

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Tinjauan Pustaka**

Menurut penelitian Novendri et al (2019), perancangan sistem informasi inventaris barang pada MTS Nurul Islam dilakukan dengan tujuan untuk mempermudah pencatatan data barang yang masih dikelola secara manual pada MTS Nurul Islam Kota Dumai. Dengan menggunakan bahasa pemrograman PHP dan database MySQL, Sistem Informasi tersebut terbagi atas pencatatan data tempat inventaris, pencatatan barang masuk, pencatatan barang keluar, mutasi barang, serta pembuatan laporan barang masuk dan keluar. Dengan adanya Sistem Informasi Inventaris Barang ini, pekerjaan pendataan inventaris barang pada MTs Nurul Islam Kota Dumai dapat dilakukan dengan cepat dan akurat.

Pada penelitian Kusuma Wardhana et al (2016) dibuat sebuah Sistem Informasi Persediaan Barang Habis Pakai Subbagian Umum Kepegawaian dan Perlengkapan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta. Sistem informasi berbasis website ini mampu menangani pengelolaan persediaan barang habis pakai yang terdiri dari proses penambahan barang, transaksi barang keluar, dan transaksi barang masuk. Selain itu, sistem informasi tersebut dapat menampilkan laporan berdasarkan kategori tertentu yang dapat bermanfaat dalam proses pengambilan keputusan.

Pada penelitian Baihaqi et al (2019) menghasilkan aplikasi sistem informasi persediaan barang habis pakai berbasis SMS *Gateway* dan telah diimplementasikan berbasis *website*. Dalam penelitian ini, menggunakan tiga aktor yaitu admin, petugas bagian umum, dan pimpinan bagian umum. Sistem diakses oleh admin yang menginput data user. Petugas bagian umum menginput data barang masuk, data distribusi, data barang, dan data bagian. Sedangkan pimpinan bagian umum menerima laporan data barang masuk dan laporan data distribusi. Sehingga pengelolaan persediaan barang habis pakai pada kantor Camat Seulimuem Kabupaten Aceh Besar dapat berjalan secara efektif dan efisien.

Pada penelitian Effendy et al (2022) dengan menggunakan *framework* CodeIgniter dihasilkan sebuah sistem informasi *inventory* pada PT. Pertamina Hulu Rokan Regional 1 Zona 4. Dengan adanya sistem informasi tersebut, data peralatan dapat diawasi dan dikontrol jumlah stoknya setiap waktu. Selain itu, masa pakai atau kadaluarsa bagi barang-barang tertentu dapat dipantau secara rutin, sehingga jumlah stok peralatan dan barang habis pakai dapat diatur dan dikelola agar aktivitas produksi perusahaan dapat berjalan dengan efisien dan maksimal.

Pada penelitian ini memiliki perbedaan dengan penelitian yang telah disebutkan diatas. Perbedaan dengan penelitian Novendri et al (2019) terletak pada data tempat inventaris. Pada penelitian ini tidak mengakomodir data tempat inventaris dikarenakan persediaan barang habis pakai pada Kantor Kecamatan Kaliwungu hanya menggunakan satu lemari persediaan saja. Perbedaan dengan penelitian Kusuma Wardhana et al (2016) terletak pada laporan berdasarkan kategori barang. Pada penelitian ini tidak menghasilkan laporan berdasarkan kategori barang. Perbedaan dengan penelitian Baihaqi et al (2019) terletak pada notifikasi barang yang telah habis melalui SMS. Pada penelitian ini tidak mengakomodir notifikasi barang melalui SMS. Perbedaan dengan penelitian Effendy et al (2022) terletak pada data masa habis pakai barang. Pada penelitian ini tidak mengakomodir data masa habis pakai barang karena persediaan barang habis pakai pada Kantor Kecamatan Kaliwungu dapat disimpan dalam waktu yang lama.

## **2.2. Teori Pendukung**

### **2.2.1. Rancang Bangun**

Rancang bangun adalah suatu istilah umum untuk membuat atau mendesain suatu objek dari awal pembuatan sampai akhir pembuatan. Rancang bangun berawal dari kata desain yang artinya perancangan, rancang, desain, bangun. Sedangkan merancang artinya mengatur, mengerjakan atau melakukan sesuatu dan perancangan artinya proses, cara, perbuatan merancang. Dapat disimpulkan arti kata desain adalah proses, cara, perbuatan dengan mengatur segala sesuatu sebelum bertindak atau merancang. Rancang Bangun adalah kemampuan untuk membuat beberapa alternatif pemecahan masalah (Ariansyah, Fajriyah, 2017).

### **2.2.2. Aplikasi**

Menurut Kamus Besar Bahasa Indonesia (KBBI), aplikasi adalah sebuah penerapan dari rancangan sistem yang digunakan untuk pengolahan data dan menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah sebuah program komputer yang diciptakan dengan tujuan untuk mengerjakan dan melaksanakan tugas khusus dari pengguna (Abdurahman & Riswaya, 2014).

### **2.2.3. Monitoring**

Monitoring adalah proses pengumpulan dan analisis informasi berdasarkan indikator yang ditetapkan secara sistematis dan kontinu tentang kegiatan program sehingga dapat dilakukan tindakan koreksi untuk penyempurnaan program kegiatan itu selanjutnya. Pemantauan yang dapat dijelaskan sebagai kesadaran (*awareness*) tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan ke arah tujuan atau menjauh dari itu (Hendini, 2016).

### **2.2.4. Persediaan**

Persediaan adalah salah satu asset lancar yang signifikan bagi perusahaan pada umumnya terutama perusahaan dagang, manufaktur, pertanian, kehutanan, pertambangan, kontraktor bangunan dan penjualan jasa tertentu (Meilano & Chandra, 2020). Persediaan adalah sejumlah barang dalam proses yang dimiliki perusahaan dengan tujuan untuk diproses lebih lanjut (Fajriyah et al., 2017).

### **2.2.5. Barang Habis Pakai**

Barang habis pakai adalah barang yang hanya dapat digunakan sekali dalam pemakaiannya. Hal ini berarti bahwa benda tersebut tidak meninggalkan bekas dalam pemakaiannya. Barang habis pakai tersebut terdiri dari alat tulis kantor, barang cetak, dan barang atau alat pembersih. Barang habis pakai merupakan barang yang pada umumnya digunakan untuk keberlangsungan kegiatan operasional dalam sebuah organisasi. Berdasarkan pernyataan diatas, persediaan barang habis pakai merupakan persediaan yang mengelola barang dalam bentuk

perlengkapan yang digunakan dalam operasional dan aktivitas administrasi sehari-hari (Meilano & Chandra, 2020).

#### **2.2.6. Website**

*Website* merupakan kumpulan dari halaman-halaman web yang berhubungan dengan *file-file* lain yang saling terkait. Halaman-halaman tersebut dapat digunakan untuk menampilkan informasi, teks, gambar, animasi, suara, dan/atau gabungan dari semua itu (Saed Novendri, Saputra, & Eri Firman, 2019).

#### **2.2.7. PHP**

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman untuk membuat *website* yang bersifat *server-side scripting*. PHP bersifat dinamis sehingga dapat dijalankan di berbagai macam sistem operasi seperti *Windows*, *Linux*, dan *Mac OS*. PHP juga mendukung beberapa *web server* lain seperti *Microsoft ISS*, *Caudium*, dan *PWS*. PHP dapat memanfaatkan database untuk menghasilkan halaman *web* yang dinamis (Saed Novendri, Saputra, & Firman, 2019).

#### **2.2.8. XAMPP**

*Xampp* merupakan perangkat lunak bebas yang mendukung banyak sistem operasi dan merupakan kompilasi dari beberapa program. *Xampp* adalah salah satu alat yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket *xampp* sudah terdapat *Apache* sebagai *web server*, *MySQL* sebagai *database*, PHP sebagai *server-side scripting*, *Perl*, *FTP server*, *PhpMyAdmin*, dan berbagai pustaka bantu lainnya (Saed Novendri, Saputra, & Firman, 2019).

#### **2.2.9. HTML**

HTML (*HyperText Markup Language*) adalah salah satu bahasa pemrograman yang digunakan untuk membuat sistem persediaan barang habis pakai ini. HTML dominan dengan menggunakan tanda *tag* < > untuk menyatakan kode-kode yang akan ditafsirkan oleh *browser* agar halaman dapat ditampilkan dan muncul sesuai dengan posisi yang telah diatur. Bahasa HTML ini sendiri digunakan untuk membantu merancang struktur dasar halaman *website* atau bila dianalogikan

HTML merupakan pondasi awal untuk menyusun berdirinya kerangka halaman *website* secara lebih terstruktur sebelum masuk ke tahap desain dan sisi fungsionalitas. HTML nantinya akan dikolaborasikan dengan Bahasa Pemrograman CSS (Sari et al., 2022).

#### **2.2.10. CSS**

CSS (*Cascading Style Sheet*) adalah bahasa pemrograman untuk memberikan tampilan desain yang akan digunakan pada *website* seperti warna, *font*, *outline*, background, menyesuaikan tampilan *website* dengan ukuran layar, dan sebagainya. CSS digunakan pada pembuatan *website* ini adalah untuk berkolaborasi dengan HTML agar dapat menghasilkan tampilan *website* yang menarik (Sari et al., 2022).

#### **2.2.11. Bootstrap**

*Bootstrap* adalah platform CSS (*Cascading Style Sheet*) yang digunakan untuk mendesain situs *website*. *Bootstrap* adalah alat yang dapat digunakan pemrogram saat membangun situs *website*. Misalnya, CSS di *Bootstrap* menyediakan tipe, tombol, navigasi, dan elemen lain yang, bersama dengan *JavaScript*, membuat pengembangan antarmuka menjadi lebih mudah dan lebih stabil. *Bootstrap* juga menyediakan banyak kelas dan *plugin* CSS siap pakai untuk membantu pengembang membuat tampilan dan nuansa situs *website*. Oleh karena itu, *bootstrap* adalah salah satu framework *front-end* yang paling banyak digunakan (Agus Supriatmaja et al., 2022) .

#### **2.2.12. MySQL**

*MySQL* merupakan sebuah perangkat lunak sistem manajemen basis data yang *multithread* dan *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL* AB membuat *MySQL* tersedia sebagai perangkat lunak gratis di bawah lisensi GNU GPL (*General Public License*), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan menggunakan GPL (Hakim et al., 2019).

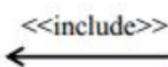
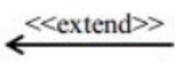
### 2.2.13. UML

UML (*Unified Modeling Language*) merupakan salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan kebutuhan, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Ernawati & Gumelar, 2020).

### 2.2.14. Use Case Diagram

*Use case diagram* merupakan pemodelan untuk kelakuann (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case diagram* dapat dilihat pada Tabel 2.1 (Hendini, 2016).

Tabel 2. 1. Simbol *Usecase Diagram*

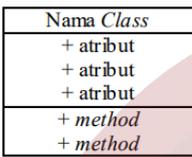
No	Simbol	Keterangan
1.		<i>Actor</i> atau aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem.
2.		<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
3.		<i>Association</i> penghubung antara aktor dengan <i>use case</i> .
4.		<i>Generalization</i> menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
5.		<i>Include</i> , menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
6.		<i>Extend</i> , menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

### 2.2.15. Class Diagram

*Class diagram* Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas

dan constraint yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi *Assosiations*, *Generalitation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau *cardinality*. Simbol *class diagram* dapat dilihat pada Tabel 2.2 (Hendini, 2016).

Tabel 2. 2. Simbol *Class Diagram*

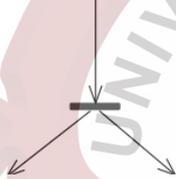
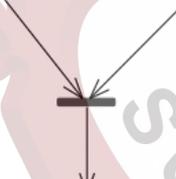
No	Simbol	Keterangan
1.		<p><i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i>. Bagian tengah mendefinisikan <i>property/atribut class</i>. Bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i>.</p>
2.		<p>Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i>. Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.</p>
3.		<p>Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.</p>
4.		<p><i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi</p>

### 2.2.16. Activity Diagram

*Activity diagram* atau diagram aktivitas adalah diagram yang menggambarkan aliran fungsionalitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor. Pada tahap pemodelan sistem, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja sistem dan kejadian (Dirgantara & Suryadarma, 2014).

Dapat ditarik kesimpulan *activity diagram* atau diagram aktivitas adalah menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol *activity diagram* dapat dilihat pada Tabel 2.3.

Tabel 2. 3. Simbol *Activity Diagram*

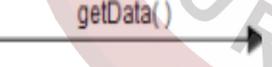
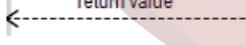
No	Simbol	Keterangan
1.		<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
2.		<i>End Point</i> , akhir aktivitas.
3.		<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis
4.		<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
5.		<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
6.		<i>Decision Points</i> , menggambar kan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .
7.		<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

### 2.2.17. Sequence Diagram

*Sequence diagram* atau dapat disebut dengan diagram urutan adalah diagram yang menunjukkan interaksi antar objek yang diatur dalam urutan waktu. Diagram ini dapat menggambarkan objek yang terlibat dalam skenario sistem. Oleh karena itu untuk menggambarkan *diagram sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah usecase serta metode-metode yang dimiliki kelas

yang diinisiasikan menjadi objek tersebut. Simbol-simbol *sequence diagram* dapat dilihat pada Tabel 2.4 (Saptia Kurnia & Risyda, 2021).

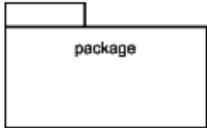
Tabel 2. 4. Simbol *Sequence Diagram*

No	Simbol	Keterangan
1.		<i>Actor</i> merupakan eksternal sistem berupa orang atau sistem lain yang mendapatkan manfaat dari <i>use-case</i>
2.		<i>Object</i> merupakan sesuatu yang ikut dalam rangkaian kegiatan dengan mengirim atau menerima pesan.
3.		<i>Lifeline</i> menunjukkan waktu hidup suatu objek dalam rangkaian kegiatan.
4.		<i>Execution Occurrence</i> menandakan ketika suatu objek mengirim atau menerima pesan.
5.		<i>Request Message</i> Merupakan permintaan informasi dari satu objek ke objek lain
6.		<i>Return Message</i> merupakan permintaan informasi yang dikirim oleh suatu objek

### 2.2.18. Component Diagram

*Component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam suatu sistem. *Component diagram* berfokus pada komponen sistem yang dibutuhkan dan ada di dalam sebuah sistem. Simbol dari *component diagram* dapat dilihat pada Tabel 2.5 (A.S & Shalahuddin, 2013).

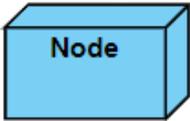
Tabel 2. 5. Simbol *Component Diagram*

No	Simbol	Keterangan
1.		<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen.
2.		<i>Component</i> , yang dimaksud tersebut adalah komponen sistem.
3.		<i>Depedency</i> , kebergantungan antar komponen. Arah panah tersebut mengarah pada komponen yang dipakai.
4.		<i>Interface</i> , sebagai antarmuka komponen agar tidak mengakses langsung komponen.
5.		<i>Link</i> , digunakan sebagai relasi antar komponen.

### 2.2.19. *Deployment Diagram*

*Deployment diagram* digunakan untuk menggambarkan detail bagaimana komponen disusun di infrastruktur sistem. Simbol *deployment diagram* dapat dilihat pada Tabel 2.6 (Hendini, 2016).

Tabel 2. 6. Simbol *Deployment Diagram*

No	Simbol	Keterangan
1.		Pada <i>deployment diagram</i> , komponen-komponen yang ada diletakkan didalam <i>node</i> untuk memastikan keberadaan posisi mereka.
2.		<i>Node</i> menggambarkan bagian-bagian <i>hardware</i> dalam sebuah sistem. Notasi untuk <i>node</i> digambarkan sebagai sebuah kubus 3 dimensi.
3.		Sebuah <i>association</i> digambarkan sebagai sebuah garis yang menghubungkan dua <i>node</i> yang mengindikasikan jalur komunikasi antara elemen-elemen <i>hardware</i> .

### **2.2.20. Framework CodeIgniter**

*Codeigniter* adalah kerangka kerja PHP: *Hypertext Preprocessor* yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi *website* berbasis PHP daripada menulis semua kode program dari awal. Beberapa keuntungan menggunakan *Codeigniter* adalah gratis, ditulis menggunakan PHP 4, ukuran kecil, menggunakan Konsep *Model View Controller* (MVC), *Uniform Resource Locator* (URL) yang sederhana (Kurniadi et al., 2022).

### **2.2.21. Pengujian Sistem dengan Interface Testing**

Pengujian sistem yang digunakan adalah *interface testing*. Pada pengujian ini terdapat dua area yang akan diuji yaitu aplikasi dan web. Pengujian aplikasi dilakukan dengan permintaan tes dikirim dengan benar ke database dan output di sisi klien ditampilkan dengan benar. Jika terdapat kesalahan hanya akan ditampilkan di halaman administrator bukan pada halaman pengguna (Hamilton, 2023).

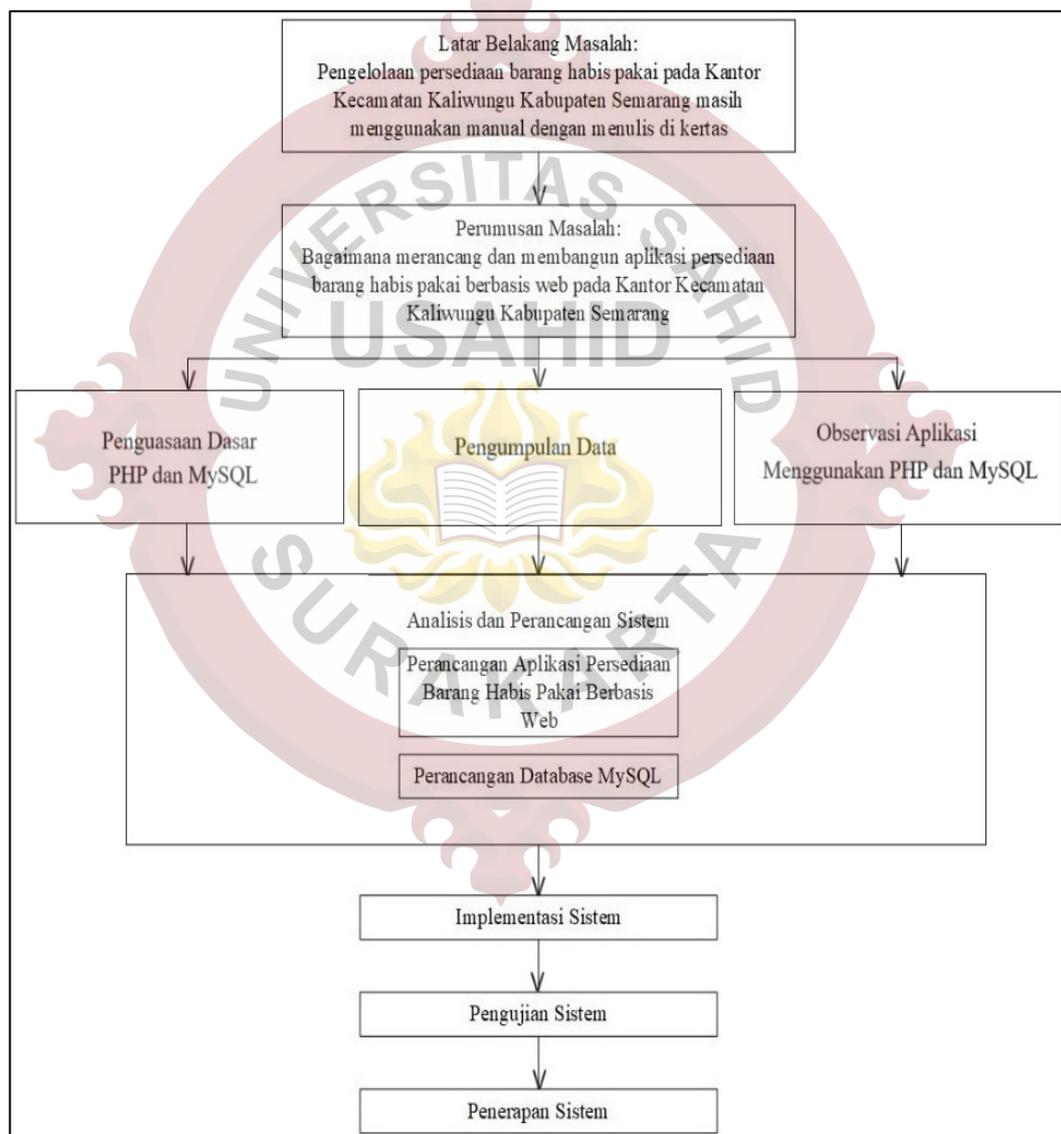
### **2.2.22. Pengujian Sistem dengan User Acceptance Testing**

*User Acceptance Testing* merupakan pengujian yang dilakukan oleh *end user* yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi telah berjalan sesuai dengan kebutuhan/fungsinya. *User acceptance test* menguji yang dilakukan oleh pengguna sistem. Hasil dari pengujian dapat dijadikan bukti bahwa sistem dapat membantu para pengguna (Wahyudi & Alameka, 2023).

## **2.3. Kerangka Pemikiran**

Kerangka pemikiran pada aplikasi *monitoring* persediaan barang habis pakai pada Kantor Kecamatan Kaliwungu Kabupaten Semarang dilatar belakangi oleh pengelolaan persediaan barang habis pakai yang masih menggunakan semi manual dengan menulis di kertas dan direkap menggunakan bantuan Ms. Excel. Permusuan masalah yang didapatkan adalah bagaimana merancangan dan membangun aplikasi persediaan barang habis pakai berbasis web pada Kantor Kecamatan Kaliwungu Kabupaten Semarang? Setelah itu dilakukan penguasaan

dasar PHP dan MySQL, pengumpulan data, dan melakukan observasi aplikasi yang serupa yang menggunakan PHP dan MySQL. Pada tahap analisis dan perancangan sistem terbagi menjadi perancangan aplikasi dan perancangan kebutuhan database. Kemudian dilakukan implementasi sistem dari hasil perancangan aplikasi dan databasenya. Setelah sistem jadi, dilakukan pengujian sistem. Setelah pengujian dilakukan, sistem sudah siap untuk diterapkan di Kantor Kecamatan Kaliwungu Kabupaten Semarang. Kerangka pemikiran dapat dilihat pada Gambar 2.1.



Gambar 2. 1. Kerangka Pemikiran Penelitian

## 2.4. Spesifikasi Alat yang Digunakan

### 2.4.1. Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam penelitian ini dapat dilihat pada Tabel 2.7 berikut:

Tabel 2. 7. Spesifikasi Perangkat Keras yang Digunakan

No	Jenis Perangkat Keras	Keterangan
1.	Laptop	ASUS P1512CEA, Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz, 8GB
2.	Monitor	LG 21"

### 2.4.2. Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam penelitian ini dapat dilihat pada Tabel 2.8 berikut:

Tabel 2. 8. Spesifikasi Perangkat Lunak yang Digunakan

No	Jenis Perangkat Lunak	Keterangan
1.	Sistem Operasi	<i>Windows 11 Home</i>
2.	<i>Web Server</i>	<i>XAMPP v3.2.2</i>
3.	<i>Web Browser</i>	<i>Google Chrome</i>
4.	<i>Text Editor</i>	<i>Sublime Text</i>
5.	Pengolah UML	<i>Astah</i>
6.	Pengolah Perancangan Tampilan	<i>Balsamiq Mockup</i>