

BAB II TINJAUAN PUSTAKA

2.1. Sistem

Menurut (Ikhsan & Ramadhani, 2020) Sistem adalah susunan komponen yang bekerja sama untuk mencapai tujuan tertentu. Agar suatu sistem dapat mencapai tujuannya, bagian-bagian penyusunnya harus berinteraksi dan berhubungan satu sama lain untuk membentuk satu kesatuan.

Adapun beberapa karakteristik sistem yaitu sebagai berikut (Hakiki et al., 2021):

a. Komponen

Sebuah sistem terdiri dari berbagai komponen yang berinteraksi dan berkolaborasi untuk membentuk suatu kesatuan. Komponen-komponen ini dapat berupa subsistem atau bagian dari sistem itu sendiri.

b. Batas Sistem (*Boundary*)

Batas sistem adalah wilayah yang memisahkan suatu sistem dari sistem lain atau dari lingkungannya. Batas ini menandakan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luar Sistem (*Environment*)

Merupakan segala sesuatu di luar batas sistem yang dapat memengaruhi cara kerja atau operasinya.

d. Penghubung Sistem (*Interface*)

Adalah sarana yang menghubungkan satu subsistem dengan subsistem lainnya.

e. Masukan Sistem (*Input*)

Masukan dapat terdiri dari masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang diperlukan agar sistem dapat berfungsi. *Signal input* adalah energi yang diproses untuk menghasilkan keluaran.

f. Keluaran Sistem (*Output*)

Merupakan output yang dihasilkan dari energi yang diproses oleh sistem

g. Pengolah Sistem (*Process*)

Adalah komponen yang mengolah masukan untuk menghasilkan keluaran yang diharapkan

h. Sasaran Sistem

Jika sistem tidak memiliki sasaran, maka operasinya menjadi tidak berarti

2.2. Informasi

Informasi adalah elemen penting dalam proses pengambilan keputusan karena ia harus bebas dari kesalahan atau ketidakakuratan. Informasi yang berkualitas tinggi memiliki nilai yang mencakup keakuratan, ketepatan waktu, dan relevansi (Maydianto & Ridho, 2021).

Sumber informasi berasal dari data. Data yang telah diproses akan menjadi bentuk yang lebih bermanfaat dan bermakna bagi penerima informasi, sehingga dapat dimanfaatkan oleh siapa saja yang memerlukan data tersebut, baik sebagai pengetahuan maupun sebagai dasar dalam pengambilan keputusan (Putu et al., 2023).

2.3. Sistem Informasi

Sistem informasi adalah kumpulan sumber daya orang, tempat, media, instrumen teknologi, protokol, dan *control* yang bekerja sama untuk mengelola jaringan komunikasi yang signifikan, memproses transaksi reguler untuk mendukung pengguna dan manajemen baik di dalam maupun di luar organisasi, dan berfungsi sebagai dasar untuk pengambilan keputusan yang bijaksana (Sitinjak et al., 2020).

Sederhananya, ada tiga fungsi utama yang ditemukan di semua sistem informasi. Tugas-tugas ini melibatkan memperoleh data sebagai *input*, memprosesnya melalui perhitungan, menggabungkan elemen data, memperbarui akun, dan proses lainnya, dan akhirnya menghasilkan informasi sebagai *output* (Agustiandra & Sabandi, 2019).

2.4. Administrasi

Administrasi adalah proses manajerial yang mencakup berbagai aktivitas seperti pencatatan, perhitungan, dan dokumentasi untuk menyediakan informasi yang diperlukan. Dalam konteks yang lebih sempit, administrasi merujuk pada kegiatan seperti pencatatan, surat-menyurat, dan tugas-tugas ketatausahaan lainnya. Secara umum, administrasi juga bisa dipahami sebagai proses menyeluruh yang melibatkan kerja tim dan penggunaan sumber daya untuk mencapai tujuan melalui perencanaan, pengorganisasian, tindakan, koordinasi, dan pengendalian kegiatan yang sistematis dan terstruktur. Jadi, administrasi adalah aktivitas yang melibatkan pengelolaan dan pelaksanaan pekerjaan secara teratur dan terarah (Wibisono et al., 2020).

2.5. Administrasi Sekolah

Administrasi sekolah adalah pengelolaan dan pemanfaatan semua sumber daya sekolah secara efektif dan efisien untuk memastikan penyelenggaraan pendidikan mencapai tujuan yang diinginkan secara optimal. Administrasi sekolah bisa dipandang sebagai suatu proses atau sebagai tugas (kewajiban). Dalam konteks proses, administrasi sekolah mencakup segala aspek manajerial yang lebih luas dari administrasi itu sendiri, yang kegiatannya meliputi (Satrio et al., 2021):

- a. Perencanaan
- b. Pengorganisasian
- c. Pengarahan atau kepemimpinan
- d. Pengawasan atau pengendalian

Keempat bagian tersebut sepenuhnya saling berhubungan satu sama lain, menjadikannya sistem terintegrasi. Ini menyiratkan bahwa pengorganisasian, pengarahan, dan pengawasan rencana diperlukan. Selain itu, perencanaan, pengarahan, dan pengendalian harus datang sebelum mengatur. Demikian pula, kontrol perlu diarahkan, direncanakan, dan dikoordinasikan.

Dalam konteks pendidikan, administrasi sebagai pekerjaan atau kewajiban juga dikenal sebagai administrasi sekolah, dan terdiri dari enam hal sebagai berikut: (Syarifudin & Setyawan, 2022):

- a. Administrasi peserta didik
- b. Administrasi tenaga pendidik dan tenaga kependidikan, serta struktur organisasinya
- c. Administrasi keuangan
- d. Administrasi sarana prasarana
- e. Administrasi hubungan sekolah dengan masyarakat
- f. Administrasi layanan khusus (bimbingan konseling, unit kesehatan siswa, unit koperasi sekolah, dan kegiatan ekstrakurikuler)

Lembaga pendidikan seharusnya mampu mengelola dan menciptakan keteraturan kinerja dengan cara yang efektif dan efisien pada setiap komponen pendidikan, termasuk kurikulum, kesiswaan, pembiayaan, pendidik dan tenaga kependidikan, sarana dan prasarana, dan komponen lainnya yang mendukung proses dan tujuan pendidikan, harus dapat dikelola dan diatur secara efektif dan efisien oleh lembaga pendidikan (Ginanjari & Purwanto, 2022).

Pembiayaan adalah salah satu komponen penting yang secara langsung membantu manajemen pendidikan berjalan dengan baik. Diharapkan setiap sekolah dapat merencanakan, melaksanakan, mengevaluasi, dan mempertanggungjawabkan pengelolaan dana secara terbuka kepada publik dan pemerintah. Tujuan pendidikan kuantitatif dan kualitatif dapat dicapai dengan adanya biaya pendidikan. Pendidikan hampir tidak dapat berlangsung dengan baik tanpa biaya, jadi pendidikan tidak akan berjalan dengan baik di sekolah (Ginanjari & Purwanto, 2022).

Pengelolaan hubungan sekolah dan masyarakat merupakan proses dialog yang bertujuan untuk meningkatkan kesadaran masyarakat akan kebutuhan dan inisiatif pendidikan sekaligus memicu minat dan kolaborasi masyarakat untuk kemajuan dan pertumbuhan lembaga pendidikan. Karena komunitas sangat penting untuk kelangsungan hidup dan kelangsungan sekolah, sekolah dan komunitas adalah komponen penting dari komunitas daripada lembaga yang terisolasi (Sidauruk et al., 2023).

Administrasi kegiatan ekstrakurikuler yang dilakukan di sekolah merupakan penyusunan data dan informasi secara akurat sebagai pengolahan data. Kegiatan ekstrakurikuler memberikan kesempatan kepada siswa untuk menggunakan pendidikan kepribadian dan menghubungkan pengetahuan yang mereka pelajari dalam kurikulum dengan kebutuhan dan keadaan lingkungan. Kegiatan ekstrakurikuler juga membantu siswa mengembangkan keterampilan, potensi, dan rasa tanggung jawab dengan memberi mereka kesempatan untuk memperluas pengalaman sosial dalam persiapan karir (Athifah et al., 2023).

Administrasi pembiayaan pendidikan di sekolah melibatkan sumber dana dan rencana pengeluaran. Pemerintah, siswa, dan masyarakat memberikan sumbangan dalam bentuk uang atau barang. Rencana pengeluaran juga mencakup gaji guru sebagai biaya utama, serta biaya lain seperti pengadaan fasilitas, pelaksanaan kegiatan belajar-mengajar, dan kegiatan ekstrakurikuler. Orang tua siswa dapat memberikan bantuan dengan memberikan sumbangan ke komite sekolah, membayar biaya pendidikan yang telah ditetapkan, dan memberikan donasi untuk pembangunan fasilitas tertentu (Wulandari et al., 2023).

Untuk menciptakan keselarasan antara orang tua dan sekolah, kerja sama mereka sangat penting. Menurut konsep yang dikemukakan oleh Epstein dan Sheldon, kerjasama antara sekolah, keluarga, dan masyarakat adalah konsep yang memiliki berbagai aspek, dan ketiga komponen ini bertanggung jawab secara bersama-sama untuk meningkatkan dan mengembangkan prestasi akademik siswa (Apriati & Widaty, 2021).

Agar berhasil dan efisien mencapai tujuan sekolah atau pendidikan, masyarakat dan sekolah harus bekerja sama. Namun, sekolah juga memiliki tanggung jawab untuk membantu pencapaian tujuan sosial, terutama yang terkait dengan pendidikan. Dengan demikian, terbukti bahwa masyarakat dan sekolah perlu memiliki hubungan yang ramah dan harmonis (Sidauruk et al., 2023).

Untuk meningkatkan kemampuan manajerial Kepala Sekolah dan Guru, administrasi pendidikan yang teratur dan sistematis sangat penting. Kemampuan ini akan berdampak positif pada peningkatan efisiensi, mutu, dan perluasan kinerja di dunia pendidikan. Informasi yang memadai diperlukan untuk

memastikan bahwa kegiatan tersebut berjalan dengan lancar dan lebih efisien. Sistem informasi pendidikan terdiri dari dua komponen utama, pencatatan data (*recording system*) dan pelaporan (*reporting system*) (Wartini, 2023).

Sistem informasi administrasi adalah sekumpulan komponen atau elemen yang saling terhubung untuk melaksanakan proses pencatatan, pengaturan, dan pengalokasian kegiatan guna mencapai tujuan tertentu, dengan memanfaatkan sarana dan peralatan yang tersedia (Supriadi & Bachtiar, 2019).

2.6. Sumbangan Pembinaan Pendidikan (SPP)

Sumbangan Pembinaan Pendidikan (SPP) adalah iuran bulanan yang harus dibayar oleh siswa sebagai kewajiban mereka di sekolah. Besaran SPP ditentukan berdasarkan kesepakatan antara komite sekolah dan orang tua siswa. Pembayaran SPP bertujuan untuk mendukung peningkatan kualitas pendidikan yang berkaitan dengan sarana dan prasarana kegiatan belajar mengajar (Taufiq et al., 2020).

Institusi pendidikan yang berhasil biasanya didukung oleh biaya pendidikan yang memadai. Kualitas pendidikan umumnya sebanding dengan jumlah biaya yang dikeluarkan, semakin besar dan mahal biaya yang digunakan, semakin baik pula kualitas layanan pendidikan yang diberikan, serta semakin tinggi pula hasil belajar dan mutu lulusan yang dihasilkan. Tanpa dukungan biaya pendidikan yang cukup untuk operasional, akan sulit mencapai mutu pendidikan yang baik (Ginanjari & Purwanto, 2022).

Sumbangan Pembinaan Pendidikan (SPP) terjadi ketika ada hubungan timbal balik antara siswa yang menerima pendidikan dari sekolah atau lembaga kursus dan guru atau tutor yang memberikan pelajaran, sekolah atau lembaga kursus berfungsi sebagai fasilitator dalam proses tersebut (Vetdri et al., 2023).

Administrasi pembayaran Sumbangan Pembinaan Pendidikan (SPP) adalah kegiatan pengelolaan administrasi keuangan yang mencakup pencatatan dan pelaporan data keuangan terkait pembayaran SPP oleh siswa setiap bulan (Desy & Sihombing, 2020). Dalam kebanyakan kasus, sekolah masih menyimpan catatan tulisan tangan dari proses pembayaran uang sekolah, baik dalam bentuk buku besar atau spreadsheet seperti Microsoft Excel. Biasanya, pembayaran uang sekolah dilakukan oleh siswa atau orang tua mereka di loket pembayaran

administrasi sekolah dengan melampirkan kartu uang sekolah sebagai catatan yang sah dan bukti pembayaran (Assalma, 2022).

Pada dasarnya, pemrosesan data keuangan dan pembayaran uang sekolah untuk siswa dapat ditangani secara manual untuk menghasilkan laporan keuangan setiap dua tahun atau tahunan. Namun, untuk mempercepat proses pemasukan data dan penyusunan laporan, sistem yang terintegrasi dengan teknologi komputer sangat penting, mengingat isu-isu seperti perekaman yang masih mengandalkan kertas atau buku, media penyimpanan data yang tidak terkomputerisasi, dan pengambilan data yang membutuhkan pemrosesan satu per satu yang memakan waktu (Waidah & Hursali, 2020).

2.7. Pembayaran

Pembayaran adalah proses yang digunakan untuk mentransfer dana dari satu pihak ke pihak lain. Ini melibatkan transaksi pembayaran untuk berbagai hal seperti layanan, barang, tagihan, dan lainnya. Dengan kata lain, pembayaran berfungsi sebagai sarana untuk memenuhi kewajiban terkait berbagai aktivitas guna mencapai suatu tujuan (Vetdri et al., 2023).

Pembayaran secara *online* dianggap lebih efektif dan praktis dalam mengelola dana, karena uang pembayaran langsung masuk ke rekening sekolah. Hal ini dapat mengurangi potensi kecurangan dalam pengelolaan keuangan sekolah. Jika sistem pembayaran masih dilakukan secara manual, kemungkinan terjadinya kecurangan dan kesalahan manusia jauh lebih tinggi dibandingkan dengan sistem pembayaran *online* yang langsung terhubung ke *bank* dan rekening sekolah (Syarifudin & Setyawan, 2022).

2.8. Payment Gateway

Payment gateway adalah sistem pembayaran *online* yang memproses dan memverifikasi informasi transaksi sesuai dengan kebijakan yang telah ditetapkan oleh penyedia layanan. *Payment gateway* adalah metode untuk memproses transaksi elektronik. Ia menyediakan berbagai alat untuk memfasilitasi pembayaran antara pelanggan, bisnis, dan *bank*. *Payment gateway* memainkan peran penting dalam transaksi antara pelanggan, bisnis, dan lembaga perbankan yang terlibat. *Payment gateway* digunakan untuk mempermudah transaksi

elektronik. Beberapa fitur utama *payment gateway* mencakup enkripsi data pembayaran dan data pribadi, komunikasi antara lembaga keuangan terkait, bisnis, dan pelanggan, serta otorisasi pembayaran (Prasetyo & Sutopo, 2020).

2.9. Midtrans

Midtrans adalah sistem *payment gateway* yang populer di Indonesia, yang menyediakan berbagai fasilitas metode pembayaran seperti pembayaran kartu, transfer bank, *direct debit*, dan *e-wallet*. *Midtrans* juga menawarkan *API* yang dapat dikembangkan menggunakan *framework* seperti *React Native* dan *NodeJS* (Pratama et al., 2020).

Midtrans telah digunakan oleh banyak *start-up* di Indonesia, yang menunjukkan bahwa platform ini memiliki tingkat keamanan yang tinggi dan dapat dipercaya sebagai solusi pembayaran untuk sekolah. Selain itu, digitalisasi pembayaran mengurangi kesalahpahaman antara sekolah dan orang tua tentang transaksi yang telah selesai. (Hariselmi et al., 2022).

2.10. Wireframe

Wireframe adalah representasi visual dari garis dan kotak yang digunakan untuk mengatur tata letak elemen aplikasi. Sebelum proses desain yang sebenarnya dimulai, aplikasi dapat diatur pada halaman aplikasi menggunakan *wireframe*. Perangkat lunak desain seperti *Figma* dapat digunakan untuk membuat *wireframe*. Ada dua kategori *wireframe*, yaitu *wireframe low-fidelity* dan *high-fidelity* (Fadilah & Sweetania, 2023).

2.11. Android

Android adalah sistem operasi untuk perangkat mobile yang berbasis *Linux*, mencakup sistem operasi, *middleware*, dan aplikasi. Karena *Android* adalah *platform* terbuka, pengembang bebas membuat aplikasi mereka sendiri di dalamnya. Ada dua cara *Android* didistribusikan, yang pertama adalah melalui *Google Mobile Services (GMS)* yang memberikan dukungan penuh, yang kedua adalah melalui *Open Handset Distribution (OHD)* yang tidak menerima dukungan langsung dari *Google* (Hidayaturrohman et al., 2020).

Android menyediakan pendekatan komprehensif untuk pengembangan aplikasi, di mana satu aplikasi *Android* yang dibuat dapat berjalan di berbagai

perangkat yang menggunakan sistem operasi *Android*, seperti *smartphone*, *smartwatch*, *tablet*, dan perangkat lainnya (Listiono et al., 2021).



Gambar 2. 1 Logo *Android* (Sumber: www.iconfinder.com)

Android berkembang dengan cepat, saat ini ada 19 versi OS, mulai dari rilis pertama hingga yang terbaru. Berikut macam-macam versi OS *Android* saat ini :

Tabel 2. 1 Perkembangan *Android* (Baskoro & Ariadi, 2023)

Versi	Nama	Rilis	Catatan
1.0	<i>Alpha</i>	23 September 2008	Dirilis untuk umum dan diintegrasikan ke dalam <i>smartphone HTC</i>
1.1	<i>Beta</i>	9 Februari 2009	Mendukung penggunaan <i>marquee</i> atau program <i>HTML</i> yang memungkinkan teks untuk bergerak
1.5	<i>Cupcake</i>	27 April 2009	Versi <i>Android</i> pertama yang diberi nama berdasarkan makanan
1.6	<i>Donut</i>	15 September 2009	<i>Android</i> versi ini diluncurkan dengan menggunakan kernel <i>Linux 2.6.29</i>
2.0 – 2.1	<i>Eclair</i>	3 Desember 2009	<i>Android Eclair</i> meningkatkan kecepatan perangkat lunak dan

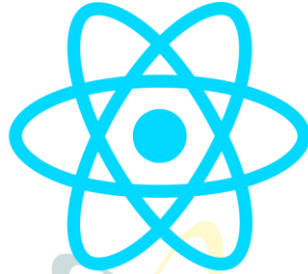
Versi	Nama	Rilis	Catatan
			memperbarui antarmuka pengguna (<i>UI</i>)
2.2	<i>Frozen Yoghurt – Froyo</i>	20 mei 2010	Pembaharuan meningkatkan kinerja hingga dua kali lebih cepat
2.3	<i>Gingerbread</i>	6 Desember 2010	Versi ini menambahkan beberapa <i>fitur</i> , termasuk peningkatan pada <i>gaming</i> dan fungsi <i>salin-tempel</i>
3.0 – 3.1	<i>Honeycomb</i>	22 Februari 2011	Didesain khusus untuk <i>tablet</i>
4.0	<i>Ice Cream Sandwich</i>	19 Oktober 2011	Versi <i>Android</i> ini, pengguna dapat mengambil tangkapan layar ponsel
4.1 – 4.3	<i>Jelly Bean</i>	27 Juni 2012	Memperbaiki kinerja fungsi antarmuka pengguna
4.4	<i>KitKat</i>	3 September 2013	Menawarkan performa sistem yang lebih baik serta layar yang merespons lebih cepat
5.0	<i>Lollipop</i>	24 juni 2014	Antarmuka pengguna dengan desain berbasis <i>grid</i> , serta animasi dan transisi yang <i>responsif</i>
6.0	<i>Marshmallow</i>	5 Oktober 2015	Versi <i>Android Marshmallow</i> memperkenalkan skema manajemen daya yang disebut <i>Doze</i>
7.0 – 7.1	<i>Nougat</i>	22 Agustus 2016	Versi <i>Android</i> ke-13 ini memiliki fitur yang

Versi	Nama	Rilis	Catatan
			memungkinkan tampilan kalibrasi warna
8.0 – 8.1	<i>Oreo</i>	5 Desember 2017	<i>Android Oreo</i> juga memperkenalkan fitur yang memungkinkan pengguna untuk menginstal aplikasi dari sumber yang tidak dikenal
9.0	<i>Pie</i>	6 Agustus 2018	Menawarkan fitur gesture untuk berpindah antara aplikasi dengan mudah
10	<i>Android Q</i>	3 september 2019	<i>Android 10</i> menghadirkan fitur Teks Otomatis yang dapat menambahkan teks pada video atau podcast yang direkam
11	<i>Red Velvet Cake</i>	8 September 2020	<i>OS Android 11</i> memungkinkan pengelompokan aplikasi dalam tampilan status notifikasi
12	<i>Snow Cone</i>	4 Oktober 2021	Versi terbaru dari <i>Android</i> ini memperkenalkan pembaruan besar pada Desain Material, yang dikenal dengan nama " <i>Material You</i> "

2.12. React Native

React Native adalah *framework* mengembangkan aplikasi seluler menggunakan *JavaScript* dan *ReactJs*. *Framework* ini mengikuti desain yang serupa dengan *ReactJs*, memungkinkan pembuatan antarmuka seluler yang kaya dengan komponen deklaratif. Aplikasi yang dibangun dengan *React Native* bukanlah aplikasi *web* seluler, karena ia menggunakan elemen dasar *UI* yang

sama seperti aplikasi *iOS* dan *Android* asli. Sebagai ganti menggunakan *Swift*, *Kotlin*, atau *Java*, *React Native* memungkinkan Anda untuk menyusun elemen tersebut dengan *JavaScript* dan *ReactJs* (Durahim et al., 2023).



Gambar 2. 2 Logo *React Native* (Sumber: www.iconfinder.com)

2.13. *JavaScript*

Bahasa pemrograman *JavaScript* adalah sekelompok skrip yang berjalan di dalam halaman *HTML*. Ini adalah bahasa skrip *online* pertama, sejak hari-hari awal internet. *JavaScript* memberikan kemampuan tambahan pada *HTML* dengan memungkinkan eksekusi perintah di sisi pengguna, yaitu di *browser*, bukan di *server web*. Bahasa ini bergantung pada *browser (navigator)* yang memuat halaman *web* berisi skrip *JavaScript*, yang terintegrasi dalam dokumen *HTML* (Sahi, 2020).



Gambar 2. 3 Logo *JavaScript* (Sumber: www.iconfinder.com)

2.14. *Web*

Website atau *web* adalah kumpulan halaman yang digunakan untuk menampilkan informasi berupa teks, gambar, suara, video, atau kombinasi dari semuanya, baik dalam format statis maupun dinamis, yang saling terhubung melalui internet. Di era informasi saat ini, *website* menjadi kebutuhan bagi siapa saja yang ingin menyebarkan informasi secara *online*, termasuk lembaga

pendidikan yang menggunakannya untuk menyampaikan informasi tentang institusi mereka, melakukan proses belajar mengajar secara daring, dan menyediakan layanan manajemen *online*. Dengan demikian, interaksi dan komunikasi antara lembaga pendidikan dan para pemangku kepentingannya tidak lagi terbatas oleh jarak, waktu, dan tempat (Kurniawan et al., 2024).

2.15. HTML

Hypertext Markup Language (HTML) adalah bahasa markup standar yang digunakan untuk membuat dokumen yang ditampilkan di *browser internet*. HTML merupakan bahasa pemrograman berbasis teks yang digunakan untuk membuat halaman *web*, biasanya dengan ekstensi **.htm* atau **.html*. Sebagai bahasa situs *web (WWW)*, *HTML* berfungsi untuk mengatur dan mendesain dokumen agar dapat ditampilkan di *browser*. Ketika pengguna menjelajah internet, mereka sebenarnya mengakses dokumen yang ditulis dalam format *HTML*. Dengan demikian, *HTML* dapat dianggap sebagai protokol untuk mentransfer data atau dokumen dari *server web* ke *browser* (Amandha et al., 2024).



Gambar 2. 4 Logo HTML (Sumber: www.iconfinder.com)

2.16. PHP

PHP atau singkatan dari *Hypertext Preprocessor* adalah bahasa pemrograman *open source* yang digunakan untuk berkomunikasi dengan sisi *server* dan mengirimkan hasilnya ke klien yang melakukan permintaan. *PHP* diciptakan oleh *Rasmus Lerdorf* pada tahun 1994, dengan singkatan awalnya adalah *Personal Home Page* (Firmansyah & Herman, 2023).

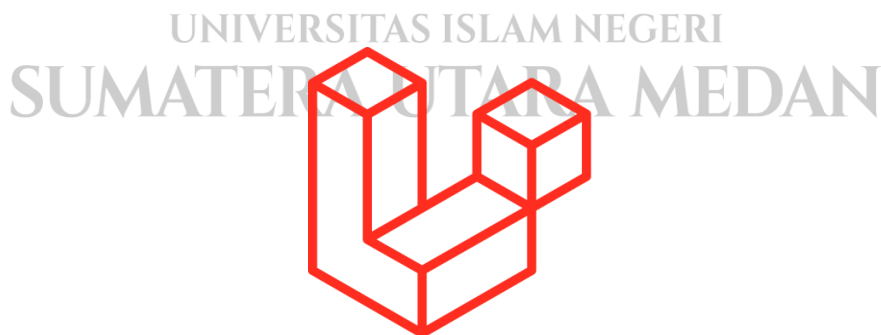
PHP adalah bahasa yang melengkapi *HTML* dan memungkinkan pembuatan aplikasi dinamis dengan kemampuan pemrosesan dan pengolahan data. Semua sintaks *PHP* dijalankan di *server*, sementara hanya hasilnya yang dikirim ke *browser*. *PHP* merupakan bahasa skrip yang ditempatkan dan diproses di *server*, kemudian hasilnya dikirimkan ke klien yang menggunakan *browser*. Dikenal sebagai bahasa *scripting*, *PHP* terintegrasi dengan *tag HTML*, dieksekusi di *server*, dan digunakan untuk membuat halaman *web* dinamis, serupa dengan *Active Server Pages (ASP)* atau *Java Server Pages (JSP)* (Hermiati et al., 2021).



Gambar 2. 5 Logo *PHP* (Sumber: www.iconfinder.com)

2.17. Laravel

Taylor Otwell mengembangkan kerangka kerja web *Laravel* gratis dan *open-source*, yang didasarkan pada *PHP* dan dirancang untuk pengembangan aplikasi *online* yang mengikuti pola *MVC*. Dibandingkan dengan desain *MVC* standar, pola *MVC* di *Laravel* memiliki struktur yang agak berbeda. Untuk mencegah pengontrol menerima permintaan langsung dari pengguna, *Laravel* menawarkan perutean sebagai perantara antara pengguna dan pengontrol (Sari & Wijinarko, 2020).



Gambar 2. 6 Logo *Laravel* (Sumber: www.iconfinder.com)

2.18. CSS

CSS (Cascading Style Sheets) adalah bahasa *styling* yang digunakan untuk merancang tampilan halaman *web*, termasuk pengaturan warna, *font*, garis tepi, latar belakang, dan penyesuaian tampilan dengan ukuran layar. *CSS* bekerja bersama dengan *HTML* dalam pembuatan *website* untuk menciptakan desain yang menarik (Sari et al., 2022).



Gambar 2. 7 Logo *CSS* (Sumber: www.iconfinder.com)

2.19. Database

Database adalah sistem yang dirancang untuk mengatur, menyimpan, dan mengambil data dengan mudah. Terdiri dari kumpulan data yang terstruktur untuk satu atau lebih tujuan, database digital dikelola menggunakan *Database Management System (DBMS)*. *DBMS* menyimpan isi database, memungkinkan pembuatan dan pemeliharaan data, serta pencarian dan akses data lainnya (Sari et al., 2022).

2.20. XAMPP

XAMPP adalah perangkat lunak *web server Apache* yang sudah dilengkapi dengan *server database MySQL* dan mendukung pemrograman *PHP*. *XAMPP* merupakan paket program yang terdiri dari *Apache Web Server*, yang bertugas menerima dan memproses permintaan dari *browser web*. *PHP* sebagai bahasa skrip untuk mengolah data dari komputer klien atau *server*. *MySQL* sebagai *database* untuk menyimpan dan mengelola data dinamis dari aplikasi *web*, serta *PhpMyAdmin* yang berfungsi untuk membuat dan mengelola *database* tanpa menggunakan konsol *MySQL* (Risti, 2023).



Gambar 2. 8 Logo XAMPP (Sumber: www.iconfinder.com)

2.21. MySQL

MySQL adalah salah satu *server database* yang sangat populer. MySQL menggunakan bahasa *SQL* untuk mengelola akses *database*-nya. MySQL memiliki lisensi *FOSS License Exception* serta versi komersial. MySQL tersedia untuk berbagai *platform*, termasuk *Windows* dan *Linux*. Untuk mempermudah administrasi MySQL, Anda dapat menggunakan perangkat lunak seperti *phpMyAdmin* dan *MySQL Workbench* (Ramadhan & Mukhaiyar, 2020).



Gambar 2. 9 Logo MySQL (Sumber: www.iconfinder.com)

2.22. phpMyAdmin

Alat *open source* yang disebut *PhpMyAdmin* dapat digunakan untuk mengelola atau memprogram *database MySQL* tanpa biaya. Beragam aktivitas MySQL, termasuk mengelola *database*, tabel, bidang, hubungan, indeks, pengguna, izin, dan banyak lagi, didukung oleh *PhpMyAdmin*, program berbasis *PHP*. Akibatnya, *phpMyAdmin* dan *MySQL* tidak sama. *PhpMyAdmin* adalah alat yang memudahkan pengelolaan *database MySQL*, sedangkan *MySQL* adalah sistem manajemen basis data yang menyimpan data dalam dan dari dirinya sendiri (Ramadhan & Mukhaiyar, 2020).



Gambar 2. 10 Logo *phpMyAdmin* (Sumber: www.iconfinder.com)

2.23. API

API (Application Programming Interface) atau antarmuka pemrograman aplikasi, adalah jenis antarmuka yang memungkinkan perangkat lunak mengakses layanan atau aplikasi. *Developer* tidak perlu lagi membuat aplikasi baru karena *API* memudahkan penggunaan fungsionalitas yang sudah ada sebelumnya dari aplikasi lain. *API* beroperasi di *web* dengan melakukan panggilan ke fungsi melalui *Hypertext Transfer Protocol (HTTP)* dan mendapatkan balasan kembali dalam bentuk *JavaScript Object Notation (JSON)* atau *Extensible Markup Language (XML)*. Tujuan utama memanfaatkan *API* adalah untuk memfasilitasi pertukaran data antara berbagai aplikasi (Hasanuddin et al., 2022).

2.24. Figma

Figma adalah salah satu alat desain yang sering digunakan untuk membuat tampilan aplikasi *mobile*, *desktop*, *website*, dan lainnya. *Figma* dapat digunakan di berbagai sistem operasi, seperti *Windows*, *Linux*, atau *macOS*, selama terhubung ke *internet*. Selain menawarkan semua fungsionalitas *Adobe XD*, *Figma* memiliki kelebihan tambahan, yaitu memungkinkan lebih dari satu orang untuk bekerja pada proyek yang sama secara bersamaan, meskipun berada di lokasi berbeda. Kemampuan ini mendukung kerja kolaboratif, menjadikan *Figma* pilihan utama bagi banyak desainer *UI/UX* untuk membuat prototipe *website* atau aplikasi dengan cepat dan efisien (Muhyidin et al., 2020).



Gambar 2. 11 Logo *Figma* (Sumber: www.iconfinder.com)

2.25. Unified Modelling Language

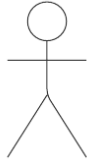
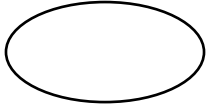


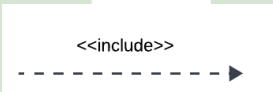

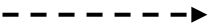

Unified Modelling Language (UML) adalah bahasa yang digunakan untuk mendefinisikan, memvisualisasikan, membangun, dan mendokumentasikan artefak-artefak (komponen informasi yang dihasilkan selama proses pengembangan perangkat lunak, seperti model, deskripsi, atau perangkat lunak) dari sistem perangkat lunak. *UML* diterapkan dalam pemodelan bisnis dan sistem non-perangkat lunak lainnya. Sebagai bahasa pemodelan yang mengadopsi konsep pemrograman berorientasi objek, *UML* dikembangkan oleh *Grady Booch*, *James Rumbaugh*, dan *Ivar Jacobson* di bawah bendera *Rational Software Corps*. *UML* menawarkan notasi yang berguna untuk memodelkan sistem dari berbagai perspektif (Prasetya et al., 2022).



2.25.1. Use Case

Use case diagram adalah diagram yang menggambarkan hubungan antara aktor dan *use case*. *Use case* menjelaskan tindakan atau aksi yang dilakukan oleh aktor. Diagram ini mencerminkan fungsionalitas yang diharapkan dari sistem, dengan fokus pada "apa" yang dilakukan oleh sistem, bukan "bagaimana" cara melakukannya. Setiap *use case* menunjukkan interaksi antara aktor dan sistem (Arianti et al., 2022).

Tabel 2. 2 Simbol-simbol *Use Case Diagram* (Nasution & Harahap, 2022)

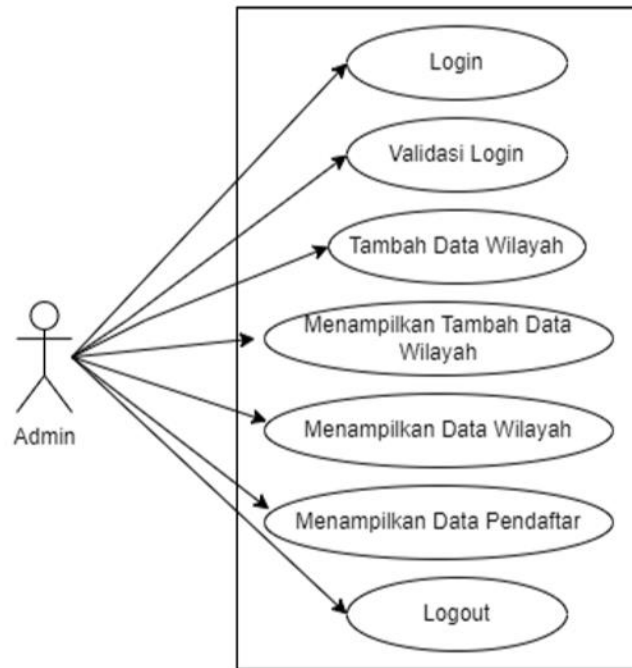
No	Gambar	Nama	Keterangan
----	--------	------	------------

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menentukan kumpulan peran yang dimainkan oleh pengguna saat berinteraksi dengan <i>use case</i>
2.		<i>Use Case</i>	Tindakan yang dapat dilakukan aktor pada sistem
3.		<i>Association</i>	Hubungan antar <i>actor</i> dengan <i>use case</i>
4.		<i>System</i>	Menentukan paket yang menggambarkan sistem dengan cakupan yang terbatas
5.		<i>Include</i>	Mengizinkan satu <i>use case</i> untuk memanfaatkan fungsi yang disediakan oleh <i>use case</i> lainnya.
6.		<i>Extend</i>	Memberikan kesempatan bagi suatu <i>use case</i> untuk memperluas fungsionalitas yang ditawarkan oleh <i>use case</i> lainnya.
7.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada satu elemen mandiri akan mempengaruhi elemen yang bergantung padanya yang tidak mandiri
8.		<i>Generalization</i>	Hubungan antara <i>dua use case</i> yang menggambarkan generalisasi dan spesialisasi

No	Gambar	Nama	Keterangan
			(umum-khusus), di mana satu <i>use case</i> memiliki fungsi yang lebih umum dibandingkan dengan yang lainnya.
9.		<i>Collaboration</i>	Kolaborasi antara berbagai aturan dan elemen yang bersama-sama menciptakan suatu perilaku yang lebih kompleks dan efektif dibandingkan dengan kontribusi masing-masing elemen secara terpisah (sinergi).
10.		<i>Note</i>	Elemen fisik yang ada selama aplikasi beroperasi dan mencerminkan sumber daya komputasi.

Contoh *Use Case Diagram*:

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN






Gambar 2. 12 Use Case Diagram (Haritsyah & Harahap, 2024)

2.25.2. Activity Diagram

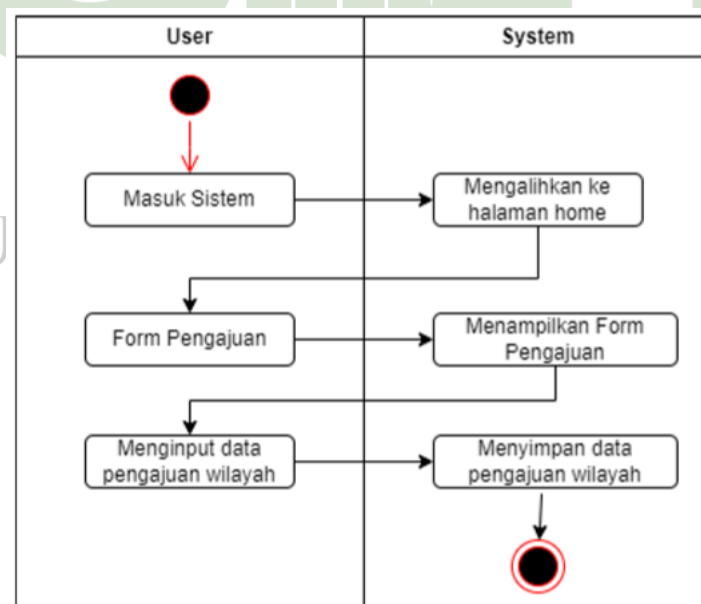
Activity diagram adalah sebuah model yang menggambarkan proses kerja sistem atau objek, dengan menampilkan alur proses dari awal hingga akhir. Diagram ini menunjukkan tahapan-tahapan yang dilakukan dalam *use case* tertentu secara terstruktur, dan setiap aktivitas diilustrasikan dengan notasi yang sesuai dengan fungsinya (Aliman, 2021).

Tabel 2. 3 Simbol-simbol *Activity Diagram* (Bantun et al., 2022)

No	Gambar	Nama	Keterangan
1.		<i>Initial Node</i>	Bagaimana objek dibuat atau dimulai
2.		<i>Activity Final Node</i>	Cara objek dibentuk dan dihapuskan
3.		<i>Activity</i>	Menunjukkan bagaimana setiap kelas antarmuka saling berinteraksi satu sama lain

4.		<i>Action</i>	<i>State</i> dalam sistem yang menggambarkan pelaksanaan suatu aksi
5.		<i>Fork Node</i>	Satu aliran yang pada suatu tahap berpecah menjadi beberapa aliran.
6.		<i>Decision</i>	Digunakan untuk menggambarkan keputusan atau tindakan yang perlu diambil dalam situasi tertentu.

Contoh *Activity Diagram*:

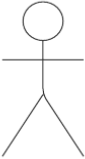
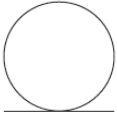
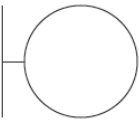
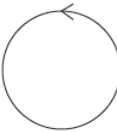

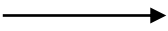


Gambar 2. 13 *Activity Diagram* (Haritsyah & Harahap, 2024)

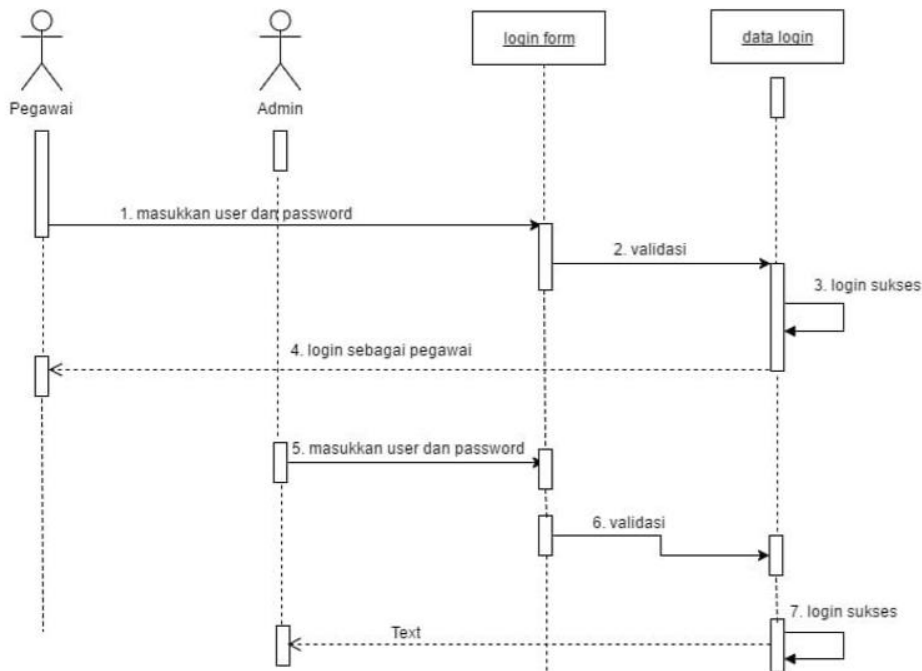
2.25.3. Sequence Diagram

Sequence Diagram adalah diagram yang menggambarkan proses yang dilakukan pengguna dalam sistem informasi sesuai dengan urutan waktu dari setiap tahapannya. *Sequence diagram* menggambarkan interaksi antara objek yang diatur berdasarkan urutan waktu. Dengan kata lain, *sequence diagram* adalah representasi langkah-langkah yang harus diambil untuk menciptakan sebuah sistem yang sesuai dengan diagram *use case* (Lubis et al., 2023).

Tabel 2. 4 Simbol-simbol *Sequence Diagram* (Saputra & Fahrizal, 2021)

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menggambarkan sebuah objek berinteraksi dengan sistem
2.		<i>Entity Class</i>	Menggambarkan hubungan yang akan digambarkan oleh sistem
3.		<i>Boundary Class</i>	Menggambarkan <i>scope</i> atau biasanya berupa <i>form</i>
4.		<i>Control Class</i>	Menggambarkan hubungan antara <i>boundary</i> dengan <i>table</i>
5.		<i>Life Line</i>	Menggambarkan tempat dimulai dan berakhirnya <i>message</i>
6.		<i>Message</i>	Menggambarkan terjadinya pengiriman pesan

Contoh *Sequence Diagram*:



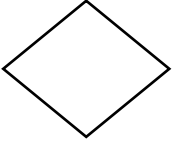


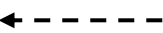


Gambar 2. 14 *Sequence Diagram* (Nasution et al., 2024)

2.25.4. *Class Diagram*

Class diagram adalah spesifikasi yang penting untuk pembuatan dan desain suatu objek karena, setelah instansiasi, ia menghasilkan objek. Karakteristik sistem digambarkan dalam gambar ini, bersama dengan fungsi atau teknik untuk menyesuaikannya. *Class diagram* menunjukkan struktur dan deskripsi kelas, paket, dan objek, serta hubungan di antara mereka seperti *contaiment*, pewarisan, asosiasi, dan lainnya (Arianti et al., 2022).

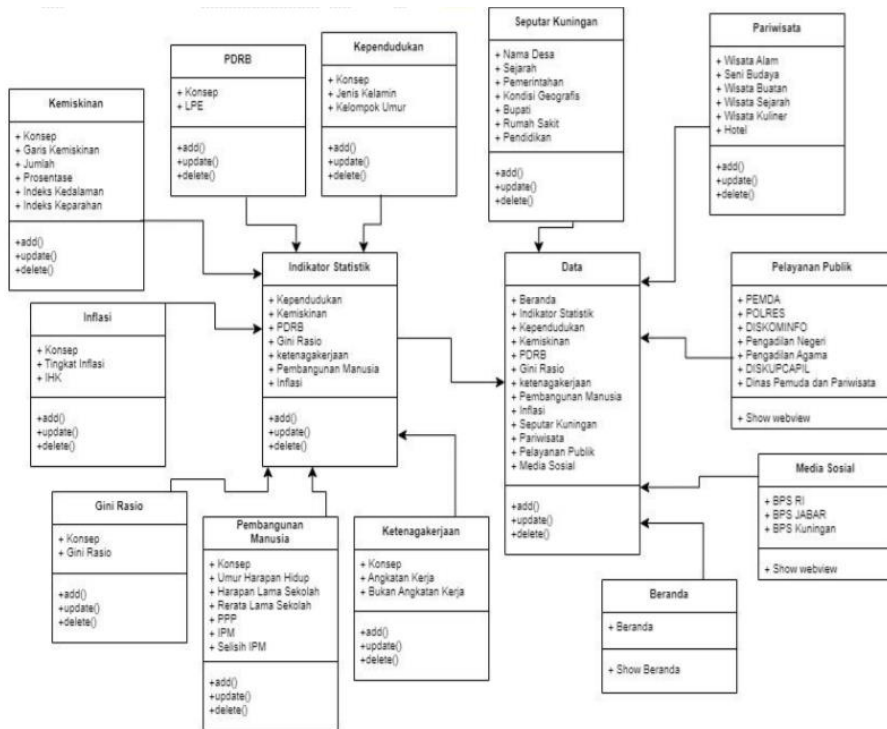
Tabel 2. 5 Simbol-simbol *Class Diagram* (Bantun et al., 2022)

No	Gambar	Nama	Keterangan
1.	—	<i>Association</i>	Keterkaitan antara satu objek dengan objek lainnya

No	Gambar	Nama	Keterangan
2.		<i>Nary Association</i>	Usaha untuk mengisolasi asosiasi antara dua objek atau lebih.
3.		<i>Class</i>	Kumpulan objek yang memiliki atribut dan operasi yang serupa.
4.		<i>Collaboration</i>	Deskripsi urutan tindakan yang ditampilkan oleh sistem dalam unit terukur untuk seorang aktor.
5.		<i>Realization</i>	Aksi yang dilaksanakan oleh objek.
6.		<i>Defedency</i>	Modifikasi yang terjadi pada suatu elemen yang berdiri sendiri (<i>independent</i>) yang memengaruhi elemen yang tidak berdiri sendiri.
7.		<i>Generalization</i>	Hubungan antara objek turunan (<i>descendant</i>) yang mewarisi perilaku dan struktur data dari objek induk (<i>ancestor</i>).

Contoh *Class Diagram*:

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN



Gambar 2. 15 *Class Diagram* (Bacsafra et al., 2022)

2.26. Penelitian Terkait

Berikut ada beberapa penelitian sebelumnya yang relevan, dan dapat dijadikan sebagai referensi serta perbandingan dengan penelitian ini. Berikut adalah beberapa penelitian terdahulu:

Tabel 2. 6 Penelitian Terdahulu

No	Judul	Tahun	Hasil Penelitian	Perbedaan
1.	Sistem Informasi Administrasi Pembayaran SPP Berbasis <i>Website</i>	2024	Hasil penelitian ini adalah sistem pembayaran SPP dengan transaksi pembayaran yang dilakukan secara <i>online</i> melalui <i>platform web</i> , mengelola data dan membuat laporan dengan cepat, serta menyediakan notifikasi otomatis untuk mengingatkan orang tua atau wali murid mengenai tanggal jatuh tempo pembayaran.	Perbedaan utama antara sistem yang akan dikembangkan dan sistem sebelumnya adalah bahwa sistem baru ini berbasis <i>Android</i>
2.	Implementasi Sistem Informasi Pembayaran SPP Berbasis <i>Android</i>	2022	Hasil dari penelitian ini yaitu sistem administrasi berbasis <i>Android</i> yang dapat melakukan pembayaran SPP	Perbedaan dengan sistem sebelumnya adalah terletak pada peningkatan antarmuka pengguna (UI) aplikasi yang

No	Judul	Tahun	Hasil Penelitian	Perbedaan
	Dengan <i>Payment Gateway Midtrans</i>		secara <i>online</i> , pembuatan laporan dan pengelolaan data dengan cepat, melihat <i>history</i> pembayaran, serta mengecek tagihan pembayaran SPP	sebelumnya masih kurang menarik, serta penambahan fitur fitur tambahan seperti donasi, berita, dan kalender pendidikan
3.	Sistem pembayaran SPP di SMK Putra Mandala 1 Kabupaten Sukabumi Berbasis Web	2023	Penelitian ini menghasilkan sistem administrasi pembayaran SPP yang mampu mengelola data terkait pembayaran SPP, termasuk pengaturan tahun ajaran, manajemen data kelas, data siswa, jenis pembayaran, transaksi pembayaran, dan tagihan.	Perbedaan antara sistem yang akan dibuat dengan sistem sebelumnya adalah kemampuannya untuk mencetak laporan dan rekap transaksi pembayaran.
4.	Sistem Informasi Pembayaran Administrasi Sekolah Berbasis Web dan Mobile	2023	Penelitian ini menghasilkan sistem pembayaran administrasi sekolah berbasis <i>web</i> dan <i>mobile</i> yang mampu mengelola pendataan	Perbedaannya terletak pada penggunaan <i>framework Laravel</i> untuk pembuatan sistem <i>web</i> -nya.

No	Judul	Tahun	Hasil Penelitian	Perbedaan
	pada MTS NU 17 Kyai Jogoreso		pembayaran administrasi sekolah dan memudahkan pembuatan laporannya, serta dapat melihat tagihan dan riwayat pembayaran	
5.	Aplikasi Pembayaran SPP Berbasis Android Di SMA Permata Pasarkemis	2020	Penelitian ini menghasilkan sistem yang dapat melihat riwayat pembayaran SPP melalui <i>fitur fees</i> , dengan penyampaian info yang cepat setelah data di <i>update</i> oleh tata usaha. Kemudian dapat menyimpan dan mengelola data pembayaran SPP secara efisien yang dapat memudahkan petugas tata usaha	Perbedaan pada penelitian ini adalah bahwa dalam penelitian sebelumnya sistem pembayarannya masih menggunakan cara tunai (<i>cash</i>), hanya dapat melihat riwayat pembayarannya saja