

BAB II

LANDASAN TEORI

2.1. Penelitian Terdahulu

Pada penelitian ini peneliti mencantumkan penelitian terdahulu yang sesuai sebagai acuan dalam memperkaya teori untuk penelitian yang dilakukan. Berikut ini adalah penelitian terdahulu yang peneliti jadikan sebagai acuan.

Penelitian yang dilakukan oleh Revaldi Fuad Azhar yang berjudul “Sistem Informasi Pemesanan Studio Foto Pada AN Studio berbasis *Website*” bertujuan menyelesaikan masalah terkait transaksi pemesanan foto di AN Studio yang masih dilakukan secara manual dan belum memiliki sebuah sistem. Dalam penelitian ini dibuatkan sebuah sistem yang mempermudah AN Studio dalam melakukan pengolahan data, memberikan informasi pada pelanggan dan membantu mempermudah proses transaksi dengan pelanggan [8].

Kemudian, penelitian yang dilakukan oleh Muhammad Zein Latif Malawat dengan judul “Sistem Informasi Pelayanan Jasa Foto Berbasis Web Pada Mitrasari Foto” bertujuan membangun sebuah sistem informasi yang membantu dalam pengolahan data pelanggan, pelayanan foto, editing foto dan pencetakan foto di Mitrasari Foto sehingga mempermudah dalam pengambilan data karena seluruh data tersimpan pada *database*. Sistem yang dibangun memiliki fitur nomor urut antrian yang mempermudah pelanggan untuk mengikuti prosedur yang ada di Mitrasari Foto [9].

Selanjutnya, penelitian yang dilakukan oleh Roni Samuel Sinaga dengan judul “Sistem Informasi Pemesanan Studio Foto Berbasis Web Pada Fakhri Almubarak Studio” dengan tujuan membangun sistem yang sudah terintegrasi secara baik menggunakan internet sehingga seluruh katalog dapat pelanggan lihat melalui *website*. Pada sistem ini pelanggan bisa melakukan booking online, lalu pembuatan laporan menjadi efektif karena seluruh transaksi tersimpan pada *database* serta menjadi media promosi dari berbagai karya foto yang dimiliki Fakhri Almubarak Studio [10].

Dari beberapa penelitian terdahulu yang telah peneliti kaji seluruhnya menawarkan solusi sebuah sistem secara *online* dengan memanfaatkan *website* karena kelebihan yang bisa diakses di berbagai perangkat. Sistem yang ditawarkan untuk memecahkan masalah terkait pelayanan dan operasional perusahaan yang berfokus pada transaksi dan pembuatan laporan namun belum adanya yang berfokus pada pengerjaan foto dan pendistribusian *file* foto. Maka dari itu dalam penelitian ini peneliti akan berfokus memberikan solusi terbaru.

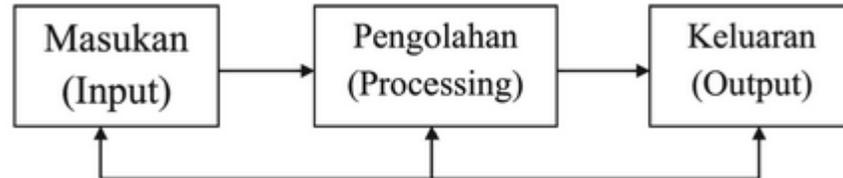
Melihat dari kelebihan dan kekurangan penelitian terdahulu, maka peneliti berusaha menemukan solusi yang terbaik dengan membuat sebuah sistem informasi pelayanan jasa foto yang memiliki fitur transaksi, pembuatan laporan, pemantauan pengerjaan foto dan pendistribusian file foto pelanggan agar mempermudah perusahaan dalam mengerjakan proses edit foto dan juga dalam mendistribusikan hasil foto tersebut. Sistem ini tentunya memiliki *database* yang menyimpan seluruh data dengan menggunakan firebase *realtime database*.

2.2. Teori Dasar

Teori – teori dasar tentunya harus dipahami peneliti dalam melakukan penelitian. Berikut adalah istilah atau teori dasar yang digunakan dalam penelitian ini.

2.2.1. Sistem

Sistem merupakan gabungan dari berbagai elemen yang digunakan untuk melakukan sesuatu, dalam sistem interaksi dilakukan secara teratur membentuk suatu kesatuan. Menurut Suryadharma dan Triyani (2019), Sistem merupakan kumpulan elemen yang saling berinteraksi dalam mencapai tujuan tertentu. Secara umum sistem memiliki unsur penting diantaranya masukan (*input*), pengolahan (*process*) dan keluaran (*output*) [11].



Gambar 2. 1 Model Sistem.

(Sumber : Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML [11,p3])

Sistem bukanlah unsur yang telah teratur, tetapi terdiri dari komponen yang saling berkaitan. Suatu sistem minimal harus memiliki empat komponen yakni masukan, pengolahan, keluaran dan kontrol. Dari semua komponen harus saling berkaitan untuk mencapai tujuan dari organisasi atau perusahaan.

2.2.2. Sistem Informasi

Sistem Informasi merupakan sistem yang mengelola transaksi harian untuk mendukung fungsi operasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk menghasilkan laporan yang diperlukan pihak luar [12]. Sebagai sebuah sistem, Sistem Informasi memiliki komponen yang saling berinteraksi membentuk kesatuan untuk mencapai tujuan. Komponen dalam sistem informasi disebut dengan blok bangunan (*building blok*) yang terdiri dari 6 komponen, yakni:

1. Blok Masukan (*input blok*)

Merupakan inputan data yang masuk dalam sistem informasi berupa metode atau media penangkap data yang dapat berupa dokumen.

2. Blok model (*model blok*)

Kombinasi prosedur, model matematika dan logika untuk memanipulasi data *input* dengan cara yang sudah ditentukan untuk menghasilkan *output* yang diinginkan.

3. Blok keluaran (*output blok*)

Produk atau hasil dari Sistem Informasi yang berupa informasi berkualitas dan berguna untuk pemakai sistem.

4. Blok teknologi (*technology blok*)

Alat yang digunakan Sistem Informasi untuk menerima *input*, menyimpan data, menjalankan model, menghasilkan *output* dan membantu pengendalian seluruh

sistem. Blok ini terdiri dari manusia (*brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*).

5. Blok basis data (*database block*)

Suatu tempat penyimpanan kumpulan data yang saling berhubungan untuk keperluan penyediaan informasi lebih lanjut. Data tersebut perlu disimpan dalam basis data (*database*) dan dalam mengakses serta memanipulasinya diperlukan perangkat lunak yang sebut DBMS (*Database Management System*).

6. Blok kendali (*control block*)

Blok yang melakukan pengendalian terhadap seluruh hal-hal yang dapat merusak suatu sistem seperti bencana alam, sabotase, kegagalan sistem dan lain sebagainya dengan cara melakukan pencegahan dan perbaikan.

Jadi sistem informasi adalah komponen yang saling berkaitan serta gabungan dari teknologi informasi dan sumber daya manusia untuk mengolah *inputan* menjadi suatu *output* yang berguna bagi pihak yang menggunakannya.

2.2.3. Pelayanan

Pelayanan adalah aktivitas orang lain yang bertujuan untuk memenuhi kebutuhan secara tidak langsung. Pelayanan yang diperlukann manusia pada dasarnya ada dua jenis, yaitu layanan fisik yang sifatnya pribadi sebagai manusia dan layanan administrative yang diberikan oleh orang lain selaku anggota organisasi, baik itu organisasi masa atau negara.

Pelayanan merupakan hal yang paling penting dalam suatu perusahaan, karena jika kualitas pelayanan yang diberikan baik maka konsumen akan merasa puas. Keberhasilan suatu produk sangat ditentukan pula baik tidaknya pelayanan yang diberikan perusahaan dalam memasarkan produknya baik itu pelayanan sewaktu penawaran produk, pelayanan kasir hingga pelayanan terhadap pendistribusian produk.

2.2.6. Studio Foto

Studio foto merupakan ruang yang ditata secara khusus untuk aktivitas fotografi. pengaturan ruang yang dimaksud meliputi tata latar, lampu dan segala pendukung yang diperlukan agar fotografer dapat melakukan pemotretan dan pelayanan yang baik. Kegiatan dalam studio foto juga perlu didukung peralatan yang sesuai seperti kamera, lensa, pencahayaan dan background.

2.2.7. Website

Website atau sering disingkat Web adalah suatu layanan pada internet yang menampilkan informasi berupa teks, gambar dan lainnya. Menurut Bekt (2015:35) *Website* merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar, animasi, suara dan gabungan semua komponen baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Perbedaan antara statis dan dinamis dalam *website* dapat dilihat berdasarkan isi konten. Berikut perbedaannya:

1. *Website* Dinamis, yaitu *website* yang seluruh konten atau isinya selalu berubah setiap saat contohnya *website* berita.
2. *Website* Statis, yaitu *website* yang kontennya sangat jarang berubah contohnya *website* kasir.

Pada pembangunan sistem informasi pelayanan ini peneliti akan membuat *website* statis karena dalam sebuah *website* pelayanan konten dan isinya akan jarang mengalami perubahan. Untuk membuat sebuah *website* ada beberapa hal yang diperlukan seperti domain, hosting dan script [13].

2.2.8. Metode Pengembangan *Prototype*

Prototype adalah metode pengembangan perangkat lunak yang menyatukan persepsi antara pengembang dan pengguna agar sistem yang dibangun dapat sesuai dengan kebutuhan pengguna. Dengan teknik *prototype* pengembang bisa membuat sebuah *prototype* sebelum nantinya dibuatkan sistem yang sebenarnya [14].

Dalam penelitian ini peneliti menggunakan *prototype* dengan tujuan agar pengembangan bisa mengumpulkan kebutuhan pengguna terlebih dahulu sehingga sistem yang dibangun dapat menjadi solusi dari permasalahan yang dihadapi pengguna.

Penggunaan metode pengembangan *prototype* ini dilakukan dengan beberapa langkah, yaitu sebagai berikut:

1. Identifikasi Kebutuhan

Merupakan tahap awal dalam pengembangan sistem yaitu melakukan identifikasi kebutuhan pengguna yang meliputi tampilan antarmuka, alur sistem dan *output*.

2. Membuat *Prototype*

Tahapan ini peneliti membuat *prototype* berupa rancangan antarmuka dan alur sistem sebagai gambaran sistem yang dibangun untuk dilakukan uji coba pada pengguna.

3. Menguji *Prototype*

Setelah *prototype* dibuat kemudian dilakukan pengujian secara langsung pada pengguna untuk mengetahui kekurangan pada sistem.

4. Memperbaiki *prototype*

Informasi dari pengujian *prototype* kemudian dianalisis untuk menemukan apa yang menjadi permasalahan dan kekurangan sistem sehingga bisa dibuat suatu solusi agar sesuai dengan kebutuhan pengguna.

5. Mengembangkan versi produk

Setelah *prototype* sesuai dengan kebutuhan pengguna kemudian dikembangkan menjadi versi produk yang akan digunakan oleh pengguna nantinya.

2.2.9. UML

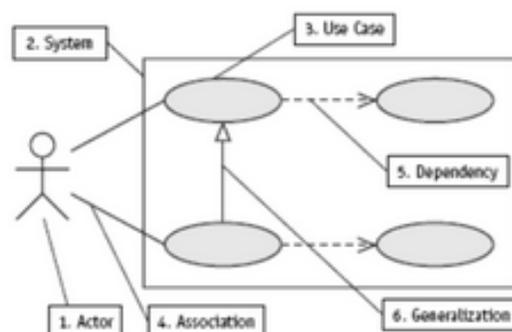
Unified Modeling Language (UML) adalah sebuah bahasa pemodelan sistem yang dirancang khusus untuk pengembangan dan analisis sistem berorientasi objek dan desain. UML merupakan teknik pengembangan sistem yang menggunakan grafis

sebagai alat bantu dalam menspesifikasikan, memvisualisasi dan mendokumentasikan sistem agar mudah dipahami [15].

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan membangun dan dokumentasi dari sistem perangkat lunak [16]. UML memiliki banyak diagram untuk melakukan pemodelan sistem berorientasi objek. Berikut adalah diagram-diagram yang terdapat pada UML.

1. *Use case diagram*

Use case diagram, merupakan diagram yang digunakan untuk menggambarkan hubungan antara sistem dengan aktor secara global. *Use case* juga digunakan untuk menggambarkan fungsi dari sebuah sistem. Karena *use case diagram* hanya menggambarkan sistem secara global, maka elemen-elemen yang digunakan menjadi sedikit [15]. Berikut adalah elemen yang digunakan pada *use case diagram*.



Gambar 2. 2 Elemen *Use case diagram*

(Sumber : Analisis dan Perancangan Sistem Informasi
Manajemen Keuangan [15,p43])

- a) **Aktor**, merupakan elemen pemicu sistem yang bisa berupa orang, mesin ataupun sistem lain.
- b) **Sistem**, merupakan alur proses yang telah dibatasi sesuai dengan yang dibuat.
- c) **Use case**, fungsi yang ada dalam sebuah sistem.
- d) **Association**, elemen yang menggambarkan interaksi antar *use case* dan aktor.
- e) **Dependency**, elemen yang menggambarkan relasi antara dua *use case* yang dibagi menjadi dua macam yaitu *include* sebuah garis yang menghubungkan kebutuhan *use case* pada *use case* lain, dan *extends* yang merupakan garis untuk menghubungkan satu *use case* dengan *use case* lain ketika dibutuhkan sesuai kondisi.
- f) **Generalization**, merupakan proses pewarisan antara dua aktor atau *use case* yang mewarisi *properties* ke aktor atau *use case* lainnya.

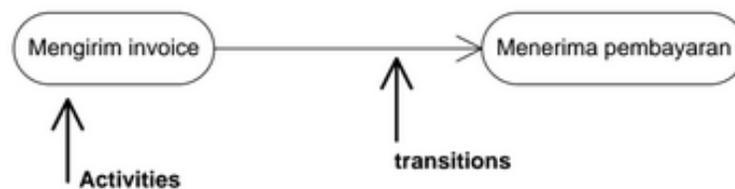
2. *Use case Scenario*

Use case scenario adalah diagram yang menggambarkan logika atau kemungkinan dari sebuah *use case* yang dibuat menjadi beberapa skenario. Untuk menentukan skenario diperlukan perhitungan titik point percabangan terlebih dahulu.

3. *Activity diagram*

Activity diagram adalah diagram yang digunakan untuk menggambarkan alur kerja pada *use case*, logika, proses bisnis dan hubungan antara aktor dengan alur-alur kerja *use case*. Dalam *activity diagram* ada beberapa elemen yang digunakan, yakni.

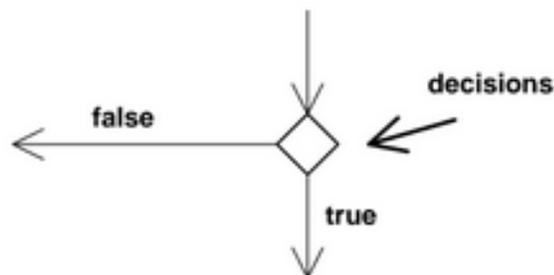
- a) **Activities**, elemen yang menggambarkan aktivitas tertentu yang ada pada sistem.
- b) **Transition**, elemen yang digunakan untuk melakukan peralihan dari aktor ke sistem atau sebaliknya.



Gambar 2. 3 Contoh Elemen *Activities* dan *Transitions*
(Sumber : Analisis dan Perancangan Sistem Informasi

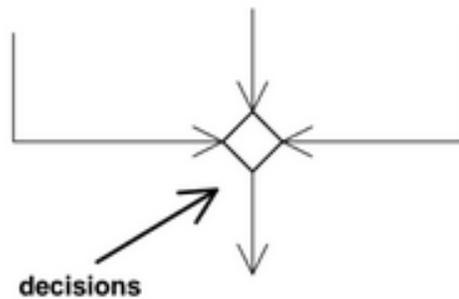
Manajemen Keuangan [15,p55])

- c) **Decisions**, yaitu elemen yang digunakan untuk menggambarkan percabangan logika pada suatu sistem.



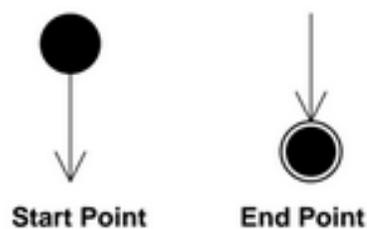
Gambar 2. 4 Contoh Elemen *Decisions*
 (Sumber : Analisis dan Perancangan Sistem Informasi
 Manajemen Keuangan [15,p56])

- d) *Merge Point*, elemen yang digunakan untuk menggabungkan dua proses. Elemen ini merupakan kebalikan dari *decisions*. Berikut adalah contoh elemen *merge point*.



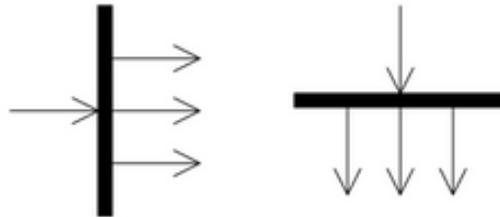
Gambar 2. 5 Contoh Elemen *Merge Point*
 (Sumber : Analisis dan Perancangan Sistem Informasi
 Manajemen Keuangan [15,p56])

- e) *Start Point* dan *End Point*, yaitu elemen untuk memulai dan mengakhiri *activity diagram* dalam suatu sistem.



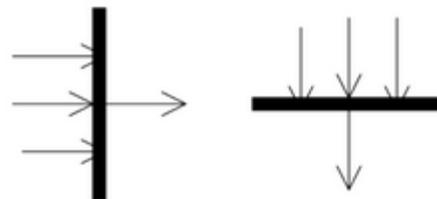
Gambar 2. 6 Contoh *Start Point* dan *End Point*
 (Sumber : Analisis dan Perancangan Sistem Informasi
 Manajemen Keuangan [15,p57])

- f) **Concurrency**, merupakan elemen yang digunakan sebagai percabangan proses bukan logika. Proses yang dimaksud dalam elemen ini bisa dilakukan secara tidak berurutan.



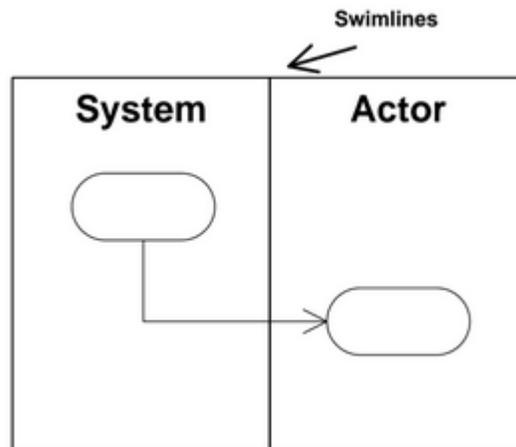
Gambar 2. 7 Contoh Elemen *Concurrency*
 (Sumber : Analisis dan Perancangan Sistem Informasi
 Manajemen Keuangan [15,p57])

- g) **Synchronization**, yaitu elemen yang digunakan untuk menggabungkan beberapa proses atau proses yang dipisah oleh *concurrency*.



Gambar 2. 8 Contoh Elemen *Synchronization*
 (Sumber : Analisis dan Perancangan Sistem Informasi
 Manajemen Keuangan [15,p58])

- h) **Swinlines**, yaitu elemen yang digunakan untuk memisah antara aktor dengan sistem, aktor dengan aktor, ataupun sistem dengan sistem.



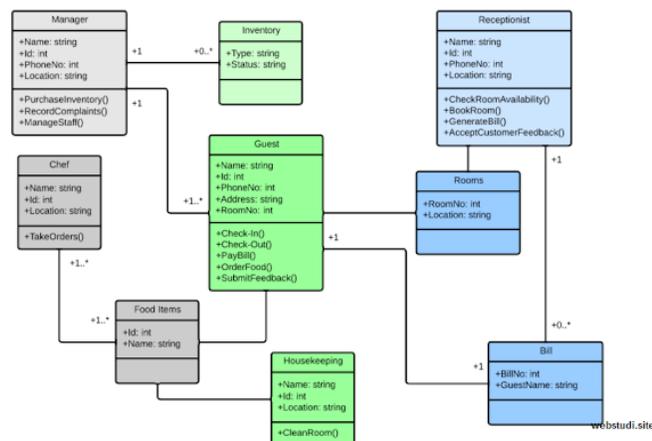
Gambar 2. 9 Contoh Elemen *Swimlines*
 (Sumber : Analisis dan Perancangan Sistem Informasi
 Manajemen Keuangan [15,p58])

4. *Class Diagram*

Class Diagram merupakan struktur statis di dalam sebuah sistem, *Class* mencitrakan sesuatu yang dapat diselesaikan oleh sistem [4]. Dalam menggambarkan spesifikasi sistem *Class Diagram* memiliki tiga area pokok yakni nama, atribut dan metode. Pada atribut dan metoda dapat memiliki beberapa sifat diantaranya *private* yaitu tidak dapat dipanggil dari luar *class* yang bersangkutan, *protected* yaitu sifat atribut dan metoda yang hanya bisa dipanggil di dalam *class* yang sama dan *public* yaitu sifat yang bisa dipanggil oleh siapa saja.

Dalam *Class Diagram* ada hubungan yang digunakan untuk menggambarkan sebuah diagram, yaitu :

- i) Asosiasi, hubungan statis antar class yang secara umum menggambarkan class yang memiliki atribut class lain.
- j) Agregasi, yaitu hubungan yang menyatakan bahwa class menjadi atribut class lain.
- k) Pewarisan, hubungan hirarkis antar class atau class yang dapat diturunkan pada class lain dengan mewarisi seluruh atribut dan metoda class tersebut untuk menambah fungsionalitas.
- l) Hubungan dinamis, rangkaian pesan yang dikirim dari suatu class pada class lain.

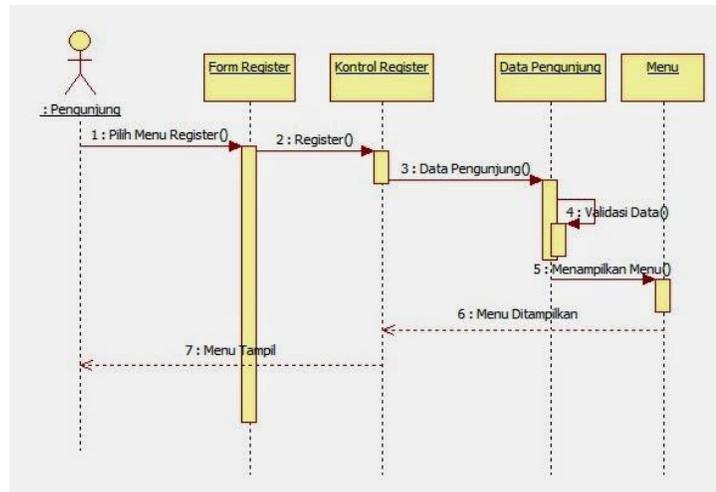


Gambar 2. 10 Contoh Class Diagram
(Sumber : www.webstudi.site)

5. Sequence Diagram

Sequence Diagram adalah sebuah diagram yang menggambarkan interaksi antar objek pada sistem yang bersifat dinamis. *Sequence Diagram* biasa digunakan untuk menggambarkan skenario yang dilakukan sebagai respon dari

sebuah *event* untuk menghasilkan output tertentu. Diawali dengan memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan [17].



Gambar 2. 11 Contoh Sequence Diagram
(Sumber : www.tutorialkampus.com)

2.2.10. Basis Data

Basis data atau *database* tentu sangat penting dalam melakukan pembangunan sistem terutama sistem informasi. Karena dalam pembangunan sistem informasi akan dibutuhkan sebuah penyimpanan untuk seluruh data yang diproses secara terstruktur.

Database sendiri merupakan sebuah kumpulan data yang disimpan dalam sebuah komputer secara terstruktur dan sistematis sehingga bisa digunakan oleh komputer lain untuk mendapatkan data tersebut [17]. Jadi secara konsep *database* adalah kumpulan data yang memiliki hubungan atau relasi satu sama lain secara terstruktur.

Pada pembangunan sistem informasi pelayanan ini digunakan *database* untuk mengelola data agar tidak mengalami redudansi atau pengulangan data yang sama serta untuk menjaga keamanan data.

Dalam pengelolaan *database* digunakan software yang disebut dengan *Database Management System (DBMS)*. DBMS merupakan software yang fungsinya agar *database* dapat digunakan oleh banyak pengguna secara bersamaan dan dalam jumlah besar [17].

2.2.11. Testing Perangkat Lunak

Pengujian perangkat lunak dilakukan untuk mengetahui apakah sistem yang di buat memiliki *error* atau tidak sehingga pengembang bisa memperbaiki terlebih dahulu sebelum diberikan pada pengguna. Sistem dapat diuji melalui dua pendekatan, dengan mencari setiap kesalahan fungsi atau biasa disebut *Black-box Testing* dan pengujian algoritma sistem atau biasa disebut *White-box Testing*.

Dalam penelitian ini pengujian perangkat lunak yang akan dilakukan terbagi menjadi dua macam yaitu:

1. Black-box Testing

Black-box testing merupakan pengujian perilaku atau fungsional, karena berfokus pada fungsi dari perangkat lunak. Dalam melakukan pengujian black-box penguji tidak perlu mengerti cara kerja dari perangkat lunak tapi hanya perlu menebak *inputan* yang membuat perangkat lunak gagal berfungsi [18].

Meskipun *black-box* berfokus pada fungsi dari perangkat lunak namun ada beberapa kesalahan yang bisa diuji dengan *black-box* di antaranya antarmuka , kesalahan *database external*, kesalahan terminasi, kesensitifan sistem terhadap input dan batasan data [19].

2. *White-box Testing*

White-box adalah suatu metode pengujian untuk algoritma atau logika pada suatu perangkat lunak secara spesifik. Pengujian ini memperlihatkan cara kerja dari perangkat lunak secara rinci yang sesuai dengan spesifikasinya [18]. Berbeda dengan *black-box* yang lebih berfokus pada fungsi, *white-box* berfokus pada struktur internal perangkat lunak. Dengan demikian prosedur pengujian dengan *white-box* mengharuskan penguji memahami tentang source code dari perangkat lunak yang sedang diuji [19].

Dengan menggunakan metode *white-box* ini penguji melihat seluruh struktur dan source code dari perangkat lunak yang dibangun untuk mendapatkan beberapa hal di antaranya:

1. Menjamin seluruh jalur independen telah dieksekusi
2. Menguji seluruh logika program
3. Menguji seluruh perulangan yang ada pada perangkat lunak
4. Menguji seluruh struktur internal perangkat lunak.

2.3. Software yang Digunakan

Dalam penelitian ini peneliti menggunakan beberapa *software* yang membantu membangun Sistem Informasi Pelayanan Studio Foto ini. Berikut adalah *software* yang digunakan.

2.3.1. Visual Studio Code

Dalam menulis sebuah *source code* tentunya dibutuhkan sebuah *text editor*. Salah satu *text editor* yang peneliti gunakan untuk membangun sistem ini adalah VS Code (Visual Studio Code). VS Code merupakan *text editor* yang dibuat oleh Microsoft pada 29 April 2015 dan terus berkembang sampai saat ini.

Peneliti menggunakan VS Code dalam menuliskan source code karena banyak sekali fitur yang membantu dalam pembangunan *website*. Berikut merupakan fitur yang dimiliki VS Code [20] :

1. *Cross Platform* – VS code bisa digunakan diberbagai sistem operasi mulai dari macOS, Ubuntu dan Windows. Dengan fitur ini pengembangan aplikasi bisa dilakukan dimanapun.
2. *Lightweight* – fitur ini membuat pengguna dapat mengontrol seluruh bahasa, tema, *commands*, *debugger* dan lainnya sesuai dengan keinginan.
3. *Code Debugging* – membantu dalam melakukan *debugging* atau mendeteksi *bug* pada *source code* dengan cara mengawasi variabel, kode, call stack dan expression.

4. *Powerful Editor* – memberikan fitur yang sangat *powerful* seperti *code snippets, intellisense, auto correct* dan *formatting*.
5. *Integrated Terminal* – terminal yang tersedia langsung pada text editor.
6. *Source Control* – dapat terintegrasi dengan Git yang membantu dalam pengembangan *aplikasi* bersama tim

2.3.2. Xampp

XAMPP adalah *software open source* yang menyediakan *localhost* atau *server local* pada komputer pengguna yang digunakan dalam pembangunan suatu *website* sebelum dilakukan *hosting*. XAMPP digunakan untuk melihat hasil source code atau menguji beberapa fitur didalam *website* tanpa harus terkoneksi pada internet. XAMPP secara gratis dapat di download pada situs resminya <https://www.apachefriends.org/>

Di dalam pembangunan suatu *website* ada beberapa hal yang perlu dipahami dari XAMPP, diantaranya :

1. *Htdocs* – suatu folder yang merupakan penghubung antara *file source code* dengan sebuah *localhost*. Seluruh *folder* dan *file source code* harus tersimpan pada *htdocs* agar bisa ditampilkan kedalam *website*.
2. *Control Panel* – merupakan sebuah panel yang memberikan akses dalam mengatur *database, FTP* dan seluruh kegiatan terkait pembangunan *website*. Fungsi control panel sama seperti *Cpanel* pada sebuah *website online*.
3. *PhpMyAdmin* – fitur XAMPP untuk mengelola data yang tersimpan pada *database*.

Dalam penelitian ini peneliti menggunakan XAMPP karena memiliki kemudahan dalam pembangunan *website*, dimana seluruh kegiatan pembangunan *website* akan dilakukan pada *localhost* sebelum dilakukan *domain* dan *hosting* sehingga dapat menghemat biaya.

2.3.3. PHP

PHP merupakan singkatan dari *Personal Home Page* yaitu bahasa pemrograman yang digunakan untuk *website* pribadi, namun seiring perkembangan PHP berubah menjadi PHP : *Hypertext Preprocessor*. PHP sendiri merupakan salah satu bahasa pemrograman berbentuk *script server-side* yang digunakan untuk pengembangan atau pembangunan *website* oleh karena itu PHP dapat dijalankan menggunakan *browser* [21]. *Script server-side* adalah bahasa pemrograman yang bisa dijalankan pada *server*. PHP juga disebut sebagai bahasa pemrograman *script* karena kodenya bisa disisipkan pada HTML.

PHP memiliki perkembangan mulai dari PHP versi 1 yang dirilis pada tahun 1995 hingga PHP versi 8 yang baru dirilis November 2020.

Di dalam pembangunan sistem ini peneliti menggunakan bahasa pemrograman PHP karena beberapa alasan, diantaranya pengembangan yang cepat, bahasa pemrograman yang *open source* dan *maintenance* yang mudah. Versi PHP yang peneliti gunakan yaitu versi 7.3.6 karena telah terintegrasi dengan *localhost* XAMPP terbaru dan juga *Firestore Database*.

2.3.4. CSS

CSS (Cascading Style Sheet) merupakan intruksi dalam sebuah *website* yang mengatur dan menentukan bagaimana *style* dari sebuah *layout*, *text*, *image* dan tampilan halaman yang akan muncul pada sebuah *website*. Secara definisi CSS adalah salah satu *style sheet language* atau bahasa desain untuk mengatur sebuah halaman *website* dengan sebuah penanda yang disebut *markup language*.

CSS digunakan untuk mendesain tampilan dari halaman HTML, XML dan SVG. Secara teknis CSS akan mengambil alih tampilan seperti layout, text dan warna dengan mendesain dan merubah bentuk keseluruhan tampilan yang dibuat HTML dengan memberi penanda atau *tag* pada html. Dalam penggunaanya CSS memiliki *file* terpisah dari tampilan *website* yang dihubungkan dengan *selector class* pada HTML.

CSS terdapat tiga versi yakni CSS 1, CSS2 dan CSS3. Peneliti menggunakan CSS3 versi terbaru karena telah memiliki fitur dari CSS1 dan CSS2. CSS3 mampu melakukan banyak hal untuk mendesain *website*, mendukung efek seperti tulisan maupun objek yang bisa langsung digunakan, lebih hemat *bandwith* karena akses yang ringan sehingga membuat *website* lebih interaktif.

2.3.5. Bootstrap

Bootstrap adalah sebuah kerangka kerja (*framework*) dari CSS yang digunakan untuk mendesain tampilan *website* menjadi lebih menarik untuk semua perangkat baik komputer atau *smartphone*. Bootstrap menjadi sangat membantu karena telah

menyediakan *template* untuk HTML, CSS dan *Javascript* yang siap pakai untuk membuat *website*.

Dalam penggunaan bootstrap perlu melakukan *instalasi* terlebih dahulu. Ada dua cara untuk *instalasi* yaitu *offline* atau *online*. Seluruh *link* untuk *online* dan *file* pendukung untuk *offline* tersedia di situs resminya yaitu <https://getbootstrap.com/>. Setelah melakukan instalasi kita hanya perlu menambahkan *tag* pada *selector class* HTML untuk menghasilkan *layout*, *grid*, *table*, *navbar* dan lainnya.

Pada pembangunan sistem ini peneliti menggunakan bootstrap agar *website* lebih menarik dan nyaman digunakan. Teknik instalasi yang digunakan yaitu *offline* karena lebih responsif dan tidak mengurangi kecepatan *website*.

2.3.6. Firebase

Firebase adalah *platform* milik Google yang membantu developer dalam mengembangkan aplikasi *mobile* maupun *web*. Firebase memungkinkan developer sistem untuk membuat REST API dengan *client* yang berbeda pada *cloud* yang sama. Firebase memiliki *SDK* yang terintegrasi dengan beberapa bahasa pemrograman seperti Java, Javascript, Kotlin, PHP dan lainnya.

Layanan utama yang dimiliki Firebase yaitu *realtime database*, sebuah *database* yang dikembangkan menggunakan MongoDB atau tipe *database* NoSQL (*No Structured Query Language*). Pada umumnya *database* berbasis tabel, namun di *realtime database* digunakan basis JSON (*JavaScript Object Notation*) yang dapat

disinkronkan pada seluruh penggunaanya. *Realtime database* juga telah mendukung fasilitas *multiplatform* seperti Android, iOS dan Web yang akan saling terhubung secara *realtime* jika ada perubahan pada *database*. Keuntungan menggunakan Realtime database salah satunya tidak memerlukan Query serta data akan tetap terjaga dalam kondisi tidak memiliki koneksi internet [22].

Dalam penggunaan Firebase terdapat manfaat yang memudahkan developer untuk mengembangkan suatu aplikasi. Berikut manfaat utama dari Firebase.

1. Hemat Waktu

Firestore akan menyesuaikan landasannya sehingga para developer bisa lebih fokus pada keinginan *client*.

2. Membantu pengambilan keputusan

Firestore memiliki Google Analytics for Firebase yang membantu dalam menganalisis data yang tersimpan.

3. *Multiplatform*

Firestore dapat berfungsi diseluruh platform seperti Android, iOS dan Web.

4. Bergerak Cepat

API Intuitif dikemas dalam satu SDK sehingga membantu dalam mengembangkan aplikasi dengan cepat.