

BAB II

LANDASAN TEORI

1.1 PERANCANGAN

Perancangan adalah sebuah proses awal dari pembuatan suatu sistem yang berupa tahapan-tahapan yang masih berupa konsep sebagai penggambaran dari sistem nyata yang akan dibangun. Sehingga dapat menjadi acuan atau panduan dalam pembuatan suatu sistem supaya tidak terjadi kesalahan dalam pembuatan sistem yang akan dibangun.

Menurut Soetam Rizky [2], “Perancangan adalah sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta didalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya”.

Muhammad Arif [3] menyatakan bahwa : “Perancangan menggambarkan rencana umum suatu kegiatan rancangan proyek dan aktivitas-aktivitas khusus yaitu teknik atau metode-metode dalam merancang sesuatu”.

Menurut Mahdiana [4] “Tahapan perancangan sistem adalah merancang sistem dengan terperinci berdasarkan hasil analisis sistem, sehingga menghasilkan model sistem baru”.

Berikut tahapan-tahapan perancangan sistem menurut pendapat Mahdiana [4] yaitu :

1. Perancangan Output

Perancangan output tidak dapat diabaikan, karena laporan yang dihasilkan harus memudahkan bagi setiap unsur manusia yang membutuhkan.

2. Perancangan Input

Tujuan dari perancangan input yaitu dapat mengefektifkan biaya pemasukan data, mencapai keakuratan yang tinggi, dan dapat menjamin pemasukan data yang akan diterima dan dimengerti oleh pemakai.

3. Perancangan Proses Sistem

Tujuan dari perancangan proses system adalah menjaga agar proses data lancar sehingga dapat menghasilkan informasi yang benar dan mengawasi proses dari sistem.

4. Perancangan Database

Database sistem adalah mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya.

5. Tahapan Perancangan Kontrol

Tujuan perancangan ini agar keberadaan sistem setelah diimplementasikan dapat memiliki kehandalan dalam mencegah kesalahan, kerusakan, serta kegagalan proses sistem

Berdasarkan beberapa pendapat diatas dapat disimpulkan bahwa perancangan adalah tahapan setelah analisis sistem yang terdiri dari perancangan output, input, proses sistem, database, dan perancangan kontrol dimana tujuannya untuk menghasilkan rancangan yang memenuhi kebutuhan yang ditentukan selama tahap analisis.

1.2 APLIKASI

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang dibuat untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. Aplikasi bisa dikatakan suatu perangkat lunak yang siap pakai dengan menjalankan instruksi-instruksi dari pengguna (user), aplikasi banyak diciptakan untuk membantu berbagai keperluan.

Menurut Syahrial Chan [5] “Aplikasi adalah koleksi window dan objek-objek yang menyediakan fungsi untuk aktivitas user, seperti pemasukan data, proses, dan pelaporan”.

Menurut Maimunah dkk. [6] “Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu”.

Menurut Eri Haeril Jana, & Tri Ginanjar Laksana [7] “Aplikasi adalah kumpulan dari unsur-unsur atau elemen-elemen yang saling berkaitan atau berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu”.

Menurut Nazrudin Safaat H [8] “perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.”

Berdasarkan uraian para ahli diatas dapat disimpulkan bahwa Aplikasi adalah sebuah perangkat lunak yang terdiri dari kumpulan unsur-unsur atau elemen yang saling berinteraksi yang memanfaatkan kemampuan komputer untuk melakukan suatu tugas yang diinginkan oleh pengguna.

1.3 SUMBANGAN PEMBINAAN PENDIDIKAN (SPP)

SPP atau Sumbangan Pembinaan Pendidikan merupakan biaya pendidikan yang dibayarkan oleh siswa dan siswi kepada bendahara sekolah yang selanjutnya akan diserahkan kepada yayasan untuk keperluan dan pembinaan pendidikan di sekolah. Besarnya SPP yang harus dibayarkan berbeda-beda untuk tiap kelasnya.

Dana sumbangan bulanan tersebut akan dialokasikan oleh sekolah yang bersangkutan untuk membiayai berbagai keperluan atau kebutuhan sekolah supaya kegiatan belajar mengajar disekolah dapat berjalan lancar dengan adanya bantuan dari dana sumbangan tersebut. Berikut definisi SPP menurut para ahli:

Menurut Nuri Piyanto dan Dewi Soyusiawaty [9] “SPP (Sumbangan Pembinaan Pendidikan) adalah dana yang harus dibayarkan oleh siswa dan akan dipergunakan untuk pembinaan pendidikan di sekolah. Besarnya SPP yang harus dibayarkan berbeda-beda untuk tiap kelasnya”.

Menurut Imam Soleh Ma’rifati [10] “SPP adalah sumbangan pembinaan pendidikan yang dibayarkan oleh siswa disekolah-sekolah.

Menurut Abdillah Baraja [11] dalam jurnalnya “SPP itu kewajiban yang harus dibayar oleh siswa kepada lembaga pendidikan, untuk kelancaran proses belajar mengajar yang meliputi biaya SPP, biaya uang gedung, biaya registrasi dan biaya-biaya lainnya”.

Rohmat Taufiq et al. [12] menyatakan “SPP (Sumbangan Penunjang Pendidikan) adalah iuran atau pembayaran setiap bulan dari siswa yang menjadi kewajiban bagi siswa disekolah. Pembayaran SPP diambil berdasarkan kesepakatan rapat Komite sekolah dan orang tua siswa”

Berdasarkan uraian para ahli diatas dapat disimpulkan bahwa SPP adalah kewajiban yang harus dibayar oleh siswa dan siswi setiap bulannya. Besarnya SPP yang harus dibayarkan berbeda-beda untuk tiap kelasnya.

1.3.1 Pihak Pihak Yang Terlibat Dalam Pembayaran SPP

1. Wali Murid (Orang Tua Siswa)
Wali Murid adalah orang yang wajib membayar SPP setiap bulannya.
2. Bendahara
Bendahara adalah orang yang bertugas melakukan pencatatan pembayaran SPP bulanan dan mengelola laporan pembayaran yang telah dibayar oleh wali murid sesuai dengan kelasnya.
3. Yayasan
Yayasan sebagai penerima seluruh laporan pembayaran SPP bulanan yang telah dikelola oleh bendahara.

1.4 WEBSITE

Website merupakan sekumpulan halaman yang berhubungan yang dapat diakses *public* melalui *world wide web*. Suatu *website* biasanya mengandung teks,

gambar, suara bahkan video interaktif dan mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah browser.

Menurut Kustiyahningsih dan Devie [13] “*Website* merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung dengan fasilitas *hypertext* untuk menampilkan data berupa teks, gambar, suara, animasi dan multimedia lainnya”.

Menurut Ippho Santoso dalam Rahmadi [14] “Membagi *website* menjadi golongan kanan dan golongan kiri. Dalam *website* dikenal dengan sebutan *website* dinamis dan *website* statis”.

1. *Website* statis

Website statis adalah *website* yang mempunyai halaman konten yang tidak berubah-ubah.

2. *Website* dinamis

Website dinamis merupakan *website* yang secara struktur ditujukan untuk update sesering mungkin.

Dari uraian teori diatas penulis menarik kesimpulan bahwa *website* adalah kumpulan halaman-halaman yang dapat menampilkan teks, gambar, animasi, video interaktif, maupun suara yang masing-masing dihubungkan dengan jaringan-jaringan halaman. *Website* dibagi menjadi dua golongan yaitu *website* statis dan *website* dinamis.

1.5 ALAT BANTU PEMODELAN SISTEM

Alat bantu pemodelan sistem merupakan alat yang digunakan untuk merancang sistem yang baru yang akan digunakan untuk memecahkan masalah yang terjadi. Alat bantu pemodelan sistem kebanyakan dibuat dalam bentuk struktural atau berorientasi objek. Penulis disini menggunakan alat pengembangan sistem berorientasi objek dikarenakan dalam pembuatan sistem menggunakan

objek-objek yang saling berkaitan. Alat pengembangan sistem yang digunakan antara lain *use case diagram*, *class diagram*, *activity diagram*.

1.5.1 Use Case Diagram

Use Case diagram adalah pola atau gambaran berbentuk diagram yang menggambarkan hubungan suatu sistem yang tengah di buat. Dalam penggambarannya, sistem yang dibuat harus berada didalam kotak sistem dan memiliki minimal satu aktor yang berada di luar sistem. Use case diagram sangat membantu dalam penyusunan kebutuhan (requirement) sebuah sistem dan perancangan semua fitur- fitur yang terdapat dalam sistem.

Rian et al. [15] Menyatakan bahwa “*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”.

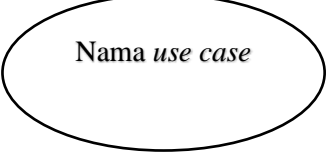
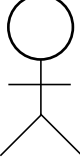




Menurut Murad [16] “Diagram *Use Case* adalah diagram yang bersifat status yang memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini memiliki 2 fungsi, yaitu mendefinisikan fitur apa yang harus disediakan oleh sistem dan menyatakan sifat sistem dari sudut pandang user”.

Menurut Rosa A.S dan M.Shalahuddin [17] : “*Use Case Diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang menggunakan fungsi tersebut”.

Dari beberapa pengertian diatas dapat diambil kesimpulan bahwa *Use Case Diagram* merupakan pemodelan untuk menggambarkan interaksi antara pengguna atau aktor dengan sistem yang ingin kita buat. Berikut adalah contoh simbol-simbol yang ada pada *Use Case Diagram*:

Tabel 2. 1 Simbol-simbol *Use Case Diagram*

(Rosa A.S dan M.Shalahuddin [17])

| NO | Simbol | Deskripsi |
|----|---|---|
| 1 | <p data-bbox="395 365 518 398"><i>Use Case</i></p>  <p data-bbox="467 454 651 488">Nama use case</p> | <p data-bbox="794 365 1449 566">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p> |
| 2 | <p data-bbox="395 622 563 656">Aktor /<i>Actor</i></p>  <p data-bbox="499 891 659 925">Nama Aktor</p> | <p data-bbox="794 622 1449 981">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor</p> |
| 3 | <p data-bbox="395 1014 675 1048">Asosiasi / <i>association</i></p>  | <p data-bbox="794 1014 1449 1149">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p> |
| 4 | <p data-bbox="395 1182 619 1216">Ekstensi / <i>extend</i></p> <p data-bbox="499 1283 659 1317"><<<i>extend</i>>></p>  | <p data-bbox="794 1182 1449 1541">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p> |
| 5 | <p data-bbox="395 1563 762 1597">Generalisasi / <i>generalization</i></p>  | <p data-bbox="794 1574 1449 1765">Hubungan generalisasi dan spesialisasi (umum-khusus) Antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p> |
| 6 | <p data-bbox="395 1787 699 1865">Menggunakan / <i>include</i> / <i>uses</i></p> <p data-bbox="499 1899 659 1933"><<<i>include</i>>></p>  | <p data-bbox="794 1798 1449 1989">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i>.</p> |

1.5.2 Class Diagram

Class diagram merupakan diagram yang selalu ada dipermodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antara *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

Menurut Rosa dan Shalahuddin [17], “diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Kelas memiliki apa yang disebut atribut pola dan metode atau operasi :

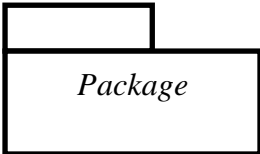
1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi yang dimiliki oleh suatu kelas.

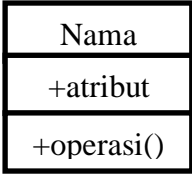
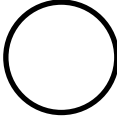






Haqi dan Setiawan [18] mendefinisikan bahwa “*Class Diagram* yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan”.

Sehingga dapat disimpulkan bahwa diagram kelas atau *class diagram* adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2. 2 Simbol-simbol *Class Diagram*

(Rosa A.S dan M.Shalahuddin [17])

| No | Nama Simbol | Simbol | Deskripsi |
|----|----------------|---|---|
| 1 | <i>Package</i> |  | Package merupakan sebuah bungkusan dari satu atau lebih kelas |

| | | | |
|---|---------------------------------------|---|---|
| 2 | Kelas |  | Kelas pada struktur sistem |
| 3 | Antarmuka / <i>interface</i> |  Nama Interface | Sama dengan konsep interface dalam pemrograman berorientasi objek |
| 4 | Asosiasi / <i>association</i> |  | Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity. |
| 5 | Asosiasi berarah / <i>directed</i> |  | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity |
| 6 | Generalisasi |  | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus) |
| 7 | Kebergantungan / <i>dependency</i> |  | Relasi antarkelas dengan makna kebergantungan antar kelas |
| 8 | Agregasi / <i>aggregation</i> |  | Relasi antarkelas dengan makna semua bagian (<i>whole-part</i>) |
| 9 | Komposisi / <i>Composition</i> |  | Suatu kelas merupakan bagian yang wajib dari kelas yang lain. Dimana keberadaan kelas tersebut mempengaruhi keberadaan kelas yang lain. |

1.5.3 Activity Diagram

Activity diagram digunakan untuk memodelkan aspek dinamis dari sistem. *Activity diagram* secara esensial mirip diagram alir atau *flowchart*, memperlihatkan aliran kendali dari suatu aktivitas ke aktivitas lainnya.

Menurut Murad [16], “*Activity diagram* merupakan diagram yang bersifat dinamis. *Activity diagram* adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem dan berfungsi untuk menganalisa proses”.

Menurut Rosa dan Shalahuddin [17], Diagram aktivitas juga banyak digunakan untuk mendefinisi hal-hal berikut:


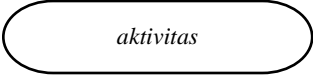
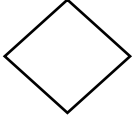


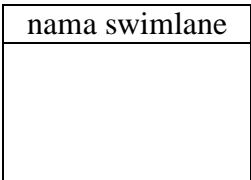
1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem / user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kamus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Dari pengertian diatas dapat ditarik kesimpulan bahwa *Activity diagram* digunakan untuk menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. *Activity diagram* sangat bermanfaat dalam memodelkan sebuah proses untuk membantu dalam memahami prosesnya secara keseluruhan. Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada sistem itu dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan model bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2. 3 Simbol-simbol *Activity Diagram*

(Rosa A.S dan M.Shalahuddin [17])

| Simbol | Deskripsi |
|--|---|
| Status Awal  | Status awal aktivasi sistem, sebuah diagram aktivitas memiliki sebuah status awal berbentuk lingkaran penuh |
| Aktivitas  | Aktivitas yang dilakukan sistem. Diawali dengan kata kerja |
| Percabangan/Decision  | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
| Penggabungan  | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu |
| Status Akhir  | Status akhir dari sebuah sistem yang berjalan. Sebuah diagram aktivitas harus memiliki sebuah status akhir |
| Swimlane  | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

1.6 ALAT BANTU PEMBUATAN PROGRAM

Alat bantu pembuatan program merupakan aplikasi yang digunakan untuk pembuatan program yang digunakan sesuai dengan kebutuhan penulis dimana

penulis menggunakan *Visual Studio Code*, *MariaDB*, *PHP*, *XAMPP*, *Figma* dan *Laravel*.

1.6.1 Visual Studio Code

“Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go, .NET)”[19].

Berdasarkan dari kutipan resmi diatas *Visual Studio Code* merupakan sebuah aplikasi atau *software* yang dikembangkan oleh Microsoft sebagai editor kode gratis yang dapat membantu programmer dalam memulai pengkodean dengan cepat dan membuat kode dalam bahasa pemrograman apa pun, tanpa beralih editor, visual studio code tersedia untuk Windows, macOS, dan juga Linux. *Visual Studio Code* memiliki dukungan untuk banyak bahasa dan runtime lain seperti C++, C#, Java, Python, PHP, Go, .NET.

Berdasarkan [20] *Visual Studio Code* adalah editor *source code* atau *text editor* yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS yang mendukung program bahasa PHP.

Berdasarkan definisi diatas dapat disimpulkan bahwa *Visual Studio Code* merupakan sebuah aplikasi atau *software* yang dikembangkan oleh Microsoft untuk Windows, MacOS dan Linux yang digunakan untuk melakukan pengembangan aplikasi.

1.6.2 MariaDB

“MariaDB Server is one of the most popular open source relational databases. It’s made by the original developers of MySQL and guaranteed to stay open source. It is part of most cloud offerings and the default in most Linux distributions”[21].

Berdasarkan dari kutipan resmi diatas *MariaDB* Server adalah merupakan salah satu server database yang paling populer di dunia. *MariaDB* ini dibuat oleh pengembang asli MySQL dan dijamin akan tetap *open source*. *MariaDB* dikembangkan sebagai perangkat lunak *open source* dan sebagai basis data relasional, ia menyediakan antarmuka SQL untuk mengakses datanya. Versi terbaru *MariaDB* sekarang juga telah menyertakan fitur *GIS* dan *JSON*.

Muhammad Raja Fadhilah et al [22] menyatakan bahwa *“MariaDB* adalah sistem manajemen database relasional yang dikembangkan dari MySQL yang dimaksudkan untuk tetap bebas di bawah *General Public License* (GPL).pengembangan dilakukan oleh beberapa orang yang sebelumnya berkontribusi untuk MySQL.”

Nurrochman Ferdiansyah et al [23] mendefinisikan *“MariaDB* adalah *relational Database Management System* (RDBMS) yang bersifat *open source* dan dikembangkan oleh pengembang yang sama dari MySQL. MySQL sendiri merupakan DBMS yang sangat populer digunakan dan saat ini telah diakuisisi oleh perusahaan Oracle.”

Menurut Jeffry [24] *“MariaDB* memuat beberapa database yang sudah ada yang digunakan oleh *MariaDB* sendiri untuk menyimpan metadata seperti informasi tentang basis data, table, kolom, penggunaan, hak akses, log dan sebagainya.”

Berdasarkan pendapat diatas, dapat disimpulkan bahwa *MariaDB* adalah sistem database tipe MySQL dan bersifat *opensource*, dikembangkan oleh pengembang asli dari MySQL.

1.6.3 *PHP*

“A popular general-purpose scripting language that is especially suited to web development. Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world”[25].

Berdasarkan dari kutipan resmi diatas *PHP* adalah sebuah bahasa pemrograman yang umum dipakai dalam mengembangkan suatu website. Secara sederhana, *PHP* merupakan tool bagi pengembangan web dinamis. *PHP* sangat populer karena memiliki fungsi built in lengkap, cepat, mudah dipelajari, dan bersifat gratis. Skrip *PHP* cukup disisipkan pada kode *HTML* agar dapat bekerja.

Menurut Supono & Putratama [26] “*PHP (PHP: hypertext preprocessor)* adalah suatu bahasa pemrograman yang digunakan untuk menterjemahkan basis kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat server-side yang ditambahkan ke *HTML*”.

Menurut Arief [27] “*PHP* adalah Bahasa server-side-scripting yang menyatu dengan *HTML* untuk membuat halaman web yang dinamis. Karena *PHP* merupakan server-side-scripting maka sintaks dan perintah-perintah *PHP* akan diesksekusi diserver kemudian hasilnya akan dikirimkan ke browser dengan format *HTML*”.

Menurut Anhar [28] “*PHP* adalah bahasa pemrograman web server-side yang bersifat open source, *PHP* juga merupakan script yang terintegrasi dengan *HTML* dan berada pada server (server side *HTML* embedded script). *PHP* juga merupakan script yang digunakan untuk membuat halaman website yang sangat dinamis, dinamis berarti halaman tampilan yang akan ditampilkan dibuat saat halaman itu diminta oleh client”.

Berdasarkan beberapa pendapat diatas dapat disimpulkan bahwa *PHP* merupakan suatu bahasa pemrograman web *server-side* yang digunakan untuk membuat halaman website yang dinamis.

1.6.4 XAMPP

“XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use”[29].

Berdasarkan dari kutipan resmi diatas XAMPP merupakan pengembangan PHP paling populer. XAMPP adalah distribusi Apache yang sepenuhnya gratis dan mudah dipasang yang berisi MariaDB, PHP, dan Perl. Paket open source XAMPP telah diatur agar sangat mudah dipasang dan digunakan.

Purbadian [30] berpendapat bahwa “XAMPP merupakan suatu software yang bersifat *open source* yang merupakan pengembangan dari LAMPP (Linux, Apache, MySQL, PHP dan Perl)”.

Menurut Pratama, I Putu Agus Eka [31] “XAMPP adalah aplikasi web server bersifat instan (siap saji) yang dapat digunakan baik di sistem operasi Linux maupun di sistem operasi Windows”.

Berdasarkan beberapa pendapat diatas dapat disimpulkan bahwa XAMPP merupakan *tool* atau *software* yang bersifat *open source* yang menyediakan paket perangkat lunak seperti Apache, MySQL, dan lain-lain ke dalam satu buah paket sehingga kita tidak perlu lagi untuk menginstal perangkat lunak tersebut.

1.6.5 Figma

“Figma is a browser-based UI and UX design application. The story of the web-based design tool—built to make design accessible to everyone”[32].

Berdasarkan dari kutipan resmi diatas Figma adalah salah satu tools atau aplikasi desain *UI/UX* berupa website yang terhubung dengan cloud sehingga bisa

digunakan kapanpun dan dimanapun melalui internet dan dapat digunakan pada windows ataupun Mac OS untuk membuat prototype aplikasi serta berbagai desain lainnya. Aplikasi ini berbasis vektor, sehingga memang sangat cocok digunakan untuk membuat user interface aplikasi atau website

1.6.6 Laravel

“Laravel is a web application framework with expressive, elegant syntax. We’ve already laid the foundation — freeing you to create without sweating the small things”[33].

Berdasarkan dari kutipan resmi diatas *Laravel* adalah framework aplikasi web dengan sintaks yang ekspresif dan elegan. Dalam mengembangkan sebuah aplikasi terutama yang berfokus pada pengembangan website diperlukan sebuah *Framework* agar pengkodean menjadi lebih mudah. Framework yang digunakan ada begitu banyak salahsatunya adalah *Laravel*.

Terdapat beberapa fitur pada *laravel* seperti :

1. MVC laravel

MVC adalah pola desain arsitektur yang membantu mengembangkan aplikasi web lebih cepat. MVC adalah singkatan dari *Model-View-Controller*.

2. Blade

Blade adalah mesin templating sederhana namun kuat yang disertakan dengan Laravel. Tidak seperti beberapa mesin templating PHP, Blade tidak membatasi kita untuk menggunakan kode PHP biasa di templat

kita. Faktanya, semua template Blade dikompilasi ke dalam kode PHP biasa dan di-*cache* sampai dimodifikasi, yang berarti Blade pada dasarnya menambahkan nol overhead ke aplikasi kita. File template blade menggunakan *.blade.php* ekstensi file dan biasanya disimpan dalam *resources/views* direktori.

3. Database

Laravel membuat interaksi dengan database menjadi sangat sederhana di berbagai database yang didukung menggunakan SQL mentah, pembuat kueri yang lancar, dan ORM Eloquent.

4. Job queues

Laravel memungkinkan kita dengan mudah membuat pekerjaan antri yang dapat diproses di latar belakang. Dengan memindahkan tugas yang membutuhkan banyak waktu ke antrean, aplikasi kita dapat merespons permintaan web dengan sangat cepat dan memberikan pengalaman pengguna yang lebih baik kepada pelanggan.

5. Notifications

Laravel menyediakan dukungan untuk mengirim notifikasi di berbagai saluran pengiriman, termasuk email, SMS (via Vonage, sebelumnya dikenal sebagai Nexmo), dan Slack. Selain itu, berbagai saluran notifikasi buatan komunitas telah dibuat untuk mengirim notifikasi melalui lusinan saluran yang berbeda. Pemberitahuan juga dapat

disimpan dalam database sehingga dapat ditampilkan di antarmuka web.

6. Mail

Laravel menyediakan API email yang bersih dan sederhana yang didukung oleh komponen Symfony Mailer yang populer. Laravel dan Symfony Mailer menyediakan driver untuk mengirim email melalui SMTP, Mailgun, Postmark, Amazon SES, dan sendmail, memungkinkan kita untuk mulai mengirim email dengan cepat melalui layanan lokal atau berbasis cloud pilihan.

1.7 PENELITIAN SEJENIS

Penelitian Sejenis merupakan tinjauan penelitian yang sejenis dengan penelitian yang diambil oleh penelitian sebagai acuan atau referensi untuk perancangan aplikasi. Dan penelitian sejenis dapat dilihat pada tabel di bawah ini :

Tabel 2. 4 Penelitian Sejenis

| Penulis dan Tahun | Judul Penelitian | Metode Penelitian | Hasil Penelitian |
|---|--|--|--|
| Abdur Rochman, Achmad Sidik, Nada Nazahah, 2018 | Perancangan Sistem informasi Administrasi Pembayaran SPP Siwa Berbasis Web Di SMK Al-Amanah. | <i>Object Oriented Analysis Design(OOAD)</i> menggunakan <i>Unified Modelling Language (UML)</i> | Hasil penelitian ini ialah meningkatnya keakuratan, kecepatan, serta ketetapan dalam pembuatan laporan sehingga mengurangi kesalahan dalam laporan pembayaran. |

| | | | |
|--|---|---|---|
| Yolanda Herlita, Ani Oktarini Sari, Eva Zuraidah. 2021 | Perancangan Sistem Informasi Pembayaran Spp Berbasis <i>Website</i> Pada Sma Fajrul Islam Jakarta. | <i>Waterfall</i> | Hasil dari penelitian ini ialah, <i>website</i> ini dapat membantu mempermudah siswa dalam melakukan pembayaran SPP di sekolah, dengan dibangunnya sistem informasi pembayaran sumbangan pebinaan pendidikan ini diharapkan dapat mempermudah pencatatan dan pencarian data pembayaran dengan menggunakan sistem terkomputerisasi. |
| Danang Rifai, Sendy Zul Friandi, Taufan Aditya Pratama. 2018 | rancangan sistem aplikasi pembayaran spp berbasis <i>website</i> dengan menggunakan metode <i>framework yii</i> (Studi Kasus di SDS Mulya Asri Kab. Tangerang). | Metode <i>Framework yii</i> | Hasil penelitian ini ialah, dengan merancang sistem aplikasi pelayanan kasir berbentuk <i>website</i> , maka pelayanan pembayaran SPP terhadap orang tua siswa menjadi efektif secara komputerisasi. Dan demikian maka SDS Mullya Arsi menjadi Sekolah Dasar yang Berstandar Pelayanan dengan sistem Komputer di Kabupaten Tangerang. |
| Indri Pujianti 2021 | Perancangan Sistem Informasi Pembayaran SPP Pada SMA Negeri 6 Tasikmalaya. | <i>Unified Modelling Language (UML)</i> | Hasil dari penelitian ini adalah memberikan kemudahan pada pihak sekolah dalam pencatatan data pembayaran SPP siswa, membantu pihak sekolah dalam pembuatan laporan mengenai |

| | | | |
|--|--|--|---|
| | | | pembayaran SPP siswa, rancangan <i>web</i> sebagai solusi menyelesaikan permasalahan yang terjadi pada sistem pembayaran SPP. |
|--|--|--|---|

Berdasarkan tabel 2.4 penelitian sejenis, maka dapat disimpulkan bahwa sistem yang dirancang memiliki perbedaan yaitu lokasi dari objek penelitian tidak sama, pada penelitian ini peneliti menggunakan SMK PGRI 2 Kota Jambi sebagai objek penelitian, dan *framework* yang digunakan berbeda yaitu peneliti menggunakan *framework php Laravel 9*.