus Euclidean Distance untuk mengukur tingkat kemiripan data terdapat pada Pers 2.2 (Nishom, 2019):

$$d(x, y) = |x - y| = \sqrt{\sum_{i=1}^N (x_i - y_i)^2} \quad \text{Pers 2.2 Euclidean Distance}$$

Keterangan

- d = jarak antara x dan y
- x = data pusat *cluster*
- y = data pada atribut
- i = setiap data
- n = jumlah data
- x_i = data pada pusat *cluster* ke i
- y_i = data pada pusat *cluster* ke i

