

## BAB II

### TINJAUAN PUSTAKA

#### II.1. Penelitian Terdahulu

Penelitian terdahulu dikutip dari beberapa jurnal yang berkaitan dengan penelitian penulis dan berikut adalah beberapa penelitian yang diperoleh :

1. Penelitian Iswanto (2021) mengenai “Perancangan Sistem *Dashboard Inventory* Pada PT. Dinamika Anugerah Dengan Metode FIFO (*First In First Out*)” disimpulkan penelitian ini menggunakan metode *First In First Out* dapat digunakan untuk mengelola *inventory* dengan baik. Penelitian penulis menggunakan metode *First In First Out* untuk permintaan dan penggunaan inventaris.
2. Penelitian Hapsari, dkk (2021) mengenai “Penerapan Teknologi *Enterprise Resource Planning* Menggunakan Modul *Inventory Management* dengan Metode FIFO Berbasis *Website* Pada CV Pandan Mas Sampit” disimpulkan penelitian ini menggunakan metode *First In First Out* dapat digunakan untuk mengelola *inventory management* dengan baik. Penelitian penulis menggunakan metode *First In First Out* untuk permintaan dan penggunaan inventaris.
3. Penelitian Hapsari, dkk (2018) mengenai “Pengembangan Aplikasi Inventaris Dan Peminjaman Peralatan Laboratorium Politeknik Gorontalo” disimpulkan penelitian ini menggunakan metode *First In First Out* dapat digunakan untuk mengelola inventaris dan peminjaman peralatan laboratorium. Penelitian penulis

menggunakan metode *First In First Out* untuk permintaan dan penggunaan inventaris.

4. Penelitian Chandrawati dan Sanjaya (2019) mengenai “Aplikasi Peminjaman Sarana Dan Prasarana Olahraga Berbasis Web Studi Kasus: SMA Negeri 1 Baleendah” disimpulkan penelitian ini mengelola inventaris peminjaman sarana dan prasarana berbasis web. Penelitian penulis mengelola permintaan dan penggunaan inventaris berbasis *client server*.
5. Penelitian Kinaswara, dkk (2019) mengenai “Rancang Bangun Aplikasi Inventaris Berbasis Website pada Kelurahan Bantengan” disimpulkan penelitian ini mengelola inventaris peminjaman sarana dan prasarana berbasis website. Penelitian penulis mengelola permintaan dan penggunaan inventaris berbasis *client server*.
6. Penelitian Oktaviani, dkk (2019) mengenai “Sistem Informasi Inventaris Barang Berbasis Web Pada SMP Negeri 1 Buer” disimpulkan penelitian ini mengelola inventaris barang di sekolah berbasis website. Penelitian penulis mengelola permintaan dan penggunaan inventaris berbasis *client server*.

## **II.2. Landasan Teori**

Berikut ini adalah landasan teori yang berkaitan dengan penelitian yang di ambil dari beberapa jurnal terkait penelitian :

### **II.2.1. Perancangan**

Perancangan merupakan sebuah alur atau proses aplikasi macam-macam teknik prinsip dengan tujuan pendefinisian suatu sistem, perangkat, atau proses detail yang mencukupi untuk

memungkinkan pengaktualan fisiknya. (Saputri, 2020 : 2).

Perancangan adalah tahap setelah analisis dari siklus pengembang sistem : Pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi : “Menggambarkan bagaimana suatu sistem dibentuk”. Perancangan dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Perancangan adalah menentukan bagaimana sistem akan menyelesaikan apa yang mesti diselesaikan. (Hidayatulloh, dkk, 2020 : 20).

## **II.2.2. Aplikasi**

Aplikasi merupakan sebuah subkelas dari perangkat lunak komputer dengan memanfaatkan kemampuan komputer sebagai penerapannya, sebagai penyimpan sesuatu hal, data, permasalahan, pekerjaan kedalam suatu sarana atau bahkan ke dalam sebuah wadah yang dapat dipakai sebagai penerapan dan pengimplementasian terkait hal atau persoalan yang ada sampai beralih menjadi suatu bentuk yang baru tanpa menghapus poin dasar dari berbagai hal diatas, seperti data, persoalan dan keharusan itu sendiri. Dan aplikasi merupakan suatu perangkat komputer siap pakai bagi *user*, seperti mengerjakan suatu tugas yang di input kan oleh pengguna. (Faris dan Wisaksono, 2021 : 62).

Aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan kedalam suatu media yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan yang ada sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal data, permasalahan, dan pekerjaan itu sendiri. (Sani, 2021 : 401).

Aplikasi secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus komputer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan. Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia, Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. (Turaina, dkk, 2021 : 15).

### **II.2.3. Permintaan**

Permintaan merupakan keinginan yang timbul akibat kebutuhan yang disertai dengan adanya kesediaan barang serta kemampuan untuk membeli barang tersebut. Hukum permintaan berbunyi “Jika harga naik, maka jumlah output yang diminta akan turun. Demikian pula sebaliknya”. Dalam ilmu ekonomi, istilah permintaan (*demand*) mempunyai arti tertentu, yaitu jumlah dari keseluruhan barang dan jasa yang diminta oleh seluruh pelaku ekonomi pada berbagai tingkat harga. Jumlah dari banyaknya suatu barang yang diinginkan sehingga dapat dibeli dan dimiliki sesuai dengan harga dan waktu tertentu dengan asumsi halhal lain tetap sama (*ceteris paribus*) dapat diistilahkan sebagai permintaan akan barang maupun jasa. Permintaan dapat dipengaruhi oleh beberapa hal, yaitu harga barang yang bersangkutan, harga barang substitusi atau komplemennya, selera, jumlah penduduk dan tingkat pendapatan. (Wijayanti, dkk, 2021 : 467).

#### **II.2.4. Penggunaan**

Menurut Kamus Besar Bahasa Indonesia (KBBI), arti kata penggunaan adalah proses, cara, perbuatan menggunakan sesuatu. Arti lainnya dari penggunaan adalah pemakaian. Penggunaan Secara Komersial adalah pemanfaatan Ciptaan dan/atau produk Hak Terkait dengan tujuan untuk memperoleh keuntungan ekonomi dari berbagai sumber atau membayar. (Kisnawan, dkk, 2019 : 72).

#### **II.2.5. Inventaris**

Inventaris atau sering disebut persediaan merupakan simpanan barang-barang mentah, material atau barang jadi yang disimpan untuk digunakan dalam masa mendatang atau dalam kurun waktu tertentu. (Nur dan Ropianto, 2020 : 5).

Inventarisasi merupakan suatu kegiatan yang dilakukan untuk mencatat suatu barang keluar dan masuk dan menyusunnya dengan benar yang telah sesuai dengan peraturan yang telah diterapkan. Inventarisasi adalah suatu cara dalam mencatat barang yang dilakukan untuk mendaftar setiap barang yang dimiliki kantor untuk dapat dipakai dalam melaksanakan setiap tugas. (Kisnawan, dkk, 2019 : 72).

#### **II.2.6. Client Server**

*Client server* merupakan metode yang di mana antara *server* dan *client* saling terpisah. *Server* memberikan informasi maupun pelayanan kepada *client*, dan *client* menerima dari fasilitas layanan yang diberikan oleh *server*. Antara komputer *Server* dan komputer *client* dapat saling melakukan komunikasi dengan menggunakan sistem aplikasi jaringan yang disebut dengan *server*. Masing-masing *client* dapat meminta data atau informasi dari *server*. Sistem *client server* didefinisikan sebagai

sistem terdistribusi. *Client Server* merupakan konektifitas pada jaringan yang membedakan fungsi komputer apakah sebagai *client* atau *server*. Arsitektur ini menjadikan sebuah komputer sebagai *server* atau pusat, fungsinya untuk memberikan layanan-layanan kepada terminal-terminal lain (*client*) yang terhubung pada sistem jaringan ini. Didalam sistem ini peneliti menerapkan metode *client server*, di mana perangkat *android* yang digunakan guru untuk peminjaman inventaris merupakan *client* dan komputer yang digunakan tata usaha untuk mengelola data inventaris merupakan *server*. (Hasibuan dan Dalimunthe, 2020 : 153).

#### **II.2.7. Web**

*Website* merupakan sebuah halaman yang terdiri dari beberapa laman sebuah *web* yang saling berhubungan, yang pada umumnya berisikan dengan gabungan data yang disediakan secara perorangan, kelompok atau bahkan lembaga. Sebuah situs *web* biasanya bertempatkan setidaknya pada sebuah *server web* yang dapat diakses melalui sebuah jaringan internet. *Website* juga berisikan sekumpulan halaman yang didalamnya terdapat berbagai keterangan dalam bentuk data digital baik berupa tulisan, gambar, video, audio bahkan animasi yang disediakan menggunakan jalur internet. Fungsi *website* pada penelitian ini untuk menampilkan dan mengelola aplikasi serta data inventaris sebagai *server*. (Faris dan Wisaksono, 2021 : 62).

#### **II.2.8. Java**

*Java* adalah perangkat lunak produksi *Sun Microsystem Inc.*, yang merupakan perangkat lunak pemrograman untuk beberapa tujuan (*multi purpose*), dapat berjalan di beberapa sistem operasi (*multipatform*), mudah dipelajari dan *powerful*.

Aplikasi-aplikasi yang dapat dibuat dengan *Java* meliputi pemrograman *web* (*web programing*), pemrograman desktop (*desktop programing*), pemrograman *handphone/mobile* (*mobile programing*). (Dewi, dkk, 2018 : 101).

## **II.2.9. *Android***

*Android* adalah sebuah sistem operasi untuk perangkat mobile berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Dalam perjalanannya *Android* telah merilis banyak versi tentunya dengan berbagai macam fitur dan perbaikan di tiap versi yang baru. Fungsi *android* pada penelitian ini untuk peminjaman inventaris yang digunakan oleh guru sebagai *client* dan dibuat menggunakan *eclipse*. Berikut versi-versi *Android* yang telah dirilis hingga saat ini :

1. *Android 1.1*
2. *Android 1.5 (Cupcake)*
3. *Android 1.6 (Donut)*
4. *Android 2.0-2.1 (Eclair)*
5. *Android 2.2.3 (Froyo)*
6. *Android 2.3-2.3.7 (Gingerbread)*
7. *Android 3.0-3.2.6 (Honeycomb)*
8. *Android 4.0-4.0.4 (Ice Cream Sandwich)*
9. *Android 4.1-4.3.1 (Jelly Bean)*
10. *Android 4.4 (KitKat)*
11. *Android 5.0 (Lollipop)*
12. *Android 6.0 (Marshmallow)*
13. *Android 7.0 (Nougat)*
14. *Android 8.0 (Oreo)*
15. *Android 9.0 (Pie)*. (Nugraha, dkk, 2020 : 139).

### II.2.10. Eclipse

*Eclipse* merupakan komunitas *open source* yang bertujuan menghasilkan *platform* pemrograman terbuka. *Eclipse* terdiri atas *framework* yang dapat dikembangkan lebih lanjut, peralatan bantu untuk membuat dan mengelola *software* sejak awal hingga diluncurkan. *Eclipse* adalah sebuah IDE (*Integrated Development Environment*) yang digunakan untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform/OS* oleh karena itu dinamakan dengan (*platform-independent*). *Eclipse* memiliki beberapa kelebihan yang membuatnya banyak digunakan dalam pengembangan perangkat lunak diantaranya :

- Bisa dijalankan berbagai sistem operasi seperti *Windows*, *Linux*, *Solaris*, *Mac*, dll.
- Dikembangkan dengan bahasa *java*, namun *Eclipse* juga mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti *C++*, *Python*, *PHP*, dll, yang membuat *Eclipse* disebut juga *multy-language*.
- *Multy-role*, selain sebagai IDE, *Eclipse* juga bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan *web*, dll. (Dewi, dkk, 2018 : 101).

### II.2.11. PHP

PHP merupakan singkatan dari “*Hypertext Preprocessor*”. PHP adalah sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaknya mirip dengan bahasa pemrograman *C*, *Java*, *ASP* dan *Perl* ditambah beberapa fungsi PHP yang Spesifik dan mudah dimengerti. PHP digunakan untuk membuat tampilan *web* menjadi lebih dinamis, dengan PHP anda bisa menampilkan atau menjalankan beberapa *file* dalam 1

*file* dengan cara di *include* dan *require*. PHP itu sendiri sudah dapat berinteraksi dengan beberapa *database* walaupun dengan kelengkapan yang berbeda yaitu seperti *Database Maria*, *MySQL*, *Oracle*. (Rahmasari, 2019 : 414).

PHP adalah bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata *Hypertext Preprocessor*. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*. (Sugijanto, dkk, 2020 : 2).

#### **II.2.12. MySQL**

MySQL merupakan *database engine* atau *server database* yang mendukung bahasa *database* pencarian SQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread*, *multi-user*. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. (Sugijanto, dkk, 2020 : 2).

Definisi MySQL merupakan *software RDMS* (*Relational Database Management System*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak pengguna dan dapat melakukan suatu proses secara sinkron atau bersamaan. (Rahmasari, 2019 : 414).

#### **II.2.13. Unified Modelling Language (UML)**

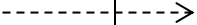
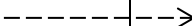
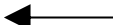
UML yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019 : 39).

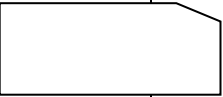
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

1. *Use Case Diagram*

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case diagram* dapat digambarkan dengan sumber-sumber pada Tabel II.1.

Tabel II.1. Simbol *Use Case*

| Gambar                                                                              | Nama                  | Keterangan                                                                                                                                                                                 |
|-------------------------------------------------------------------------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <i>Actor</i>          | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .                                                                                        |
|  | <i>Depedency</i>      | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ). |
|  | <i>Generalization</i> | Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).                                        |
|  | <i>Include</i>        | Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.                                                                                                                           |
|  | <i>Extend</i>         | Menspesifikasikan bahwa <i>use case</i> target                                                                                                                                             |

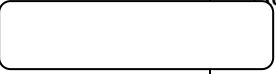
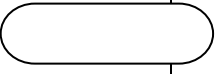
|                                                                                     |                      |                                                                                                                                                   |
|-------------------------------------------------------------------------------------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                     |                      | memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.                                                                  |
|    | <i>Association</i>   | Apa yang mnghubungkan antara objek satu dengan objek lainnya.                                                                                     |
|    | <i>System</i>        | Menspesifikan paket yang menampilkan sistem secara terbatas.                                                                                      |
|    | <i>Use Case</i>      | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .                      |
|    | <i>Collaboration</i> | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
|  | <i>Note</i>          | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi                                                     |

(Sumber : Andikos, 2019 : 39)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

**Tabel II.2. Simbol *Activity Diagram***

| Gambar                                                                              | Nama            | Keterangan                                                                                 |
|-------------------------------------------------------------------------------------|-----------------|--------------------------------------------------------------------------------------------|
|  | <i>Activity</i> | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain. |
|  | <i>Action</i>   | State dari sistem yang mencerminkan eksekusi dari suatu aksi.                              |

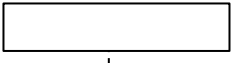
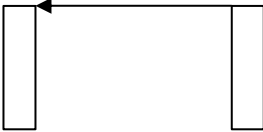
|                                                                                   |                       |                                                                      |
|-----------------------------------------------------------------------------------|-----------------------|----------------------------------------------------------------------|
|   | <i>Initial Node</i>   | Bagaimana objek dibentuk atau diawali.                               |
|  | <i>Activity Final</i> | Bagaimana objek dibentuk dan dihancurkan                             |
|  | <i>Fork Node</i>      | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

(Sumber : Andikos, 2019 : 39)

3. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

Tabel II.3. Simbol *Sequence Diagram*

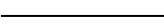
| Gambar                                                                              | Nama            | Keterangan                                                                   |
|-------------------------------------------------------------------------------------|-----------------|------------------------------------------------------------------------------|
|  | <i>Lifeline</i> | Objek entity, antarmuka berinteraksi.                                        |
|  | <i>Message</i>  | Spesifikasi dari komunikasi objek yang memuat informasi tentang apa terjadi. |
|  | <i>Message</i>  | Spesifikasi dari komunikasi objek yang memuat informasi tentang apa terjadi. |

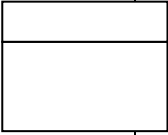
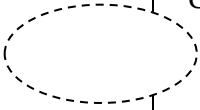
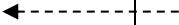
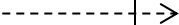
(Sumber : Andikos, 2019 : 39)

4. *Class Diagram* (Diagram Kelas)

*Class* adalah sebuah spesifikasi yang jika diinstansi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

Tabel II.4. *Class Diagram*

| Gambar                                                                              | Nama                  | Keterangan                 |
|-------------------------------------------------------------------------------------|-----------------------|----------------------------|
|  | <i>Generalization</i> | Hubungan dimana objek anak |

|                                                                                     |                         |                                                                                                                                                                    |
|-------------------------------------------------------------------------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                     |                         | ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).                                           |
|    | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.                                                                                                        |
|    | <i>Class</i>            | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.                                                                                            |
|    | <i>Collaboration</i>    | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.                                               |
|    | <i>Realization</i>      | Operasi yang benar-benar dilakukan oleh suatu objek.                                                                                                               |
|  | <i>Depedency</i>        | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
|  | <i>Assocation</i>       | Apa yang menghubungkan antara objek satu dengan objek lainnya                                                                                                      |

(Sumber : Andikos, 2019 : 39)