

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Rancang Bangun**

Satzinger, Jackson, dan Burd (2012) Rancang bangun atau *System Design* adalah proses menentukan secara rinci bagaimana komponen-komponen sistem informasi diimplementasikan. Ini mencakup desain antarmuka, basis data, dan arsitektur perangkat lunak.

Rancang bangun sebuah sistem merupakan sebuah proses metodologis yang menggabungkan elemen analisis kebutuhan dengan implementasi teknis untuk menghasilkan solusi perangkat lunak yang spesifik. Menurut Hesinto (2023), rancangan sistem yang baik harus mampu meningkatkan kinerja pengolahan data administrasi dan memastikan keamanan data melalui integrasi *database* yang terstruktur. Dalam konteks perbendaharaan sekolah, rancang bangun sistem bertujuan untuk memigrasikan prosedur manual yang lambat menjadi sistem informasi yang terkomputerisasi guna meminimalisir kesalahan pelaporan (Hesinto et al, 2023).

Proses rancang bangun ini melibatkan tahapan perancangan antarmuka (*interface*) dan arsitektur basis data agar aplikasi yang dihasilkan bersifat intuitif bagi pengguna. Sebagaimana dijelaskan oleh Maulana et al., (2022), perancangan sistem informasi pada instansi pendidikan sangat krusial untuk memicu kinerja administrasi yang lebih baik, terutama dalam menangani transaksi rutin seperti pembayaran SPP dan pendataan peserta didik secara akurat (Karunia et al., 2022)

#### **2.2 Sistem Informasi**

Menurut Fat dalam buku Konsep Sistem Informasi yang ditulis oleh Jeperson Hutahaean menyebutkan bahwa sistem adalah suatu himpunan dari suatu "benda" nyata atau abstrak (*a set of thing*) yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan, saling mendukung, yang secara keseluruhan bersatu dalam kesatuan (*unity*) untuk mencapai tujuan tertentu secara efisien dan efektif. Menurut Jerry Futz Gerald,

(1981 : 45 ) sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul, bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

Sedangkan informasi sendiri memiliki definisi sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber informasi adalah data nyata (fakta) yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Gordon B. Davis mengemukakan bahwa informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi penerima dan mempunyai nilai nyata atau yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau yang akan datang.

Dari definisi-definisi diatas, dapat disimpulkan bahwa sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan yang terdiri dari beberapa komponen, yaitu *input*, *model*, *output*, *tecnologi*, *data base*, dan *control*.

### 2.3 Penggajian Guru (*Payroll*)

Sistem penggajian pada institusi pendidikan bukan sekadar proses administratif rutin, melainkan instrumen krusial dalam menjaga stabilitas kinerja tenaga pendidik (Ibrahim et al., 2025). Secara fundamental, penggajian merupakan totalitas kompensasi yang diberikan sebagai imbal balik atas jasa yang diserahkan. Pada konteks sekolah swasta, struktur gaji sering kali bersifat dinamis karena melibatkan variabel kehadiran fisik yang berimplikasi pada insentif tambahan atau honorarium di luar gaji pokok.

Secara umum, alur penggajian diawali dengan pengumpulan data yang menjadi dasar perhitungan gaji, seperti data kehadiran, beban kerja, maupun komponen tunjangan dan potongan yang berlaku. Data tersebut kemudian melalui proses verifikasi untuk memastikan kesesuaiannya dengan kondisi sebenarnya sebelum dilakukan perhitungan besaran gaji yang akan diterima oleh setiap tenaga pendidik. Setelah proses perhitungan selesai, hasil penggajian direkap sebagai dasar

pembuatan laporan dan selanjutnya dilakukan proses pembayaran kepada masing-masing guru sesuai dengan ketentuan yang berlaku. Oleh karena itu, ketepatan pencatatan data pada setiap tahapan menjadi faktor yang sangat menentukan agar proses penggajian dapat berlangsung secara akurat, transparan, dan meminimalkan terjadinya kesalahan perhitungan (Dyah et al., 2026; Nugraha et al., 2025).

Salah satu organisasi dalam bidang pendidikan yang menjalankan proses penggajian adalah SMA Asy-Syuja'iyah. Dalam pelaksanaan proses penggajian di setiap instansi tidak selalu sama, hal tersebut tergantung dengan kondisi dan kesepakatan karyawan dengan instansi tersebut. Sistem penggajian yang tergolong menggunakan sistem manual akan membutuhkan waktu relatif lebih lama dan ketelitian yang sangat mendalam.

## **2.4 Presensi dan Data Kehadiran**

Presensi merupakan proses pencatatan kehadiran seseorang sebagai bentuk dokumentasi terhadap pelaksanaan tugas atau aktivitas pada waktu dan tempat tertentu. Proses ini bertujuan untuk memperoleh informasi mengenai kehadiran, ketepatan waktu, serta kedisiplinan individu dalam menjalankan tanggung jawabnya. Hasil dari proses presensi berupa data kehadiran yang selanjutnya dapat dimanfaatkan sebagai sumber informasi dalam berbagai kegiatan administrasi, seperti penyusunan laporan, evaluasi kinerja, maupun pengambilan keputusan yang berkaitan dengan pengelolaan sumber daya manusia (Nurfitri et al., 2024)

Dalam lingkungan pendidikan, data kehadiran guru memiliki peranan yang penting karena memberikan informasi mengenai pelaksanaan kegiatan mengajar sesuai dengan jadwal yang telah ditetapkan. Data tersebut tidak hanya digunakan untuk mengetahui tingkat kehadiran guru, tetapi juga menjadi salah satu dasar dalam pelaksanaan administrasi sekolah, pemantauan kedisiplinan, serta evaluasi pelaksanaan proses pembelajaran. Oleh karena itu, pencatatan kehadiran perlu dilakukan secara sistematis agar data yang dihasilkan memiliki tingkat akurasi yang baik dan dapat dipertanggungjawabkan (Gunawan et al., 2023)

Perkembangan teknologi informasi telah mendorong perubahan mekanisme presensi dari pencatatan manual menuju sistem berbasis digital. Pemanfaatan

teknologi memungkinkan proses pencatatan kehadiran dilakukan secara lebih cepat, terstruktur, dan terdokumentasi dengan baik sehingga dapat meminimalkan kesalahan pencatatan serta mempermudah proses pengolahan data. Selain meningkatkan efisiensi administrasi, data kehadiran yang tersimpan secara digital juga lebih mudah diintegrasikan dengan sistem informasi lain sesuai kebutuhan instansi (Nurfitri et al., 2024)

Dalam penelitian ini, data kehadiran tidak hanya berfungsi sebagai dokumentasi kehadiran guru, tetapi juga dimanfaatkan sebagai dasar perhitungan Jam Berdiri. Setiap data kehadiran yang telah melalui proses validasi selanjutnya diolah untuk menghasilkan akumulasi Jam Berdiri yang menjadi salah satu komponen pendukung dalam proses penggajian guru di SMA Asy-Syu'ja'iyah. Dengan demikian, kualitas data kehadiran menjadi aspek yang penting karena berpengaruh terhadap ketepatan informasi yang dihasilkan oleh sistem.

## **2.5 Konsep Jam Berdiri**

"Jam Berdiri" adalah istilah yang digunakan di SMA Asy-Syu'ja'iyah untuk mengacu pada total jam mengajar yang benar-benar dilakukan oleh guru dalam setiap kegiatan pembelajaran. Di lingkungan sekolah tersebut, Jam Berdiri digunakan sebagai salah satu acuan dalam menentukan insentif mengajar yang diberikan selain dari gaji pokok. Sebab itu, besarnya insentif yang diterima oleh seorang guru tergantung pada banyaknya jam mengajar yang sudah berjalan dan tercatat selama masa tertentu.

Secara konseptual, "Jam Berdiri" dapat diartikan sebagai cara untuk mencatatkan jam mengajar yang sudah dilakukan oleh guru. Meskipun istilah ini adalah kebijakan internal sekolah, konsep di baliknya berkaitan dengan tugas pekerjaan guru dan sistem penggajian yang bergantung pada prestasi kerja. Dalam sistem tersebut, insentif diberikan berdasarkan kerja yang sudah dilakukan, sehingga mencatat aktivitas mengajar menjadi hal penting untuk mendukung keakuratan perhitungan kompensasi.

Berdasarkan konsep itu, data "Jam Berdiri" harus dicatat dengan tepat agar informasi yang didapat bisa dipakai sebagai acuan dalam proses pembuatan gaji.

Ketepatan dalam mencatat tidak hanya memengaruhi perhitungan insentif mengajar, tapi juga membantu membuat proses administrasi lebih objektif, transparan, dan bisa dipertanggungjawabkan. Berdasarkan hal tersebut, penelitian ini menggunakan sistem absensi berbasis *QR Code* yang digabungkan dengan teknologi *geofencing* sebagai cara untuk membantu proses pencatatan jam berdiri secara lebih teratur dan sesuai dengan kondisi kehadiran guru di dalam lingkungan sekolah.

## 2.6 Teknologi Quick Response

### 2.6.1 Quick Response Code (QR Code)

*QR Code* merupakan evolusi dari kode batang (*barcode*) tradisional yang dikembangkan dalam bentuk matriks dua dimensi. Berbeda dengan barcode satu dimensi yang hanya terdiri dari pola garis hitam-putih, *QR Code* memiliki struktur piksel yang mampu menyimpan data hingga ribuan karakter alfanumerik. Menurut Ray G. Ramadhanti et al., (2025), keunggulan utama teknologi ini terletak pada kemampuannya untuk menyimpan informasi secara kompleks namun tetap dapat dikenali oleh sistem komputer dengan sangat cepat. Karakteristik ini menjadikannya instrumen yang efisien untuk pengarsipan data identitas digital secara praktis (Rahmatuloh et al., 2023; Ray Ghina Ramadhanti, 2025).



Gambar 2. 1 Contoh Gambar QR-Code  
Sumber : Wikipedia

Secara fungsional, *QR Code* menawarkan fleksibilitas tinggi karena dapat diimplementasikan pada berbagai media, mulai dari cetakan fisik hingga layar digital. Kelebihan kapasitas penyimpanan dan kecepatan akses informasi menggunakan perangkat seluler (*mobile*) menjadikannya solusi ideal untuk sistem

presensi modern (Gunawan et al., 2023). Dalam penelitian ini, penggunaan *QR Code* difungsikan sebagai media validasi kehadiran yang meminimalisir interaksi fisik dan mempercepat proses input data ke dalam database sekolah, sehingga alur administrasi menjadi lebih responsif dibandingkan metode manual.

### 2.6.2 *MediaDevices API (Akses Kamera pada Browser)*

*MediaDevices API* merupakan *Application Programming Interface (API)* yang disediakan oleh *HTML5* untuk memberikan akses kepada aplikasi web terhadap perangkat keras, seperti kamera dan mikrofon, melalui browser setelah pengguna memberikan izin akses. Salah satu metode yang tersedia pada *API* ini adalah *getUserMedia()*, yang digunakan untuk memperoleh aliran video (*video stream*) atau audio secara langsung dari perangkat. Demi menjaga keamanan dan privasi pengguna, penggunaan *MediaDevices API* hanya diizinkan pada halaman yang menggunakan protokol *HTTPS* atau lingkungan *localhost* selama proses pengembangan (Nurfaizal, 2026)

*MediaDevices API* banyak dimanfaatkan dalam pengembangan aplikasi berbasis web yang memerlukan akses kamera, salah satunya pada proses pembacaan *QR Code*. Untuk melakukan proses tersebut, pengembang umumnya memanfaatkan pustaka *JavaScript*, seperti *html5-qrcode*, yang menggunakan fungsi *getUserMedia()* untuk mengambil tampilan video dari kamera, kemudian mendeteksi dan menerjemahkan pola *QR Code* menjadi data digital yang dapat diproses oleh aplikasi. Pendekatan ini memungkinkan proses pemindaian dilakukan secara langsung melalui *browser* tanpa memerlukan aplikasi tambahan pada perangkat pengguna.

Berdasarkan karakteristik tersebut, *MediaDevices API* menjadi salah satu teknologi yang banyak digunakan dalam pengembangan aplikasi web modern yang membutuhkan akses kamera secara langsung melalui *browser*. Pemanfaatannya dinilai mampu meningkatkan fleksibilitas implementasi karena dapat dijalankan pada berbagai perangkat yang telah mendukung standar *HTML5*, tanpa bergantung pada platform sistem operasi tertentu (Nurfaizal, 2026).

## 2.7 Teknologi Validasi Lokasi

### 2.7.1 *Global Positioning System (GPS)*

*Global Positioning System (GPS)* merupakan sistem navigasi berbasis satelit yang digunakan untuk menentukan posisi suatu objek di permukaan bumi berdasarkan koordinat geografis berupa garis lintang (*latitude*) dan garis bujur (*longitude*). Posisi tersebut diperoleh dari hasil penerimaan sinyal beberapa satelit yang kemudian diolah oleh perangkat penerima menjadi informasi lokasi. Saat ini, GPS telah dimanfaatkan dalam berbagai layanan berbasis lokasi karena mampu menyediakan data koordinat yang dapat digunakan sebagai dasar pengambilan keputusan pada suatu sistem (Saputra et al., 2026).



Gambar 2. 2 Animasi GPS  
sumber : <https://www.flaticon.com/search?word=gps>

### 2.7.2 *Geofencing*

Sebagai salah satu penerapan teknologi *GPS*, *geofencing* merupakan teknologi yang membentuk batas virtual pada suatu wilayah berdasarkan titik koordinat tertentu. Batas virtual tersebut digunakan untuk menentukan apakah suatu objek berada di dalam atau di luar area yang telah ditetapkan. Berdasarkan bentuk wilayahnya, *geofencing* dapat diterapkan menggunakan batas berbentuk lingkaran (*circular geofence*) maupun poligon (*polygon geofence*). Pemilihan bentuk *geofencing* disesuaikan dengan kebutuhan sistem serta karakteristik area yang akan divalidasi.



Gambar 2. 3 Ilustrasi Konsep *Geofencing*  
Sumber : Dikelola oleh peneliti (2026)

Hubungan antara *GPS* dan *geofencing* terletak pada proses validasi lokasi. *GPS* berfungsi menyediakan data koordinat lokasi pengguna, sedangkan *geofencing* memanfaatkan data tersebut untuk menentukan apakah posisi pengguna masih berada pada area yang diizinkan. Dengan kata lain, *geofencing* tidak dapat melakukan proses validasi lokasi tanpa adanya informasi koordinat yang diperoleh dari *GPS*. Oleh karena itu, kedua teknologi tersebut saling mendukung dalam penerapan sistem berbasis lokasi (Akmal Oktaviawan, 2025).

### 2.7.3 Algoritma *Haversine*

Algoritma *Haversine* merupakan metode yang digunakan untuk menghitung jarak antara dua titik di permukaan bumi berdasarkan koordinat garis lintang (*latitude*) dan garis bujur (*longitude*). Perhitungan dilakukan dengan mempertimbangkan bentuk bumi yang diasumsikan menyerupai bola, sehingga hasil yang diperoleh lebih akurat dibandingkan perhitungan jarak menggunakan bidang datar. Oleh karena itu, algoritma ini banyak dimanfaatkan pada sistem berbasis lokasi (*Location Based Service/LBS*), navigasi, maupun aplikasi yang memerlukan informasi jarak antar koordinat geografis (Akmal Oktaviawan, 2025).

Secara umum, Algoritma *Haversine* bekerja dengan menghitung selisih nilai *latitude* dan *longitude* dari dua titik koordinat, kemudian mengolahnya menggunakan fungsi trigonometri untuk memperoleh jarak permukaan bumi. Hasil perhitungan dinyatakan dalam satuan meter atau kilometer sesuai kebutuhan sistem. Pendekatan ini dinilai efisien karena mampu menghasilkan estimasi jarak yang baik

tanpa memerlukan proses komputasi yang kompleks, sehingga sesuai digunakan pada aplikasi yang dijalankan melalui perangkat bergerak (Dian Saputra et al., 2026).

#### **2.7.4 HTML5 Geolocation API**

*HTML5 Geolocation API* merupakan *Application Programming Interface* (*API*) yang disediakan oleh standar *HTML5* untuk memperoleh informasi lokasi geografis pengguna melalui *browser*. Informasi lokasi tersebut dapat diperoleh dari beberapa sumber, seperti *GPS* pada perangkat bergerak, jaringan *Wi-Fi*, maupun jaringan seluler, bergantung pada perangkat yang digunakan. Sebelum data lokasi diberikan kepada aplikasi, *browser* akan terlebih dahulu meminta persetujuan dari pengguna sebagai bentuk perlindungan terhadap privasi. Dengan mekanisme tersebut, pengguna tetap memiliki kendali terhadap informasi lokasi yang akan dibagikan kepada aplikasi (Sukron et al., 2026).

Secara umum, *HTML5 Geolocation API* bekerja dengan menerima permintaan akses lokasi dari aplikasi web yang dijalankan pada *browser*. Setelah pengguna memberikan izin, *browser* akan mengambil koordinat lokasi perangkat berupa *latitude* dan *longitude*, kemudian mengirimkan informasi tersebut kepada aplikasi melalui fungsi yang disediakan oleh *API*. Mekanisme ini memungkinkan aplikasi web memperoleh data lokasi secara langsung tanpa memerlukan instalasi aplikasi tambahan pada perangkat pengguna.

Selain mempermudah proses pengambilan lokasi, penggunaan *HTML5 Geolocation API* juga mendukung penerapan validasi lokasi secara otomatis pada sistem presensi berbasis web. Meskipun demikian, keberhasilan memperoleh koordinat tetap dipengaruhi oleh persetujuan pengguna terhadap permintaan akses lokasi, kondisi layanan lokasi pada perangkat, serta tingkat akurasi pembacaan sinyal *GPS*. Oleh karena itu, sistem perlu memastikan bahwa layanan lokasi telah aktif sebelum proses validasi lokasi dilakukan (Sukron et al., 2026).

## 2.8 Konsep *E-mail* dan *Google Mail (G-mail)*

### 2.8.1 Pengertian dan Sejarah *E-Mail*

*E-mail (Electronic Mail)* atau surat elektronik merupakan salah satu layanan komunikasi digital yang digunakan untuk mengirim dan menerima pesan melalui jaringan internet. Berbeda dengan surat konvensional yang memerlukan waktu pengiriman relatif lama, *e-mail* memungkinkan informasi dikirim dan diterima dalam hitungan detik selama pengirim dan penerima terhubung dengan jaringan internet (Master Dukom, 2023)

Menurut Widya Aryo Damar dalam bukunya yang berjudul "Cerdas ber-e-mail" (2023) menjelaskan bahwa sejarah *e-mail* dimulai pada tahun 1971, seorang insinyur bernama Ray Tomlinson menciptakan program yang memungkinkan seseorang dapat saling mengirim pesan satu sama lain dengan mesin yang sama. Pengguna dapat membuat dokumen teks yang kemudian akan dikirimkan ke dalam kotak surat juga pada mesin yang sama. Pada 1972, Ray mengenalkan icon @ sebagai identitas *e-mail* untuk memisahkan user id dan domain sebuah alamat *e-mail*.

Dalam pengembangan sistem informasi, email tidak hanya digunakan sebagai media komunikasi antar pengguna, tetapi juga sebagai sarana penyampaian informasi secara otomatis (*automatic notification*). Berbagai aplikasi memanfaatkan email untuk mengirimkan pemberitahuan, laporan, verifikasi akun, maupun dokumen digital kepada pengguna sehingga informasi dapat diterima dengan cepat dan terdokumentasi dengan baik.

### 2.8.2 Karakteristik *E-mail*

Secara struktur penulisan, e-mail merupakan bentuk turunan yang lebih ringkas dari surat. Dengan fitur-fitur yang dimiliki *e-mail*, pesan dapat ditulis tanpa menulis kembali penanggalan. Penanda tanggal akan otomatis saat e-mail masuk ke kotak masuk (*inbox*). Jika korespondensi mensyaratkan adanya bentuk surat konkret sebagai dokumen (*file word, JPEG, PDF, dll*) dapat dilampirkan sebagai attachment, dengan karakteristik :

1. Tidak terikat ruang dan waktu.
2. Menyimpan informasi apa saja yang telah berlangsung.
3. Memberikan waktu tenggang sebelum pesan dikirim.

4. Informasi dalam e-mail dapat disortir terlebih dahulu saat kita ingin meneruskan pesan ke orang lain.
5. Dapat menyertakan informasi tambahan (Widya Aryo Damar, 2023)

### **2.8.3 Google Mail**

*G-mail* merupakan layanan surat elektronik berbasis *web* yang dikembangkan oleh *Google* dan diperkenalkan kepada publik pada tahun 2004. *G-mail* menyediakan berbagai fasilitas, seperti kapasitas penyimpanan yang besar, sistem penyaringan spam, pencarian email yang cepat, serta dukungan terhadap protokol *SMTP*, *IMAP*, dan *POP3*. Fasilitas tersebut menjadikan *G-mail* sebagai salah satu layanan e-mail yang banyak digunakan oleh individu maupun organisasi untuk kebutuhan komunikasi dan integrasi dengan berbagai aplikasi.

Selain berfungsi sebagai layanan email bagi pengguna, Gmail juga menyediakan layanan *SMTP* (*Simple Mail Transfer Protocol*) yang memungkinkan aplikasi pihak ketiga mengirimkan email secara otomatis setelah melalui proses autentikasi. *Google* juga menerapkan berbagai mekanisme keamanan, seperti autentikasi pengirim dan penggunaan koneksi terenkripsi, untuk meningkatkan keamanan serta keandalan proses pengiriman *e-mail* (Master Dukom, 2023).

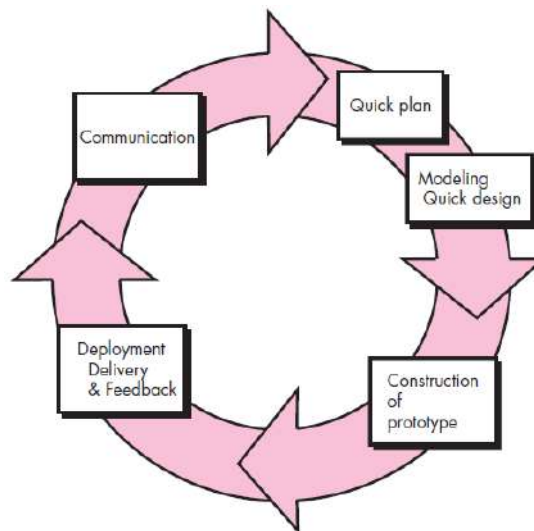
Pada penelitian ini, *G-mail* dimanfaatkan sebagai layanan *SMTP* untuk mengirimkan slip gaji kepada setiap guru melalui alamat *e-mail* yang telah terdaftar pada sistem. Dengan mekanisme tersebut, proses distribusi slip gaji dapat dilakukan secara otomatis, lebih cepat, mengurangi penggunaan kertas, serta menjaga kerahasiaan informasi gaji karena setiap slip dikirim langsung ke alamat *e-mail* masing-masing guru.

## **2.9 Metode Pengembangan Perangkat Lunak**

### **2.9.1 Metode Prototyping**

Metode *Prototyping* merupakan pendekatan rekayasa perangkat lunak yang berfokus pada pembuatan model kerja atau sketsa cepat guna memungkinkan pengguna berinteraksi dengan sistem sebelum tahap implementasi final. Menurut Pressman (2015), pendekatan ini sangat efektif untuk mendapatkan umpan balik langsung dari pengguna guna menyempurnakan kebutuhan spesifik sistem. Sejalan

dengan itu, Kustanto et al., (2025) mengaitkan *prototyping* erat dengan proses analisis dan desain, di mana prototipe berfungsi sebagai alat evaluasi untuk memvalidasi apakah arsitektur sistem informasi yang dibangun sudah sesuai dengan ekspektasi operasional di lapangan.



Gambar 2. 4 Paradigma *Prototype*

Sumber : Rhoedy Setiawan et al., (2025). Penerapan Metode *Prototype* dalam Pengembangan Sistem Informasi Kusiner, 14(2), hlm. 837

Dalam pengembangannya, metode ini menawarkan fleksibilitas melalui keterlibatan pengguna secara aktif selama proses berlangsung. *Prototipe* berperan sebagai jembatan komunikasi yang efektif antara pengembang dan pengguna untuk meminimalkan perbedaan persepsi terhadap kebutuhan sistem (Kustanto et al., 2025). Dengan menerapkan model ini, pengembang dapat menguji dan memvalidasi fungsionalitas sistem secara dini, sehingga risiko kesalahan logika pengembangan dapat dikurangi. Hal ini sangat relevan dalam membangun sistem presensi yang dinamis, di mana setiap iterasi atau perbaikan prototipe bertujuan untuk memastikan bahwa fitur kalkulasi insentif benar-benar akurat dan sesuai dengan prosedur internal lembaga (Setiawan et al., 2025).

### 2.9.2 Karakteristik Metode *Prototyping*

Metode *Prototyping* memiliki karakteristik utama berupa pengembangan yang bersifat iteratif (berulang) dan partisipatif. Menurut Pressman (2015), karakteristik unik dari model ini adalah fokus pada pembangunan representasi sistem yang bisa

dilihat dan diuji langsung oleh pengguna sebelum pengembangan skala penuh dilakukan. Karakteristik lainnya adalah adanya komunikasi dua arah yang intens antara pengembang dan pengguna, sehingga spesifikasi kebutuhan sistem menjadi lebih transparan dan mudah dipahami oleh kedua belah pihak (Kustanto et al., 2025).

### 2.9.3 Tahapan Pengembangan Metode *Prototyping*

Secara umum, tahapan pengembangan dalam penelitian ini mengikuti siklus sistematis menurut Pressman (2015) dalam jurnal karya Kustanto et al., (2025) yang disesuaikan dengan kebutuhan sistem presensi :

1. Pengumpulan kebutuhan (*Communication*), melalui tahap mengidentifikasi kendala pada sistem manual dengan cara mendengar keluhan dari pelanggan dan menentukan kebutuhan fitur seperti pemindaian *QR Code*.
2. Membangun *Prototype (Quick Design)*, perancangan desain antarmuka dan logika dasar sistem sebagai model kerja awal.
3. Evaluasi *Prototype*, adalah tahap di mana staf sekolah mencoba *prototype* untuk memastikan alur kerja sistem sudah sesuai.
4. Mengodekan Sistem, menerjemahkan desain yang telah disetujui ke dalam bahasa pemrograman *PHP*.
5. Menguji Sistem, dengan pengujian teknis menggunakan metode *Black Box* atau *White Box* untuk memastikan fungsionalitas fitur dan evaluasi akhir untuk memastikan aplikasi siap diimplementasikan.

### 2.9.4 Kelebihan dan Kekurangan Metode *Prototyping*

Penerapan metode *prototyping* dalam pengembangan perangkat lunak menawarkan sejumlah keunggulan signifikan, terutama dalam meningkatkan efektivitas komunikasi antara pengembang dan pengguna. Melalui pendekatan ini, staf sekolah dapat terlibat secara aktif sejak tahap awal pengerjaan, sehingga setiap kekurangan fitur atau kesalahan logika pada sistem presensi dapat dideteksi dan diperbaiki sebelum tahap pengkodean akhir selesai. Hal ini tidak hanya meminimalisir risiko revisi besar di akhir proyek, tetapi juga mempercepat pemahaman pengembang terhadap aturan internal organisasi yang spesifik dan

sering kali rumit jika hanya dijelaskan secara lisan (Fitria Nur H & Rahmania Sri U., 2023)

Di sisi lain, metode ini juga memiliki beberapa batasan teknis yang perlu diantisipasi selama proses penelitian berlangsung. Salah satu kelemahan utamanya adalah potensi terjadinya pengembangan yang meluas melebihi rencana awal (*scope creep*) apabila proses iterasi dilakukan secara berulang tanpa batasan waktu yang jelas. Selain itu, penggunaan metode ini terkadang menimbulkan persepsi yang keliru di sisi pengguna, di mana prototipe awal sering kali dianggap sebagai produk final yang sudah sempurna, padahal masih memerlukan pengujian mendalam pada aspek keamanan dan integritas basis data (Kustanto et al., 2025).

### 2.9.5 Alasan Penggunaan Metode *Prototyping*

Pemilihan metode *Prototyping* didasarkan pada argumen Satzinger et al., (2012) bahwa metode ini paling efektif jika kebutuhan pengguna masih belum terdefinisi secara kaku atau bersifat dinamis. Dalam kasus di SMA Asy-Syuja'iyah, aturan perhitungan insentif sering kali memiliki detail teknis yang hanya diketahui oleh staf TU saat melihat alur kerja secara visual. Dengan metode ini, risiko kegagalan sistem akibat salah interpretasi aturan sekolah dapat ditekan secara signifikan melalui validasi *prototipe* yang dilakukan secara berkala.

## 2.10 Website

### 2.10.1 Pengertian Website

*Website* merupakan kumpulan halaman digital dalam suatu domain di internet yang saling terhubung melalui konsep hyperlink untuk menyajikan informasi baik secara statis maupun dinamis.



Gambar 2. 5 Logo Website

Sumber : <https://www.flaticon.com/search?word=website>

Menurut Firdaus et al., (2022), sebuah situs *web* dibangun sebagai rangkaian bangunan informasi yang mencakup data teks, gambar, hingga elemen multimedia yang terintegrasi dalam suatu jaringan halaman. Dalam pengembangan sistem informasi sekolah, *website* tidak hanya berfungsi sebagai media penyebaran informasi searah, melainkan sebagai platform interaktif yang memungkinkan terjadinya pengolahan data secara real-time melalui jaringan *World Wide Web* (WWW) (Firdaus et al., 2022)

Secara teknis, sebuah *website* profesional dibangun atas sekumpulan halaman *web* yang mengandung informasi spesifik dan saling berkaitan satu sama lain (Yuhefizar dalam Marwanto et al., 2023). Dalam penelitian ini, *website* digunakan sebagai antarmuka utama (interface) bagi staf Tata Usaha dan pihak manajemen sekolah untuk memantau data presensi yang masuk. Implementasi teknologi berbasis web memungkinkan sistem dapat diakses secara fleksibel melalui berbagai perangkat, sehingga proses monitoring kinerja guru dan validasi insentif tidak lagi terhambat oleh batasan fisik perangkat keras tertentu, melainkan terpusat dalam satu domain yang terintegrasi.

#### **2.10.2 Jenis-Jenis Website**

Menurut Ade Johar Maturidi dalam buku yang berjudul "Pengembangan Website" (2023), *website* terbagi menjadi 2 kategori yaitu sebagai berikut.

##### **a. Website Berdasarkan Sifat**

###### **1. Website Statis**

*Website* statis adalah *website* yang isi didalam halaman *website* tersebut tidak mengalami perubahan dan hanya administratornya saja yang dapat merubah isi dalam *website* tersebut karena *website* statis tidak memerlukan database yang otomatis tidak selalu *update* informasi pada halamannya. *Website* statis biasanya dipakai oleh perusahaan yang hanya memberikan informasi mendasar seperti alamat, kontak, dan profil perusahaan saja.

###### **2. Website Dinamis**

*Website* dinamis adalah keterbalikan dari *website* statis, dimana isi didalam halamannya selalu berubah secara berkala. *Website* ini sangat

mudah dikelola karena menggunakan *database* untuk menampilkan hasil kontennya yang hanya dapat diakses oleh administratornya (*webmaster/developer*).

b. *Website Berdasarkan Platform*

1. *Content Management System (CMS)*

*CMS* adalah sistem management konten yang biasa digunakan untuk membuat website secara instant dan tidak membutuhkan usaha berlebih, tinggal klik dan menunggu beberapa saat saja maka *website* sudah dapat dimiliki. Contohnya seperti *WordPress*, *Magento*, *Joomla*, *Drupal*, dll.

2. *Website Builder*

*Website* ini sangat mudah digunakan karena memiliki platform yang membantu dalam proses pembuatan *website* dengan cepat tanpa harus memahami *coding* dan kemampuan desain. Beberapa contoh *website* ini adalah *Wix*, *Site*, dan *Weebly*.

3. *HTML dan CSS*

*Cascading Style Sheets (CSS)* adalah bahasa yang digunakan untuk menentukan dan format halaman website. Dengan *CSS* kita bisa mengatur jenis *font*, warna tulisan, dan latar belakang *website* kita. *CSS* digunakan bersama dengan bahasa *markup*, seperti *HTML* dan *XML* (Maturidi J.A., 2023 : 32)

## 2.11 Teknologi Pengembangan Website

### 2.11.1 *Hypertext Preprocessor (PHP)*

Dalam buku "Mahir Bahasa Pemrograman PHP" (2018), Miftahul Jannah dan Sarwandi mendefinisikan PHP merupakan bahasa pemrograman script server-side yang didesain untuk pengembangan web. Disebut demikian karena diproses pada computer server, berbeda dengan Bahasa pemrograman client-side seperti javascript yang diproses didalam web server. *PHP* pertama kali dibuat oleh Ramus Lerdorf pada tahun 1995 dan merupakan kependekan dari "*Personal Home Page*" dengan nama *Form Interpreted (FI)* yang wujudnya berupa sekumpulan skrip dan digunakan untuk mengolah data formulir dari web.



Gambar 2. 6 Logo PHP

Sumber : <https://friconix.com/icon/fi-xnsuxl-php-logo>

Secara teknis, *PHP* bekerja sebagai interpreter yang menerjemahkan setiap baris kode sumber menjadi instruksi yang dipahami langsung oleh mesin saat dijalankan. Karena kemampuannya yang terintegrasi dengan berbagai sistem manajemen basis data relasional (*RDBMS*), *PHP* menjadi pilihan yang efisien untuk membangun sistem informasi yang memerlukan pengolahan data kompleks. Dalam penelitian ini, *PHP* digunakan untuk membangun logika sistem presensi, mulai dari pemrosesan data *QR Code* hingga kalkulasi otomatis insentif guru yang disimpan ke dalam database (Krisno To S, Nirsal 2023).

### 2.11.2 *HyperText Markup Language (HTML)*

Pratama Angga Buana dalam buku "*Pengembangan Website*" (2023) berpendapat bahwa *HTML* adalah sebuah bahasa pemrograman tingkat dasar untuk sebuah pemrograman website karena untuk dapat melanjutkan ke pemrograman berbasis web tingkat tinggi seperti *PHP*, wajib terdapat kerangka *HTML* terlebih dahulu. Hal ini telah distandarisasi melalui W3C (World Wide Web Consortium) dan telah mendapatkan pengakuan dunia internasional untuk kontribusi dalam perkembangan web.

*HTML* berfungsi untuk menyusun struktur dan mengatur elemen-elemen yang terdapat pada halaman web, seperti judul, paragraf, gambar, tabel, daftar, formulir, maupun tautan. Struktur tersebut menjadi acuan bagi peramban (*browser*) dalam menampilkan informasi kepada pengguna sehingga halaman web dapat ditampilkan dengan susunan yang sesuai dengan rancangan pengembang.

Perkembangan teknologi web telah mendorong *HTML* terus mengalami penyempurnaan. Salah satu perkembangan yang banyak digunakan saat ini adalah *HTML5* yang menghadirkan berbagai elemen baru serta peningkatan dukungan terhadap pengembangan aplikasi berbasis web. Kehadiran *HTML5* memberikan kemudahan bagi pengembang dalam membangun website yang lebih terstruktur, responsif, dan mampu memanfaatkan berbagai fitur yang didukung oleh peramban modern (Buana A.P., 2023 : 102).

### **2.11.3 *Cascading Style Sheets (CSS)***

Menurut Iwan Adhicandra dalam buku "*Pengembangan Website*" (2023) *CSS* merupakan bahasa yang digunakan untuk mengatur tampilan dan penyajian suatu halaman website. *CSS* berfungsi mengendalikan berbagai aspek visual, seperti warna, jenis huruf, ukuran teks, jarak antar elemen, tata letak, hingga penyesuaian tampilan pada berbagai ukuran layar. Dengan adanya *CSS*, struktur halaman yang dibuat menggunakan *HTML* dapat disajikan dalam bentuk yang lebih rapi, menarik, dan mudah dipahami oleh pengguna.

Seiring dengan perkembangan teknologi web, *CSS* terus mengalami penyempurnaan untuk mendukung kebutuhan desain antarmuka yang semakin beragam. Berbagai fitur yang tersedia pada *CSS* memungkinkan pengembang membuat tata letak yang lebih fleksibel, responsif, serta mampu menyesuaikan tampilan pada berbagai jenis perangkat, seperti komputer, tablet, maupun telepon pintar. Selain itu, *CSS* juga mendukung pembuatan efek visual sederhana, animasi, dan transisi yang dapat meningkatkan kenyamanan pengguna saat berinteraksi dengan halaman website (Adhicandra I., 2023 : 166)

## **2.12 Basis Data (*Database*) dan *MySQL***

Basis data merupakan sekumpulan informasi yang terorganisir secara digital dalam sebuah wadah yang memungkinkan data tersebut diakses, dikelola, dan diperbarui dengan mudah. Menurut Andi Asari et al., dalam bukunya yang berjudul "*Sistem Basis Data*" (2024), basis data berfungsi sebagai media penyimpanan terpusat yang memuat seluruh entitas dan relasi data yang diperlukan untuk mendukung operasional suatu aplikasi. Dalam konteks sistem presensi ini, basis

data bertindak sebagai media penyimpanan utama untuk seluruh informasi guru, catatan waktu mengajar, hingga akumulasi nominal insentif yang bersifat sensitif dan memerlukan tingkat akurasi tinggi (Andi Asari et al., 2024) .

Untuk mengelola pangkalan data tersebut secara efektif, diperlukan sebuah *Relational Database Management System (RDBMS)*, di mana *MySQL* menjadi pilihan yang paling umum digunakan karena sifatnya yang open-source dan efisien (Jurisman Waruwu et al., 2024). Lukas Vileikis (2024) dalam bukunya yang berjudul *Hacking MySQL* menjelaskan bahwa *MySQL* mampu menangani volume data yang besar serta mendukung akses multi-pengguna secara bersamaan tanpa mengorbankan integritas data. Keunggulan *MySQL* terletak pada skalabilitas dan keamanannya, yang menjamin bahwa data perbendaharaan sekolah terkait honorarium guru tetap konsisten dan terlindungi dari risiko manipulasi data yang tidak sah (Andreki Ariesta et al., 2025)

Implementasi basis data *MySQL* dalam pengembangan web ini berfungsi sebagai jembatan antara logika pemrograman *PHP* dengan penyedia data fisik di sisi server. Menurut Madcoms (2025), keterhubungan ini memungkinkan data di dalam basis data dimanipulasi sesuai dengan kebutuhan fungsional sistem, seperti proses validasi kode *QR* maupun penarikan laporan bulanan. Dengan demikian, sinkronisasi antara perangkat lunak pengelola (*DBMS*) dan basis data itu sendiri merupakan aspek krusial untuk menjamin bahwa seluruh transaksi keuangan sekolah dapat berjalan secara transparan dan akuntabel.

## **2.13 Tools Pendukung Pengembangan Website**

### **2.13.1 XAMPP**

*XAMPP* merupakan perangkat lunak yang digunakan sebagai lingkungan *web* server lokal untuk mendukung proses pengembangan aplikasi berbasis *web*. Melalui *XAMPP*, beberapa komponen utama seperti *Apache*, *MySQL* atau *MariaDB*, *PHP*, serta *phpMyAdmin* telah terintegrasi dalam satu paket sehingga memudahkan proses instalasi dan konfigurasi server pada komputer pengembang. Dengan adanya integrasi tersebut, pengembang dapat menjalankan aplikasi

berbasis web beserta pengelolaan basis datanya tanpa harus melakukan konfigurasi setiap komponen secara terpisah.

*XAMPP* banyak dimanfaatkan sebagai media pembelajaran maupun pengembangan aplikasi karena menyediakan lingkungan yang menyerupai server sebenarnya. Pengguna dapat melakukan pembuatan basis data, pengelolaan tabel melalui *phpMyAdmin*, serta menghubungkan basis data dengan aplikasi berbasis *PHP* secara lebih mudah. Pemanfaatan *XAMPP* juga membantu proses pengujian aplikasi sebelum sistem diimplementasikan pada server produksi sehingga kesalahan yang muncul selama proses pengembangan dapat diketahui lebih awal (Kurniawansyah et al., 2025) .

### 2.13.2 *Visual Studio Code*

*Visual Studio Code* merupakan perangkat lunak *source code* editor yang digunakan untuk membantu proses penulisan, penyuntingan, dan pengujian kode program. Editor ini mendukung berbagai bahasa pemrograman melalui penyediaan fitur seperti penyorotan sintaks (*syntax highlighting*), pelengkapan kode (*code completion*), serta integrasi dengan berbagai ekstensi yang dapat disesuaikan dengan kebutuhan pengembang. Ketersediaan fitur tersebut menjadikan *Visual Studio Code* banyak digunakan sebagai media pengembangan perangkat lunak karena mampu meningkatkan efisiensi dalam proses penulisan program (Abidah et al., 2025).

## 2.14 *Unified Modeling Language (UML)*

Sebagai tokoh pengembang asli UML (sering disebut *The Three Amigos*), James Rumbaugh, Grady Booch, dan Ivar Jacobson (2005) mendefinisikan *UML* sebagai bahasa visual untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. Sedangkan menurut Pressman, *UML* adalah standar bahasa pemodelan yang menyediakan sekumpulan teknik notasi grafis untuk membuat model visual dari sistem perangkat lunak yang berorientasi objek (Dr.Henderi et al., 2023).

Menurut Salam et al., (2023), penggunaan *UML* sangat penting dalam rancang bangun aplikasi berbasis objek karena dapat mempermudah pengembang dalam memetakan interaksi antara pengguna dan sistem secara logis (Salam et al., 2023). Hal ini karena *UML* menyediakan Bahasa permodelan visual yang memungkinkan pengembang sistem membuat blue print atas visinya dalam bentuk yang baku (Henderi et al, 2023).

Tabel 2. 1 Komponen Umum *UML*

| No | Gambar                                                                              | Nama                  | Keterangan                                                                                        |
|----|-------------------------------------------------------------------------------------|-----------------------|---------------------------------------------------------------------------------------------------|
| 1  |    | <i>Activity</i>       | Memperlihatkan bagaimana masing-masing <i>class</i> antarmuka saling berinteraksi satu sama lain. |
| 2  |   | <i>Action</i>         | <i>State</i> dari <i>system</i> yang mencerminkan eksekusi dari suatu aksi.                       |
| 3  |  | <i>Initial State</i>  | Awal dimulainya suatu aliran kerja pada <i>activity diagram</i> .                                 |
| 4  |  | <i>Activity State</i> | Bagian akhir dari suatu aliran kerja pada sebuah <i>activity diagram</i> .                        |
| 5  |  | <i>Fork Node</i>      | Sebuah aliran pada tahap tertentu berubah menjadi beberapa aliran.                                |

Sumber : *UML Powered*, 2023

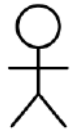
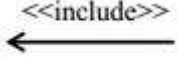
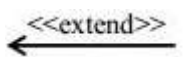
*UML* mempunyai banyak diagram yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. Diagram-diagram tersebut digunakan untuk mengkomunikasikan ide, melahirkan ide dan peluang baru, menguji ide dan membuat prediksi, serta membantu memahami struktur dan relasinya (Rahayu Swastika, 2024).

#### 2.14.1 Use Case Diagram

*Use Case diagram* digunakan untuk menggambarkan hubungan antara aktor (pengguna) dengan sistem yang dibangun. Diagram ini berfungsi untuk mendefinisikan fitur-fitur apa saja yang tersedia di dalam aplikasi pembayaran

sekolah, seperti proses login, input data siswa, hingga pencetakan laporan (Sati et al., 2023).

Tabel 2. 2 Komponen *Use Case Diagram*

| No | Gambar                                                                              | Nama                | Keterangan                                                                                                                       |
|----|-------------------------------------------------------------------------------------|---------------------|----------------------------------------------------------------------------------------------------------------------------------|
| 1  |    | <i>Actor</i>        | Mewakili peran orang, <i>system</i> yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .                           |
| 2  |    | <i>Use Case</i>     | Abstraksi dan interaksi antara <i>system</i> dan <i>actor</i> .                                                                  |
| 3  |    | <i>Association</i>  | Abstraksi dari penghubung antara <i>actor</i> dengan <i>use case</i> .                                                           |
| 4  |  | <i>Generalisasi</i> | Menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>use case</i> .                                        |
| 5  |  | <i>include</i>      | Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.                        |
| 6  |  | <i>extend</i>       | Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi. |

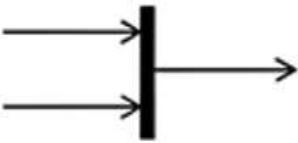
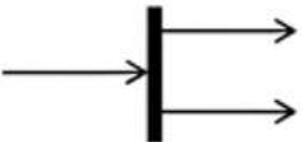
Sumber : Dicoding.com (2024)

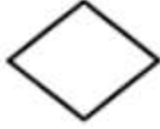
Secara umum, *Use Case* memiliki 3 level, yaitu *sea-level* yang biasanya mewakili interaksi diskrit antara *actor* utama dan sistem, *fish-level* ada karena di-*include* oleh *Use Case sea-level*, dan *kite-level* menunjukkan bagaimana *use case sea-level* sesuai dengan interaksi bisnis yang lebih luas sedangkan level pertama dan kedua itu khusus untuk sistem. Setiap *use case* dapat membuat satu diagram aktivitas. (Rahayu Swastika, 2024).

### 2.14.2 Activity Diagram

*Activity diagram* adalah diagram yang digunakan untuk memodelkan proses dalam suatu sistem informasi dengan menggambarkan kegiatan utama dan hubungan diantara kegiatan dalam suatu proses (Rahayu Swastika, 2024).

Tabel 2. 3 Komponen *Activity Diagram*

| Simbol                                                                              | Nama                           | Keterangan                                                                                   |
|-------------------------------------------------------------------------------------|--------------------------------|----------------------------------------------------------------------------------------------|
|    | <i>Activity</i>                | Merepresentasikan suatu aktivitas atau proses yang dilakukan dalam alur kerja sistem.        |
|    | <i>Control Flow</i>            | Menunjukkan urutan perpindahan dari satu aktivitas ke aktivitas berikutnya.                  |
|  | <i>Object Flow</i>             | Menggambarkan perpindahan objek atau data antaraktivitas dalam sistem.                       |
|  | <i>Start Point</i>             | Menunjukkan titik awal dimulainya suatu proses atau aktivitas.                               |
|  | <i>End Point</i>               | Menunjukkan titik akhir dari suatu proses atau aktivitas.                                    |
|  | <i>Join /<br/>Penggabungan</i> | Digunakan untuk menggabungkan kembali beberapa aktivitas yang berjalan secara paralel.       |
|  | <i>Fork</i>                    | Digunakan untuk membagi satu alur menjadi beberapa aktivitas yang berjalan secara bersamaan. |

| Simbol                                                                            | Nama            | Keterangan                                                                  |
|-----------------------------------------------------------------------------------|-----------------|-----------------------------------------------------------------------------|
|  | <i>Decision</i> | Menunjukkan percabangan proses berdasarkan kondisi atau keputusan tertentu. |

Sumber : Ramdany et al. (2024)

Setiap komponen pada *Activity Diagram* memiliki peran yang berbeda dalam menggambarkan alur kerja sistem. *Start Point* digunakan sebagai penanda dimulainya proses, sedangkan *Activity* merepresentasikan setiap aktivitas yang dilakukan selama sistem berjalan. Apabila terdapat kondisi tertentu yang memerlukan percabangan proses, maka digunakan simbol *Decision*. Sementara itu, simbol *Fork* dan *Join* digunakan untuk menggambarkan aktivitas yang berlangsung secara paralel beserta proses sinkronisasinya sebelum alur kembali menjadi satu. Seluruh rangkaian aktivitas tersebut diakhiri dengan *End Point* yang menandakan bahwa proses telah selesai. Dengan memahami fungsi setiap komponen tersebut, *Activity Diagram* dapat digunakan untuk merepresentasikan alur sistem secara jelas, terstruktur, dan mudah dipahami oleh pengembang maupun pengguna (Wayahdi & Ruziq, 2023).

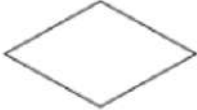
### 2.14.3 *Entity-Relationship Diagram (ERD)*

*Entity-Relationship Diagram (ERD)* merupakan sebuah teknik pemodelan data konseptual yang digunakan untuk merancang struktur basis data relasional secara sistematis. *ERD* berfungsi untuk menggambarkan hubungan antar data yang tersimpan dalam sistem melalui pemodelan entitas, atribut, dan relasi Brady & Loonam (2010). Secara praktis, teknik ini digunakan oleh *System Analyst* pada tahap analisis persyaratan untuk memetakan kebutuhan data organisasi ke dalam model yang dapat dipahami secara teknis maupun manajerial (Prima Eza P., & Kartika Khairunnisa, 2025)

Sejarah perkembangan *ERD* tidak lepas dari kontribusi Charles Bachman dan James Martin yang memperjelas simbolisasi entitas serta relasi dalam pemodelan data. *Entity-Relationship Diagram (ERD)* adalah diagram berbentuk notasi grafis

yang berada dalam pembuatan database yang menghubungkan antara data satu dengan yang lain (Zaimah Fira A, & Azaroby Dwi A, 2025).

Tabel 2. 4 Komponen *Entity Relationship Diagram* (ERD)

| Simbol                                                                              | Nama    | Keterangan                                                                                               |
|-------------------------------------------------------------------------------------|---------|----------------------------------------------------------------------------------------------------------|
|    | Entitas | Merepresentasikan objek atau konsep yang datanya disimpan di dalam basis data.                           |
|    | Relasi  | Menunjukkan hubungan yang terjadi antar entitas dalam sistem.                                            |
|    | Atribut | Menjelaskan karakteristik yang dimiliki oleh suatu entitas, termasuk atribut yang berfungsi sebagai key. |
|  | Garis   | Menghubungkan entitas dengan relasi maupun atribut yang berkaitan.                                       |

Sumber : Triaulia, E. (2021)

Fungsi *ERD* adalah sebagai alat bantu dalam pembuatan database dan memberikan gambaran bagaimana kerja database yang akan dibuat. Di dalam *ERD* terdapat 3 elemen dasar, yaitu entitas, atribut, dan relasi (Zaimah Fira A, & Azaroby Dwi A, 2025).

#### 2.14.4 *Class Diagram*

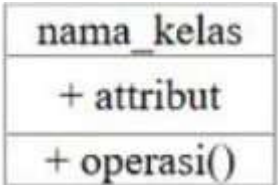
*Class Diagram* merupakan diagram yang digunakan untuk memodelkan struktur sistem dari sudut pandang pemrograman berorientasi objek. Menurut Satzinger, Jackson, dan Burd (2012), diagram ini berfungsi untuk menunjukkan kelas-kelas objek dalam sistem serta hubungan antar objek tersebut. Fokus utama dari class diagram adalah pada komponen internal dan organisasi data yang mendefinisikan bagaimana sistem akan berperilaku secara teknis.

Secara visual, class diagram menggambarkan deskripsi kelas serta hubungan relasional antar kelas lainnya. Meskipun sekilas memiliki kemiripan dengan Entity Relationship Diagram (*ERD*) pada pemodelan terstruktur, terdapat perbedaan fundamental pada elemen yang dimuat (Rizki & Zulkifli, 2026). Jika *ERD* hanya

berfokus pada atribut data, class diagram mencakup tiga komponen utama dalam setiap kelasnya, yaitu :

1. Nama Kelas (Class Name): Identitas unik dari kelas tersebut.
2. Atribut (Attributes): Variabel atau data yang dimiliki oleh kelas.
3. Metode/Operasi (Methods/Operations): Fungsi atau tindakan yang dapat dilakukan oleh kelas tersebut (seperti create, edit, atau calculate).

Tabel 2. 5 Komponen *Class Diagram*

| No | Gambar                                                                              | Nama               | Keterangan                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----|-------------------------------------------------------------------------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  |   | <i>Class</i>       | Kelas pada struktur system, didalamnya terdapat himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. Attribute merupakan property yang menggambarkan keadaan suatu objek dan dapat diturunkan dari attribute lain. Operation mewakili tindakan atau fungsi kelas, dapat diklasifikasikan sebagai konstruktor, permintaan, atau operasi pembaruan. Termasuk tanda kurung yang berisi parameter atau informasi yang diperlukan untuk operasi. |
| 2  |  | <i>Association</i> | Relasi antar beberapa class atau dirinya sendiri, dilabeli menggunakan frasa kata kerja atau nama peran, berisi symbol multiplisitas.                                                                                                                                                                                                                                                                                                                          |
| 3  |  | Generalization     | Relasi sejenis diantara beberapa class.                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 4  |  | Aggregation        | Relasi sebagian (a part of) yang logis diantara beberapa class atau suatu kelas dengan kelas itu sendiri, terbuat dari objek-objek yang bisa dibagi atau                                                                                                                                                                                                                                                                                                       |

| No | Gambar                                                                            | Nama                     | Keterangan                                                                                                                                                         |
|----|-----------------------------------------------------------------------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                                                                                   |                          | ditukar, dengan angka boleh lebih dari 1.                                                                                                                          |
| 5  |  | Composition              | Relasi fisik antara beberapa class dan class itu sendiri, terdiri dari objek-objek yang bisa dibagi atau ditukar, dengan angka disebelah kanan boleh lebih dari 1. |
| 6  |  | Transition / Association | Menghubungkan aktivitas selanjutnya setelah aktivitas sebelumnya.                                                                                                  |

Sumber : Buku ajar pemodelan sistem berbasis objek

*Class diagram* adalah model statis yang menunjukkan kelas dan hubungan diantara kelas yang tetap konstan dalam sistem dari waktu ke waktu, menggambarkan kelas, yang meliputi perilaku dan keadaan, dengan hubungan antar kelas (Rahayu Swastika, 2024).

## 2.15 Metode Pengujian Sistem

Metode pengujian sistem merupakan tahapan yang dilakukan untuk memastikan bahwa sistem yang telah dikembangkan dapat berfungsi sesuai dengan kebutuhan dan tujuan yang telah ditetapkan. Proses pengujian dilakukan dengan memeriksa setiap fungsi pada sistem guna mengetahui apakah seluruh fitur mampu menghasilkan keluaran yang sesuai dengan masukan yang diberikan. Melalui pengujian sistem, kesalahan atau ketidaksesuaian fungsi dapat diidentifikasi sehingga sistem yang dihasilkan memiliki tingkat keandalan yang lebih baik sebelum digunakan oleh pengguna.

*Black Box Testing* merupakan metode pengujian perangkat lunak yang berfokus pada pengujian fungsi sistem berdasarkan masukan (*input*) dan keluaran (*output*) yang dihasilkan tanpa memperhatikan struktur atau kode program di dalamnya. Pengujian dilakukan dengan mengacu pada spesifikasi kebutuhan sistem

sehingga penilaiannya didasarkan pada kesesuaian fungsi yang dijalankan dari sudut pandang pengguna. Melalui pendekatan ini, penguji tidak perlu memahami proses implementasi program, melainkan cukup memastikan bahwa setiap fitur menghasilkan keluaran yang sesuai dengan kebutuhan yang telah ditetapkan

Metode *Black Box Testing* banyak digunakan untuk memverifikasi fungsionalitas perangkat lunak karena mampu membantu menemukan kesalahan pada proses input maupun output tanpa harus melakukan pemeriksaan terhadap kode sumber. Setiap fungsi diuji menggunakan sejumlah skenario yang telah dirancang sebelumnya, kemudian hasil pengujian dibandingkan dengan keluaran yang diharapkan. Apabila hasil yang diperoleh sesuai dengan spesifikasi, maka fungsi tersebut dapat dinyatakan berjalan dengan baik. Oleh karena itu, metode ini dinilai efektif untuk memastikan bahwa sistem telah memenuhi kebutuhan fungsional sebelum digunakan oleh pengguna (Ardiansyah R et al., 2024).

## **2.16 Penelitian Terdahulu**

Penelitian terdahulu merupakan salah satu landasan yang digunakan untuk memperoleh gambaran mengenai penelitian yang telah dilakukan sebelumnya pada bidang yang relevan. Melalui kajian terhadap penelitian terdahulu, penulis dapat memahami perkembangan metode, teknologi, serta pendekatan yang telah diterapkan dalam pengembangan sistem informasi presensi, pengelolaan data kehadiran, maupun sistem yang berkaitan dengan proses administrasi penggajian. Selain itu, kajian ini juga menjadi referensi dalam menentukan metode dan teknologi yang sesuai untuk diterapkan pada penelitian yang dilakukan.

Penelitian oleh Andriana & Priyadi (2023), Putu Maha Putra et al. (2025), Rosiana et al. (2025), Fatwa & Pirous (2024), Rizaldi (2024), serta Abadi et al. (2025) berfokus pada pengembangan sistem penggajian guru/pegawai. Namun, penginputan data kehadiran masih bergantung pada catatan manual, belum memanfaatkan teknologi pemindaian *QR Code*, dan tidak memvalidasi keberadaan fisik pengguna di lokasi mengajar.

Penelitian yang dilakukan oleh Festiana & Komarudin (2023), Gunawan et al. (2021), *Management of Teacher Discipline* (2023), *Design of a Web-Based*

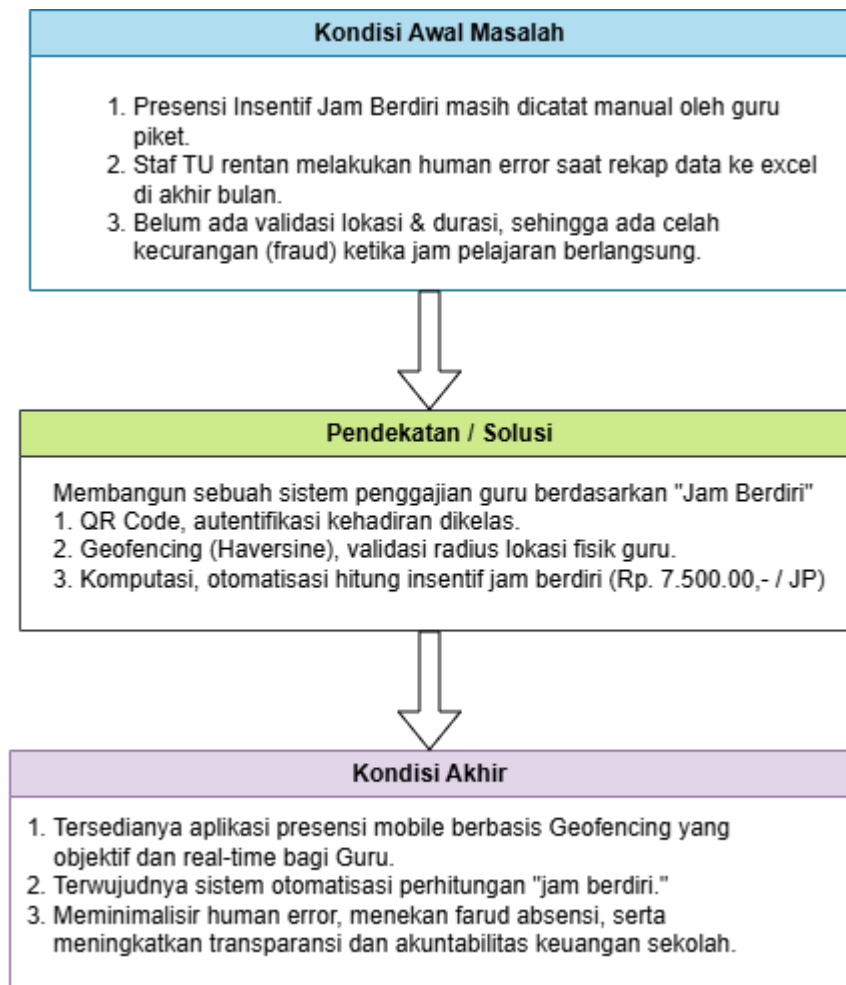
*Lecturer* (2023), *Development of Student E-Attendance* (2022), serta *Web-Based Teacher Attendance* (2022) telah menggunakan teknologi Kode QR atau situs web untuk melakukan presensi secara real-time. Namun demikian, sistem-sistem tersebut hanya berhenti pada fungsi rekapitulasi kehadiran dan belum melakukan korelasi data presensi sebagai dasar untuk menghitung insentif, honorarium, atau kinerja pengajar.

Penelitian oleh Zahroh et al. (2024) serta Sutjiadi & Kurniawan (2023) telah mengintegrasikan lokasi atau insentif. Namun, validasi presensi belum menggunakan mekanisme *double-validation* (*QR Code* + *Geofencing*) serta belum mengakomodasi variabel durasi nyata mengajar (seperti Insentif Yayasan, jam hidup, jam masuk kelas, insentif mengajar, jam berdiri, dan sebutan lainnya) sebagai parameter utama penghitungan.

Berdasarkan paparan penelitian-penelitian terdahulu tersebut, dapat disimpulkan bahwa belum ada penelitian yang secara spesifik mengintegrasikan validasi presensi ganda (*QR Code* dan *Geofencing*) berbasis lokasi fisik kelas dengan sistem kalkulasi penggajian insentif "Jam Berdiri" bagi guru secara otomatis dan *real-time*.

## **2.17 Kerangka Berpikir**

Penyusunan kerangka berpikir ini didasarkan pada alur pemikiran yang sistematis untuk menyelesaikan permasalahan tata kelola administrasi di SMA Asy-Syuja'iyah. Kerangka berpikir dalam penelitian ini dibagi menjadi beberapa tahapan, yaitu :



Gambar 3. 1 Kerangka Berpikir Tata Kelola Administrasi SMAASY  
Sumber : Diolah oleh peneliti (2026)

Berdasarkan bagan kerangka berpikir pada gambar 3.2, penelitian ini diawali dari kondisi awal di SMA Asy-Syuja'iyah, dimana sistem pencatatan kehadiran mengajar untuk insentif "Jam Berdiri" masih bersifat manual. Hal tersebut menimbulkan masalah berupa resiko kesalahan hitung oleh staf tata usaha, serta tidak adanya alat validasi objektif terhadap durasi dan lokasi mengajar guru dikelas.

Sebagai pendekatan/solusi, untuk mengatasi masalah tersebut, penulis mengusulkan sistem informasi presensi berbasis *QR Code* dan *Geolocations*. *QR Code* ditempatkan di setiap kelas untuk dipindai oleh guru, sedangkan teknologi *Geofencing* dengan *algoritma Haversine* akan menghitung jarak koordinat *GPS* perangkat guru terhadap titik pusat sekolah agar presensi hanya sah jika dilakukan dalam radius 50 meter.

Proses pengembangan ini dipandu menggunakan metode *prototyping* secara iteratif guna memastikan perangkat lunak yang dibangun menggunakan Bahasa pemrograman *PHP Native* dan basis data *MySQL* benar-benar adaptif dengan regulasi sekolah. Melalui implementasi tersebut, diperoleh kondisi akhir berupa sistem penggajian digital yang mampu menghitung nilai insentif secara otomatis, meminimalisir *human error*, serta menciptakan laporan kehadiran yang akurat dan transparan bagi bendahara maupun kepala sekolah.