

## BAB III

### PEMBAHASAN

#### 3.1. Tahap Analisis Perencanaan (*Requirement Analysis*)

Pada bab ini, kami akan menguraikan analisa sistem berjalan dari mitra dan uraikan dalam setiap prosedur berjalannya. Tahap Analisis Perencanaan merupakan langkah awal dalam proses penelitian bertujuan untuk memastikan bahwa sistem yang akan dikembangkan benar-benar menjawab permasalahan operasional dan manajerial yang terjadi di MTs Irsyadul Athfal Depok. Pada tahap ini, peneliti menyusun arah penelitian, menentukan pendekatan yang digunakan, serta mengidentifikasi kebutuhan yang diperlukan untuk membangun sistem penggajian online dengan fitur prediksi anggaran berbasis *machine learning*.



Gambar III.1.

Kerangka Pemikiran

Berikut penjelasan kerangka pemikiran pada Gambar III.1. sebagai berikut :

### 1. Permasalahan

Permasalahan yang terjadi di MTs Irsyadul Athfal Depok adalah:

- a. Proses absensi masih manual (tanda tangan kertas).
- b. Perhitungan gaji menggunakan Excel sehingga sering terjadi kesalahan.
- c. Rekapitulasi gaji memakan waktu lama.
- d. Tidak ada sistem prediksi anggaran gaji untuk tahun berikutnya.
- e. Slip gaji tidak terdigitalisasi dan harus dicetak satu per satu.
- f. Data tidak terpusat sehingga rentan hilang atau tidak sinkron.

Permasalahan-permasalahan ini menjadi dasar perlunya sistem penggajian yang terintegrasi dengan fitur prediksi anggaran.

### 2. Konstruksi

Konstruksi yang dirancang untuk mengatasi permasalahan tersebut mencakup:

- a. Digitalisasi data karyawan & jabatan dalam basis data terpusat.
- b. Sistem absensi online yang otomatis masuk ke perhitungan gaji.
- c. Perhitungan gaji otomatis sesuai komponen (gaji pokok, tunjangan, potongan).
- d. Pembuatan slip gaji digital yang dapat dicetak atau diunduh.
- e. Dashboard laporan gaji bulanan/tahunan.
- f. Integrasi algoritma Regresi Linear Sederhana untuk memprediksi total anggaran gaji untuk periode berikutnya.
- g. Sistem dibangun berbasis web dengan *PHP Native* dan *MySQL*.

Konstruksi ini menggambarkan alur solusi dari input → proses → output → prediksi → pelaporan.

### 3. Implementasi

Sistem penggajian dan absensi online diimplementasikan sesuai dengan perancangan dan diuji oleh admin dan guru (user) untuk memastikan fitur berjalan sesuai harapan.

### 4. Pegujian

Melakukan pengujian terhadap kinerja sistem (performance) dan keamanan (security) untuk memastikan bahwa aplikasi dapat digunakan secara aman dan efisien.

## **3.2. Tahap Desain Sistem (*System Design*)**

Tahap desain sistem merupakan langkah penting dalam proses pengembangan sistem, fase krusial di mana kita mengubah rumusan "Apa yang harus dilakukan oleh sistem" (dari tahap Analisis Kebutuhan) menjadi "Bagaimana sistem akan melakukannya" secara teknis. Ibarat membangun gedung, ini adalah proses pembuatan "cetak biru" (blueprint) secara mendetail sebelum *programmer* mulai menyusun batu bata (menulis kode).

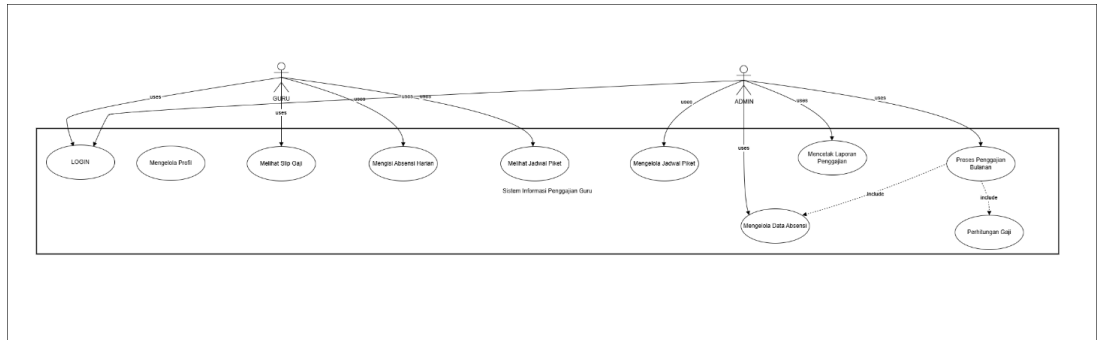
Tahap ini mencakup dua proses utama, yaitu Tahap Analisis dan Tahap Desain, yang menjadi fondasi dalam merancang sistem penggajian online beserta fitur prediksi anggaran berbasis machine learning.

### **3.2.1. Pemodelan Alur dan Proses Sistem (*System Modeling*)**

Tahap analisis bertujuan untuk memetakan bagaimana sistem akan bekerja secara logis dan bagaimana aktor berinteraksi dengan sistem tersebut. Alat yang biasanya digunakan adalah UML (Unified Modeling Language).

1. **Desain Use Case Diagram:** Menggambarkan batasan sistem, mengetahui

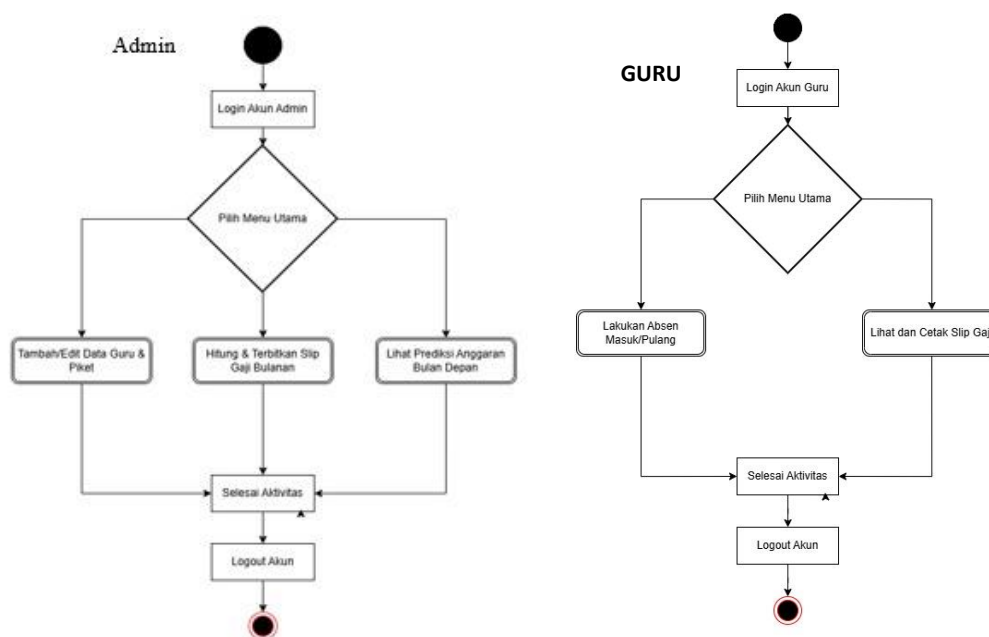
siapa saja aktor yang terlibat, dan apa saja hak akses (fungsionalitas) yang bisa mereka lakukan di dalam aplikasi.



Gambar III.2.

*Use Case Diagram*

- A. Aktor Guru: Hanya memiliki akses ke fitur pendaftaran/login, melakukan absensi (datang/pulang) dengan validasi lokasi, dan melihat/mengunduh slip gaji pribadi.
  - B. Aktor Admin: Memiliki akses penuh (CRUD) ke data master Guru, mengelola jadwal piket, memproses perhitungan gaji bulanan, dan mengeksekusi fitur prediksi anggaran.
2. Desain Activity Diagram: Merancang urutan langkah per aktivitas. Misalnya, merancang alur keputusan (Decision Node) pada saat absensi:



Gambar III.3.

*Activity Diagram*

Berikut adalah penjelasan langkah demi langkah alur aktivitas pengguna di dalam sistem:

## 1. Alur aktivitas Guru)

- A. Langkah 1: Mulai. Pengguna (Guru) membuka aplikasi/sistem.
- B. Langkah 2: Login Akun Guru. Guru wajib memasukkan kredensial (seperti *username* dan *password*) untuk masuk ke dalam sistem.
- C. Langkah 3: Pilih Menu Utama. Setelah berhasil *login*, sistem akan menampilkan pilihan menu. Di sini terjadi percabangan (*decision*) di mana Guru dapat memilih salah satu dari dua menu yang tersedia:
  1. Jalur A (Menu Absen): Jika Guru memilih menu ini, langkah selanjutnya adalah melakukan proses "Lakukan Absen Masuk / Pulang".
  2. Jalur B (Menu Gaji): Jika Guru memilih menu ini, langkah selanjutnya adalah melakukan proses "Lihat & Cetak Slip Gaji".

PDF" untuk melihat rincian gaji mereka.

- D. Langkah 4: Selesai Aktivitas. Setelah melakukan salah satu aktivitas di atas (baik absen maupun mengecek gaji), aktivitas utama Guru dianggap selesai.
- E. Langkah 5: Logout Akun. Guru keluar dari akun mereka demi keamanan.
- F. Langkah 6: Selesai. Alur sistem untuk Guru berakhir.

## 2. Alur Aktivitas Admin

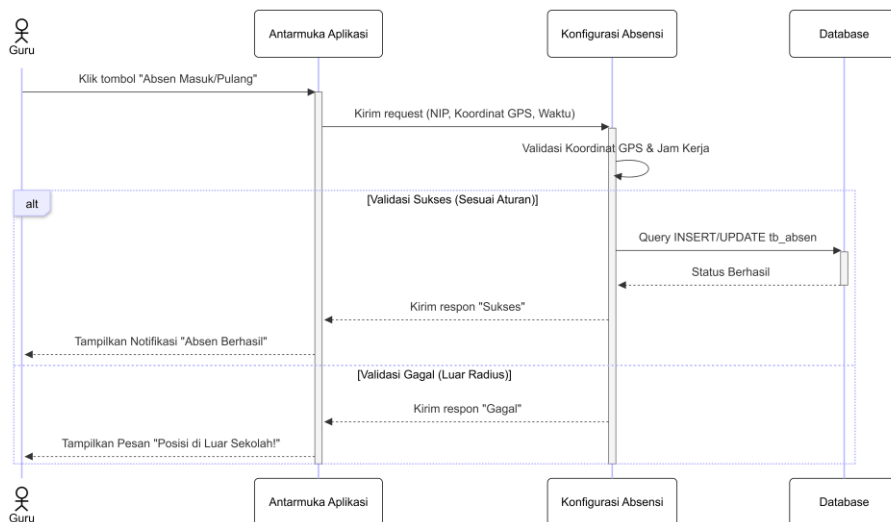
- A. Langkah 1: Mulai. Admin membuka aplikasi/sistem.
- B. Langkah 2: Login Akun Admin. Admin memasukkan kredensial khusus mereka untuk mendapatkan hak akses manajerial.
- C. Langkah 3: Pilih Menu Utama. Setelah berhasil *login*, Admin dihadapkan pada percabangan (*decision*) dengan tiga pilihan menu utama yang melayani fungsi operasional sekolah yang berbeda:
  - 1. Jalur A (Master Data): Jika Admin memilih menu ini, aktivitas yang dilakukan adalah "Tambah/Edit Data Guru & Piket". Di sini admin mengelola data kepegawaian.
  - 2. Jalur B (Penggajian): Jika Admin memilih menu ini, aktivitas yang dilakukan adalah "Hitung & Terbitkan Slip Gaji Bulanan". Ini adalah proses transaksional mengolah absen menjadi uang.
  - 3. Jalur C (Prediksi): Jika Admin memilih menu analitik ini, aktivitas yang dilakukan adalah "Lihat Prediksi Anggaran Bulan Depan".
- D. Langkah 4: Selesai Aktivitas. Setelah menyelesaikan tugas di salah satu menu tersebut, aktivitas manajerial Admin di tahap itu selesai. (Catatan: Admin tentu bisa kembali ke "Pilih Menu Utama" dalam praktiknya, namun diagram ini menggambarkan satu siklus lurus aktivitas).

- E. Langkah 5: Logout Akun. Admin keluar dari sistem untuk menjaga keamanan data sensitif sekolah.
- F. Langkah 6: Selesai. Alur sistem untuk Admin berakhir.

3. Desain Sequence Diagram

Sequence Diagram mendetailkan interaksi antar objek berdasarkan urutan waktu (*time-series*) pada satu kasus spesifik, yaitu "Proses Prediksi Anggaran". Diagram ini memperlihatkan penerapan arsitektur MVC (*Model-View-Controller*).

A. Sequence diagram Guru



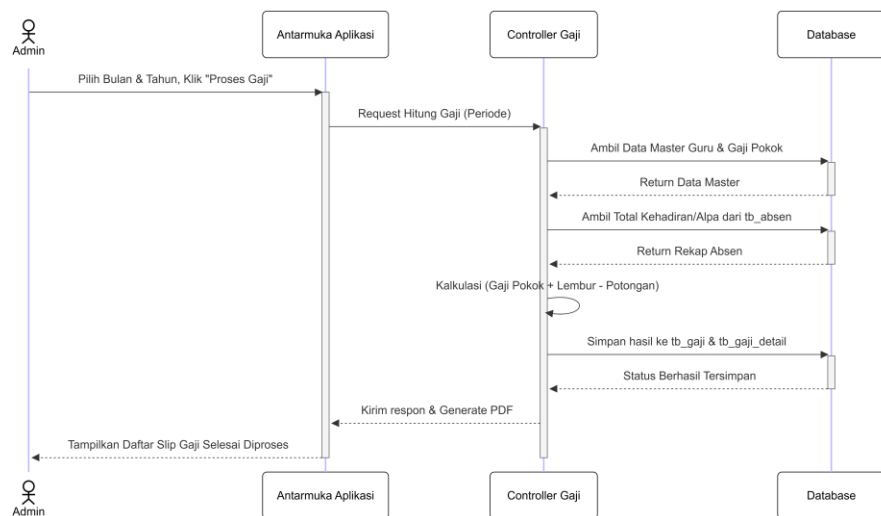
Gambar III.4.

Sequence Diagram Guru

Penjelasan Alur Absensi:

1. Guru menekan tombol absen di layar *smartphone* (Antarmuka Aplikasi).
2. Antarmuka Aplikasi (UI) menangkap titik lokasi (GPS) dan waktu saat itu, lalu mengirimkannya ke Sistem (Controller).
3. Sistem mengecek apakah lokasi Guru masuk dalam radius sekolah.
4. Jika Sukses: Sistem menyimpan data ke tabel absen di Database, lalu mengembalikan pesan sukses hingga muncul notifikasi di layar Guru.
5. Jika Gagal: Sistem menolak menyimpan data dan langsung mengembalikan pesan *error* ke layar Guru.

#### B. *Sequence* diagram Admin (Penggajian)



Gambar III.5.

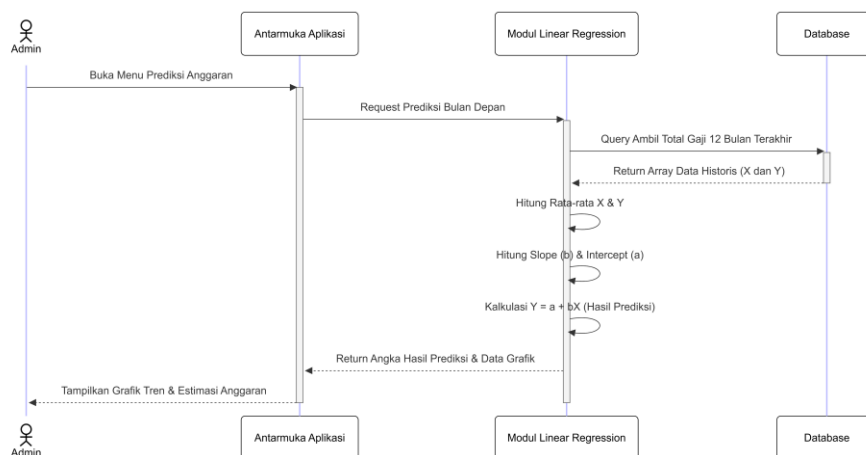
#### *Sequence* Diagram Admin (Penggajian)

Penjelasan Alur Penggajian:

1. Admin memilih periode bulan dan menekan tombol proses di Antarmuka Aplikasi.

2. Sistem secara otomatis melakukan dua kali penarikan data ke Database: pertama mengambil data standar gaji guru, kedua mengambil rekap jumlah kehadiran dan alpa pada bulan tersebut.
3. Setelah data terkumpul, Sistem menjalankan rumus matematika untuk menghitung total uang yang diterima (gaji bersih).
4. Hasil hitungan disimpan kembali secara permanen ke Database.
5. Sistem menginformasikan ke Admin bahwa proses selesai dan slip gaji sudah siap dicetak/diunduh.

### C. *Sequence* diagram Admin (Prediksi anggaran)



Gambar III.6.

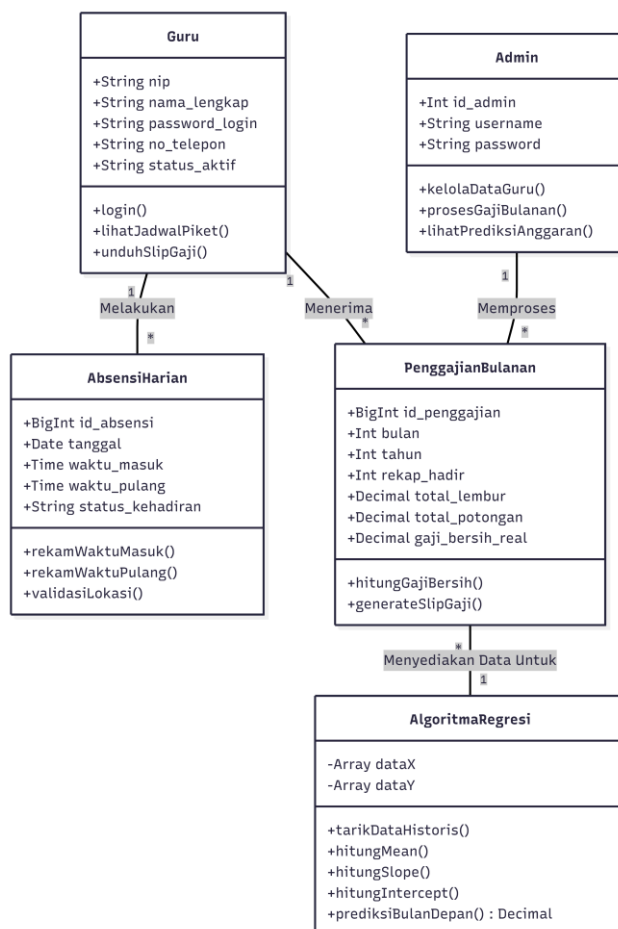
### *Sequence* Diagram Admin (Prediksi Anggaran)

#### Penjelasan Alur Prediksi:

1. Admin membuka halaman prediksi anggaran di Antarmuka Aplikasi.
2. UI memerintahkan Modul Algoritma untuk mulai bekerja.
3. Algoritma meminta Database untuk memberikan data total pengeluaran gaji selama setahun terakhir sebagai bahan bakar perhitungan.

4. Algoritma menjalankan serangkaian proses matematis (menghitung rata-rata, mencari kemiringan/ slope, dan garis potong/ intercept).
5. Setelah angka estimasi untuk bulan depan didapatkan, algoritma mengirimkan angka tersebut beserta data koordinat garis (trendline) kembali ke UI.
6. Admin melihat hasil akhirnya berupa angka nominal kebutuhan dana dan grafik visual di layar monitor.

#### 4. Desain Class Diagram



Gambar III.7.

Class Diagram

berikut adalah penjabaran alur logika sistem yang terbagi menjadi 3 proses utama:

#### 1. Alur Proses Absensi

- A. Awal Interaksi: Objek dari kelas Guru pertama kali menggunakan metode `+login()` untuk masuk ke dalam sistem.
- B. Proses Eksekusi: Setelah login, kelas Guru akan berinteraksi dengan kelas `AbsensiHarian` melalui relasi "Melakukan".
- C. Kinerja Metode: Di dalam kelas `AbsensiHarian`, sistem akan memanggil metode `+validasiLokasi()` terlebih dahulu. Jika lokasi valid, barulah metode `+rekamWaktuMasuk()` atau `+rekamWaktuPulang()` dijalankan untuk mengisi atribut-atribut data (seperti `waktu_masuk` dan tanggal).
- D. Kardinalitas: Angka 1 ke \* menunjukkan bahwa satu objek Guru secara logis akan menghasilkan dan terhubung dengan banyak sekali objek/data `AbsensiHarian` seiring berjalannya hari kerja.

#### 2. Alur Transaksional Bulanan (Proses Penggajian)

- A. Awal Interaksi: Kelas Admin memicu proses ini melalui metode `+prosesGajiBulanan()`. Admin memiliki relasi "Memproses" terhadap kelas `PenggajianBulanan`.
- B. Proses Eksekusi: Pemanggilan tersebut membuat kelas `PenggajianBulanan` bekerja. Metode `+hitungGajiBersih()` dieksekusi dengan mengolah atribut-atribut yang ada di dalamnya (mengurangi gaji standar dengan `total_potongan` dan menambahkannya dengan `total_lembur`).

- C. Output: Setelah hitungan selesai, metode `+generateSlipGaji()` dieksekusi.
- D. Kardinalitas & Penerimaan: Hasil dari kelas `PenggajianBulanan` ini kemudian dihubungkan ke kelas `Guru` melalui relasi "Menerima". Metode `+unduhSlipGaji()` pada kelas `Guru` kini bisa digunakan untuk menarik data tersebut. Relasi 1 ke \* berarti satu `Guru` akan menerima banyak data `PenggajianBulanan` (satu setiap bulannya).

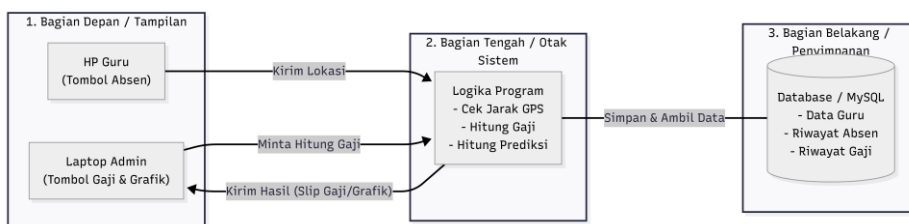
### 3. Alur Analitik Cerdas (Prediksi Anggaran)

- A. Awal Interaksi: Kelas `Admin` memicu perhitungan dengan memanggil metode `+lihatPrediksiAnggaran()`.
- B. Tarik Data: Ini akan membangunkan kelas `AlgoritmaRegresi`. Hal pertama yang dilakukan kelas `algoritma` ini adalah menjalankan metode `+tarikDataHistoris()`.
- C. Suplai Data: Melalui relasi "Menyediakan Data Untuk", objek dari sekumpulan `PenggajianBulanan` yang sudah ada sebelumnya memberikan nilai nominal gajinya untuk disimpan ke dalam atribut `private -dataX` (waktu) dan `-dataY` (total gaji) di dalam kelas `algoritma`.
- D. Proses Matematis: Setelah data terkumpul, kelas `AlgoritmaRegresi` menjalankan operasi matematikanya secara berurutan: `+hitungMean()`, lalu `+hitungSlope()`, dan `+hitungIntercept()`.
- E. Output Final: Terakhir, metode `+prediksiBulanDepan()`: `Decimal` mengembalikan hasil akhir berupa angka desimal (nominal uang prediksi) untuk ditampilkan kembali ke antarmuka `Admin`.

### 3.2.2. Desain Arsitektur Sistem

Desain Arsitektur Sistem adalah pemodelan konseptual yang mendefinisikan struktur, perilaku, dan interaksi antara berbagai komponen di dalam aplikasi Anda. Ibarat membangun gedung, ini adalah peta yang menunjukkan di mana letak fondasi, saluran air, dan jalur listrik agar semuanya berfungsi selaras.

Untuk "Sistem Penggajian & Prediksi Anggaran Berbasis Machine Learning di MTs Irsyadul Athfal", arsitektur yang paling ideal dan umum digunakan adalah Arsitektur Client-Server 3-Tier (Tiga Lapis). Pemilihan arsitektur ini sangat cocok diterapkan menggunakan teknologi pengembangan web standard industri seperti kombinasi PHP untuk logika peladen (server) dan MySQL untuk basis datanya.



Gambar III.8.

#### Desain Arsitektur Sistem

Berikut Penjelasan Singkat alur desain arsitektur sistem:

#### 1. Bagian Depan / Tampilan

Ini adalah aplikasi yang dilihat dan disentuh langsung oleh pengguna.

- A. Guru menggunakan HP mereka untuk menekan tombol "Absen". Sistem di HP akan menangkap lokasi (GPS) mereka.
- B. Admin/Bendahara menggunakan Laptop sekolah karena layarnya lebih besar untuk melihat tabel laporan gaji dan grafik prediksi.

#### 2. Bagian Tengah / Otak Sistem

Di sinilah aplikasi "berpikir" dan bekerja. Pengguna tidak melihat bagian ini.

- A. Saat Guru menekan tombol absen, "otak" ini akan mengecek: "*Apakah lokasi guru ini sudah sampai di MTs Irsyadul Athfal?*"
- B. Saat Admin menekan tombol proses gaji, "otak" ini akan mengambil kalkulator otomatis untuk menambah gaji pokok dengan lembur, lalu memotong denda absen.
- C. Di sini juga rumus *Linear Regression* bekerja diam-diam untuk menebak anggaran bulan depan.

### 3. Bagian Belakang / Penyimpanan Data

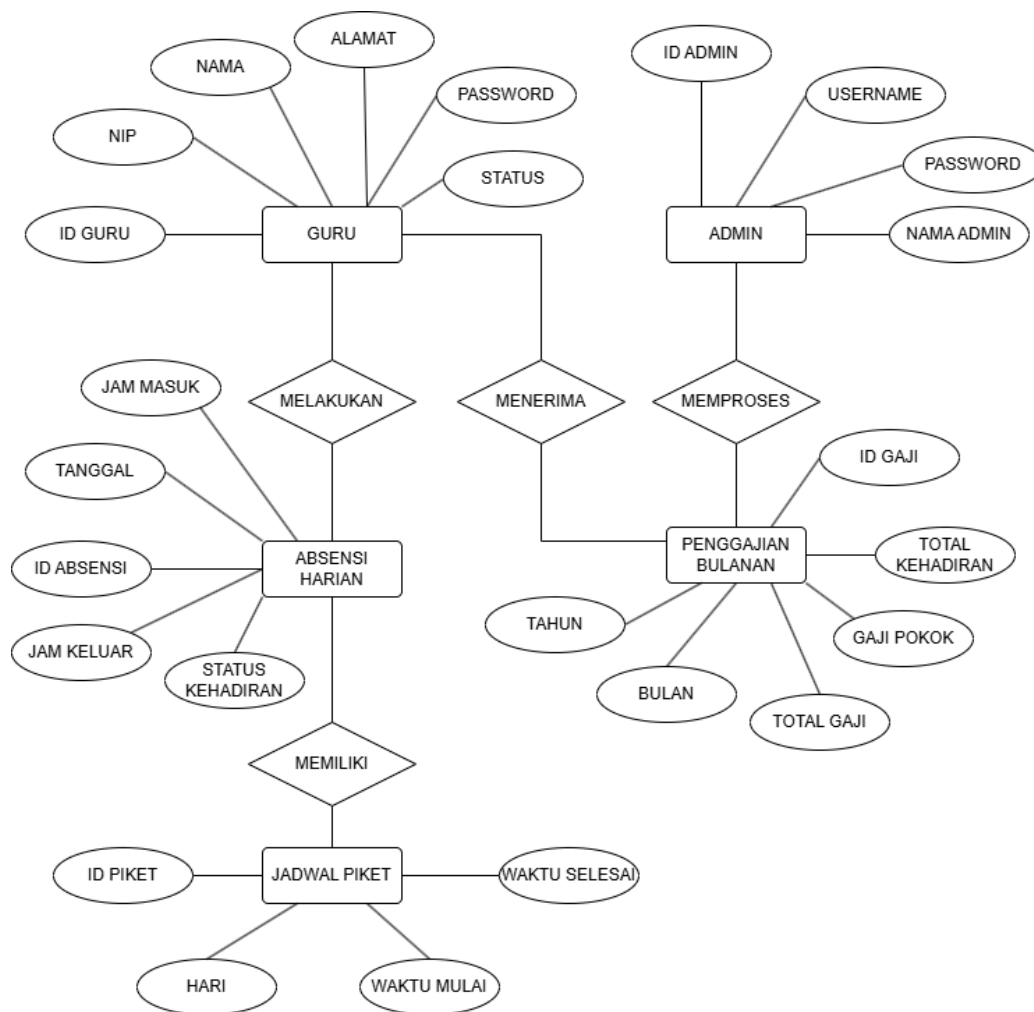
Ini adalah lemari brankas tempat semua data sekolah disimpan dengan aman (Database/MySQL).

- A. "Otak sistem" mengambil daftar nama guru dari sini.
- B. Setelah gaji selesai dihitung, "Otak sistem" menyimpan hasilnya kembali ke lemari ini.
- C. Ketika mau melakukan prediksi anggaran, "Otak sistem" akan membuka lemari ini untuk melihat catatan pengeluaran gaji selama setahun terakhir.

Intinya: Guru/Admin meminta (di Depan)  $\rightarrow$  Sistem menghitung (di T

#### 3.2.3. Desain Permodelan Basis Data

*Entity Relationship Diagram* (ERD) ini menggambarkan struktur penyimpanan data dan hubungan antar tabel dalam basis data. Model ini menggunakan pola Header-Detail pada tabel gaji untuk fleksibilitas komponen pembayaran.



Gambar III.9

## Entity Relationship Diagram (ERD)

## 1. Identifikasi Entitas

Berdasarkan analisis kebutuhan sistem, diperoleh lima entitas utama yang membentuk struktur basis data, yaitu:

- A. Guru
- B. Admin
- C. Absensi\_Harian
- D. Jadwal\_Piket
- E. Penggajian\_Bulanan

Setiap entitas merepresentasikan objek nyata dalam sistem yang

memiliki data tersendiri.

## 2. Deskripsi Entitas dan Atribut

### A. Entitas Guru

Entitas Guru menyimpan data identitas tenaga pendidik.

1. Primary Key: id\_guru
2. Atribut: nip, nama, alamat, no\_hp, email, password, status.

Primary key id\_guru digunakan sebagai identitas unik setiap guru dan menjadi referensi pada entitas lain.

### B. Entitas Admin

Entitas Admin menyimpan data pengguna dengan hak akses pengelolaan sistem.

- A. Primary Key: id\_admin
- B. Atribut: username, password, nama\_admin.

Entitas ini berdiri sendiri dan berfungsi sebagai pengelola sistem.

### C. Entitas Absensi\_Harian

Entitas Absensi\_Harian menyimpan data kehadiran guru.

- A. Primary Key: id\_absensi
- B. Foreign Key: id\_guru
- C. Atribut lainnya: tanggal, jam\_masuk, jam\_keluar, status\_kehadiran

Foreign key id\_guru menunjukkan bahwa setiap data absensi dimiliki oleh satu guru.

### D. Entitas Jadwal\_Piket

Entitas Jadwal\_Piket menyimpan informasi jadwal piket guru.

- A. Primary Key: id\_piket

B. Foreign Key: id\_guru

C. Atribut: hari, waktu\_mulai, waktu\_selesai.

Setiap jadwal piket terkait dengan satu guru.

#### E. Entitas Penggajian\_Bulanan

Entitas Penggajian\_Bulanan menyimpan data hasil perhitungan gaji guru setiap bulan.

A. Primary Key: id\_gaji

B. Foreign Key: id\_guru

C. Atribut: bulan, tahun, total\_kehadiran, gaji\_pokok, potongan, total\_gaji

Entitas ini menggunakan data absensi sebagai dasar perhitungan gaji.

### 3. Relasi Antar Entitas

Relasi antar entitas dalam ERD ditentukan berdasarkan kebutuhan sistem dan aturan bisnis.

A. Relasi Guru – Absensi\_Harian

Kardinalitas:

1. Guru : N Absensi\_Harian

Artinya satu guru dapat memiliki banyak data absensi, tetapi satu data absensi hanya dimiliki oleh satu guru.

B. Relasi Guru – Jadwal\_Piket

Kardinalitas:

1. Guru : N Jadwal\_Piket

Artinya satu guru dapat memiliki lebih dari satu jadwal piket.

C. Relasi Guru – Penggajian\_Bulanan

Kardinalitas:

## 1. Guru : N Penggajian\_Bulanan

Artinya satu guru memiliki banyak data penggajian (setiap periode).

## D. Relasi Admin – Penggajian\_Bulanan

Kardinalitas:

## 1. Admin : N Penggajian\_Bulanan

Satu admin dapat memproses banyak data penggajian.

## 1. Spesifikasi Tabel dan Penjelasan Relasi

- a) Nama Tabel : Pengguna  
 Akronim : tb\_pengguna  
 File : Master  
 Format : Dynamic  
 Akses file : Random Access File  
 Primary Key : nip  
 Size : 32 KiB

Tabel III.3.  
 Spesifikasi Tabel Pengguna

No.	Elemen Tabel	Akronim	Tipe Data	Length	keterangan
1	NIP	nip	Varchar	20	Primary Key
2	Nama Pengguna	nama_pengguna	Varchar	100	-
3	Gaji Pokok	gaji_pokok	Decimal	12,2	-
4	Jabatan	jabatan	Varchar	50	-
5	Tanggal Mulai Mengajar	tanggal_mulai_mengajar	Date	-	-
6	Status Guru	status_guru	Varchar	50	-
7	Username	username	Varchar	50	Foreign Key
8	Password	password	Varchar	100	-
9	Status	status	Enum	-	-
10	Keterangan	keterangan	text	-	-

- b) Nama Tabel : Absen  
 Akronim : tb\_absen  
 File : Master  
 Format : Dynamic  
 Akses file : Random Access File  
 Primary Key : ida

Size : 16 KiB

Tabel III.4.  
Spesifikasi Tabel Absen

No.	Elemen Tabel	Akronim	Tipe Data	Length	keterangan
1	ID Absen	ida	Varchar	20	<i>Primary Key</i>
2	NIP	nip	Varchar	20	-
3	Tanggal Masuk	tanggal_masuk	Date	-	-
4	Jam Masuk	jam_masuk	Time	-	-
5	Status Masuk	status_masuk	Eunm	-	-
6	Tanggal Pulang	tanggal_pulang	Time	-	-
7	Jam Pulang	jam_pulang	Date	-	-
8	Status Pulang	status_pulang	Enum	-	-
9	Status	status	Enum	-	-
10	Keterangan	keterangan	Text	-	-
11	Koordinat 1	Koordinat1	Varchar	50	-
12	Lokasi 1	lokasi1	Varchar	100	-
13	Jarak 1	jarak1	Varchar	30	-
14	Koordinat 2	koordinat2	Varchar	30	-
15	Lokasi 2	lokasi2	Varchar	100	-
16	Jarak 2	jarak2	Varchar	30	-
17	Hasil	hasil	Varchar	30	-

c) Nama Gaji : Gaji

Akronim : tb\_gaji  
 File : Master  
 Format : Dynamic  
 Akses file : Random Access File  
 Primary Key : id\_gaji  
 Size : 16 KiB

Tabel III.5.  
Spesifikasi Tabel Gaji

No.	Elemen Tabel	Akronim	Tipe Data	Length	keterangan
1	ID Gaji	id_gaji	Int	-	<i>Primary Key</i>
2	Periode	periode	Varchar	20	-
3	Tertanda 1	tertanda1	Varchar	20	-
4	Tertanda 2	tertanda2	Varchar	20	-
5	Status	status	Eunm	-	-
6	Keterangan	keterangan	Text	-	-

d) Nama : Gaji Detail

Akronim : tb\_gaji\_detail  
 File : Master  
 Format : Dynamic  
 Akses file : Random Access File

*Primary Key* : idd  
 Size : 80 KiB

Tabel III.6.  
 Spesifikasi Tabel Detail Gaji

No.	Elemen Tabel	Akronim	Tipe Data	Length	keterangan
1	ID Detail	idd	Int	-	<i>Primary Key</i>
2	ID Gaji	id_gaji	Varchar	20	-
3	NIP	nip	Varchar	20	-
4	Masa Kerja	masa_kerja	Decimal	4,1	-
5	Bonus MasaKerja	bonus_masa_kerja	Decimal	12,2	-
6	JumlahJamMengajar	jumlah_jam_mengajar	Int	-	-
7	Jumlah Honor	jumlah_honor	Decimal	12,2	-
8	Jumlah Hadir	jumlah_hadir	Int	-	-
9	Transport	transport	decimal	12,2	-
10	Total	total	Decimal	12,2	-
11	Catatan	catatan	Text	-	-
12	Gaji Pokok	gaji_pokok	Int	-	-

e) Nama Tabel : Piket  
 Akronim : tb\_piket  
 File : Master  
 Format : Dynamic  
 Akses file : Random Access File  
*Primary Key* : idp  
 Size : 16 KiB

Tabel III.7.  
 Spesifikasi Tabel Piket

No.	Elemen Tabel	Akronim	Tipe Data	Size	keterangan
1	ID Piket	idp	Int	-	<i>Primary Key</i>
2	NIP	nip	Varchar	20	-
3	Hari	hari	Enum	-	-
4	Jam Mulai	jam_mulai	Time	-	-
5	Jam Selesai	jam_selesai	Time	-	-
6	Total Jam	total_jam	Int	-	-
7	Keterangan	keterangan	Text	-	-

f) Nama File : Config

Akronim : tb\_config  
 File : Master  
 Format : Dynamic  
 Akses file : Random Access File  
 Primary Key : idc  
 Size : 16 KiB

Tabel III.8.  
Spesifikasi Tabel Config

No.	Elemen Tabel	Akronim	Tipe Data	Size	keterangan
1	ID Config	idc	Int	-	<i>Primary Key</i>
2	Uraian	uraian	Varchar	100	-
3	Biaya	biaya	Decimal	12,2	-
4	Keterangan	keterangan	Text	-	-

### 3.3. Tahap Implementasi / Pengkodean (*Implementation / Coding*)

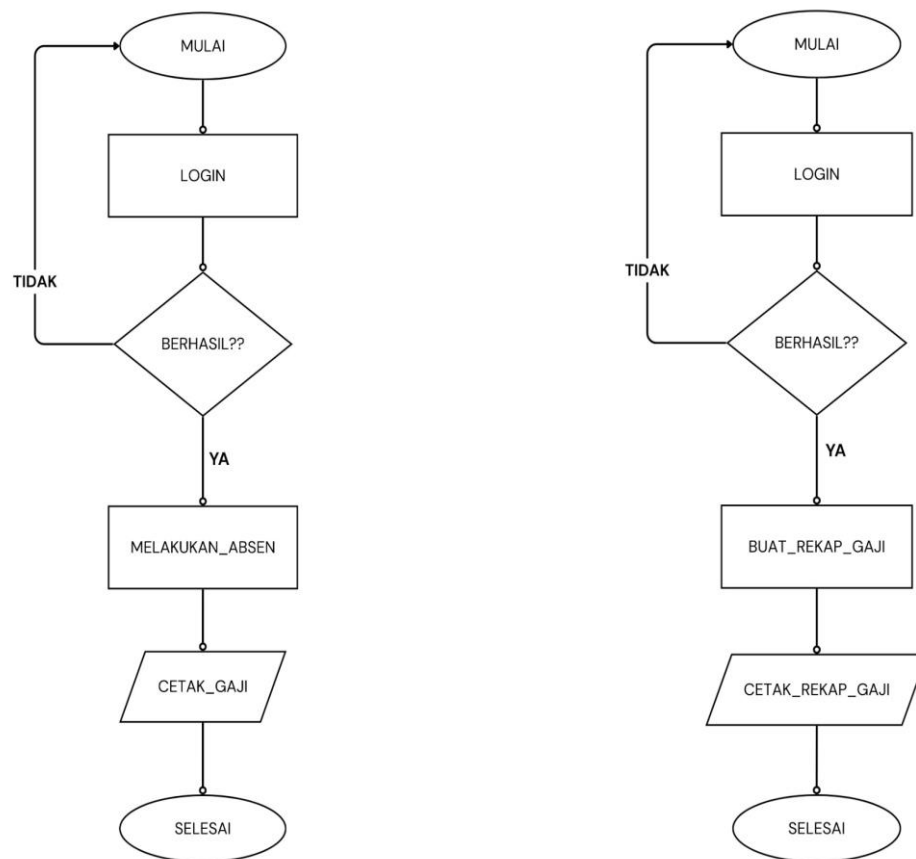
Tahap konstruksi merupakan fase implementasi kode program berdasarkan rancangan yang telah dibuat sebelumnya. Pada tahap ini, seluruh komponen sistem mulai dibangun, diuji, dan digabungkan menjadi aplikasi penggajian online beserta fitur prediksi anggaran berbasis machine learning.

Tahap konstruksi juga memastikan bahwa sistem dapat berjalan dengan baik sesuai kebutuhan fungsional maupun non-fungsional.

#### 1. Teknologi Aplikasi

##### A. Algoritma yang digunakan

Dalam penelitian ini, metode pengembangan perangkat lunak yang digunakan adalah System Development Life Cycle (SDLC) dengan model Waterfall. Model Waterfall adalah pendekatan klasik yang bersifat sistematis dan berurutan dalam membangun perangkat lunak.



Gambar III.7.

## Flowchart Login Guru dan Admin

Menurut Roger S. Pressman, model ini mengusulkan sebuah pendekatan kepada pengembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan.

Berikut adalah penjelasan dari setiap alur pada flowchart :

1. Tahap Analisis Kebutuhan (Requirement Analysis)

Ini adalah fase fondasi di mana pengembang harus memahami sepenuhnya apa yang dibutuhkan oleh pihak sekolah sebelum mulai bekerja. Tujuannya Adalah

untuk mengumpulkan, menganalisis, dan mendokumentasikan semua aturan bisnis dan kebutuhan fungsional aplikasi. Aktivitas detail dalam konteks aplikasi:

- a. Wawancara Mendalam: Berdiskusi dengan pemangku kepentingan utama di sekolah, terutama Bendahara dan Kepala Sekolah.
- b. Definisi Aturan Penggajian: Mencatat secara rinci komponen gaji (gaji pokok per jabatan, tunjangan tetap), cara menghitung lembur (tarif per jam), dan aturan potongan (denda keterlambatan atau ketidakhadiran).
- c. Definisi Aturan Absensi: Menentukan mekanisme absensi yang diinginkan (misalnya, apakah menggunakan validasi GPS/radius lokasi, jam berapa batas waktu masuk dan pulang).
- d. Kebutuhan Prediksi: Menentukan data historis apa yang diperlukan agar fitur prediksi anggaran (menggunakan regresi linear) dapat berjalan.

Hasil (Output): Catatan Aturan Gaji & Absensi. Dokumen ini menjadi acuan tunggal yang disepakati bersama.

## 2. Tahap Desain Sistem (System Design)

Pada tahap ini, kebutuhan yang telah dikumpulkan diterjemahkan menjadi rancangan teknis atau "cetak biru" aplikasi. Tujuannya adalah merancang struktur data dan antarmuka pengguna agar sesuai dengan kebutuhan yang telah didefinisikan.

Aktivitas detail dalam konteks aplikasi:

- a. Perancangan Database (ERD): Membuat desain Entity Relationship Diagram (seperti yang telah dibahas sebelumnya) yang mencakup tabel GURU, ABSENSI\_HARIAN, PENGGAJIAN\_BULANAN, dan JADWAL\_PIKET. Ini memastikan semua data gaji dan absensi dapat disimpan dengan terstruktur.
- b. Desain Antarmuka (UI/UX): Membuat *mockup* atau rancangan tampilan layar. Contohnya: Merancang tampilan *dashboard* admin untuk proses hitung gaji, dan

merancang tampilan aplikasi *mobile* bagi guru untuk menekan tombol absen.

- c. Desain Algoritma: Merancang alur logika (flowchart/pseudocode) untuk perhitungan gaji bulanan dan algoritma regresi linear untuk fitur prediksi.

Hasil (Output): Gambar Rancangan Database & Tampilan. Rancangan ini siap untuk diserahkan kepada programmer.

### 3. Tahap Penulisan Kode / Coding (Implementation)

Ini adalah tahap konstruksi di mana rancangan diubah menjadi aplikasi yang sebenarnya oleh programmer. Tujuan: Membangun sistem perangkat lunak berdasarkan dokumen desain. Aktivitas detail dalam konteks aplikasi:

- a. Pembuatan Database: Mengimplementasikan rancangan ERD ke dalam sistem manajemen database (misalnya MySQL).
- b. Coding Backend: Menulis kode program (misalnya menggunakan PHP atau Python) untuk menangani logika bisnis, membuat fungsi otomatisasi perhitungan gaji berdasarkan kehadiran dan fungsi matematika untuk prediksi anggaran.
- c. Coding Frontend: Membuat tampilan antarmuka (Web atau Mobile App) agar pengguna (Admin dan Guru) dapat berinteraksi dengan sistem, seperti mengisi form absensi atau melihat slip gaji.

Hasil (Output): Aplikasi Jadi / Belum Diuji. Sistem sudah berbentuk perangkat lunak tetapi mungkin masih mengandung *bug* (kesalahan).

### 4. Tahap Pengujian / Testing (Verification)

Sebelum digunakan secara nyata, aplikasi wajib diperiksa secara menyeluruh untuk memastikan tidak ada kesalahan dan sudah sesuai dengan aturan sekolah. Tujuannya adalah Menemukan dan memperbaiki kesalahan (*bug*) serta memverifikasi bahwa sistem berfungsi sesuai spesifikasi kebutuhan.

Aktivitas detail dalam konteks aplikasi:

- a. Pengujian Fungsional: Menguji apakah fitur absensi GPS berfungsi memblokir lokasi yang salah. Menguji apakah fitur "Proses Gaji" menghasilkan angka yang sama persis dengan perhitungan manual Bendahara.
- b. Pengujian Fitur Prediksi: Memasukkan data historis dummy dan memastikan algoritma regresi linear menghasilkan prediksi angka yang logis.
- c. User Acceptance Test (UAT): Meminta seluruh guru, bendahara, tata usaha dan kepala sekolah untuk mencoba aplikasi secara langsung dalam skenario nyata untuk memastikan aplikasi mudah digunakan dan memenuhi harapan.

Hasil (Output): Aplikasi Lolos Uji & Siap Pakai. Aplikasi yang stabil dan siap untuk diinstalasi.

#### 5. Tahap Penerapan & Pemeliharaan (Deployment & Maintenance)

Ini adalah tahap akhir di mana aplikasi mulai digunakan untuk operasional sehari-hari di sekolah dan dirawat keberlangsungannya. Tujuannya mengoperasikan sistem di lingkungan nyata dan memberikan dukungan teknis berkelanjutan. Aktivitas

Detail dalam Konteks Aplikasi:

- a. Penerapan (Deployment): Menginstal aplikasi di server sekolah atau hosting cloud. Melakukan migrasi data awal (input data guru yang ada). Memberikan pelatihan (training) kepada Admin dan Guru tentang cara penggunaan aplikasi.
- b. Pemeliharaan (Maintenance): Memonitor kinerja aplikasi. Memperbaiki *bug* kecil yang mungkin baru ditemukan setelah penggunaan rutin. Melakukan *update* pada sistem jika ada perubahan kebijakan sekolah di masa depan (misalnya, kenaikan tarif gaji pokok tahun depan).

Kondisi Akhir: Sistem Berjalan di Sekolah dan digunakan untuk kegiatan operasional sehari-hari.

#### B. *Tools* Pendukung pengembangan Sistem

Pada proses pembangunan sistem, digunakan beberapa teknologi untuk mendukung pengembangan aplikasi berbasis web, meliputi:

#### 1. Bahasa Pemrograman

##### a. PHP

Digunakan sebagai server-side scripting untuk membangun logika bisnis aplikasi, memproses data, dan berkomunikasi dengan database.

##### b. JavaScript

Digunakan pada sisi client untuk validasi form, manipulasi DOM, dan menampilkan data secara dinamis.

#### 2. Database

##### a. MySQL

Digunakan sebagai sistem manajemen basis data (DBMS) untuk menyimpan data karyawan, absensi, jabatan, komponen gaji, perhitungan gaji, serta data historis untuk prediksi.

#### 3. Web Server

##### a. Laragon

Digunakan sebagai web server lokal untuk menjalankan aplikasi selama proses pengembangan.

#### 4. Antarmuka / UI

##### a. HTML5 & CSS3

Untuk membangun struktur tampilan.

##### b. Bootstrap

Untuk mempercepat pembuatan tampilan yang responsif dan konsisten.

#### 5. Teknologi Machine Learning

##### a. Regresi Linear Sederhana

Diimplementasikan menggunakan perhitungan manual melalui PHP (tanpa library ML berat), karena dataset relatif kecil.

## 6. Tools Pengembangan

- a. Visual Studio Code
- b. phpMyAdmin
- c. Browser (Chrome/Edge)

## C. Tahap Pengujian Aplikasi

Pengujian dilakukan untuk memastikan sistem penggajian dan prediksi anggaran pada MTs Irsyadul Athfal berjalan sesuai spesifikasi (fungsional), tahan terhadap beban (performance), dan aman dari serangan (security). Tahap pengujian merupakan fase verifikasi dan validasi untuk memastikan seluruh komponen perangkat lunak bekerja secara harmonis.

### 1. Pengujian Performance

Pada pengujian performance ini penulis menggunakan *software Apache Jmeter* yang berfungsi untuk pengujian kinerja (performance testing) dan pengujian beban (load testing) pada aplikasi berbasis web, database, FTP, dan layanan lainnya.

#### a. Load Testing

Load testing adalah proses pengujian performa website dengan memberikan beban kerja (trafik/pengguna) yang sesuai dengan ekspektasi normal hingga batas puncak (peak) yang diperkirakan.

Tujuannya adalah memastikan website tetap responsif dan stabil saat diakses oleh jumlah pengguna yang diharapkan dan untuk mengidentifikasi *bottleneck* (hambatan) seperti database yang lambat atau kode yang tidak efisien sebelum website *live*.

#### 1) Login

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only:  Errors  Successes  Configure

Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Time...
31	16:55:21.026	Load Testing 1...	Login	3693	✓	13806	133	3683	110
32	16:55:21.590	Load Testing 1...	Login	2155	✓	12006	122	2147	81
33	16:55:21.119	Load Testing 1...	Login	3693	✓	13806	133	3676	76
34	16:55:21.654	Load Testing 1...	Login	3182	✓	13806	133	3176	78
35	16:55:20.838	Load Testing 1-7	Login	4267	✓	13806	133	4260	76
36	16:55:20.723	Load Testing 1-1	Login	4395	✓	13806	133	4391	167
37	16:55:20.963	Load Testing 1...	Login	4219	✓	13806	133	4213	99
38	16:55:21.553	Load Testing 1...	Login	3692	✓	13806	133	3685	88
39	16:55:21.632	Load Testing 1...	Login	3845	✓	13806	133	3837	77
40	16:55:21.673	Load Testing 1...	Login	3634	✓	13806	133	3627	71
41	16:55:21.692	Load Testing 1...	Login	3831	✓	13806	133	3821	78
42	16:55:20.807	Load Testing 1-5	Login	4759	✓	13806	133	4752	85
43	16:55:21.412	Load Testing 1...	Login	4189	✓	13806	133	4179	89
44	16:55:20.932	Load Testing 1...	Login	4688	✓	13806	133	4681	83
45	16:55:21.072	Load Testing 1...	Login	4640	✓	13806	133	4633	76
46	16:55:21.333	Load Testing 1...	Login	4971	✓	13806	133	4966	79
47	16:55:21.314	Load Testing 1...	Login	4094	✓	13806	133	4088	82
48	16:55:21.394	Load Testing 1...	Login	4652	✓	13806	133	4647	79
49	16:55:21.434	Load Testing 1...	Login	4649	✓	13806	133	4643	72
50	16:55:21.372	Load Testing 1...	Login	5425	✓	13806	133	5419	79

Scroll automatically?  Child samples?  No of Samples: 50 Label Samples: 5425 Bytes: 13337 Received: 1333

Gambar III.8.  
Hasil *respons load testing* halaman login

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only:  Errors  Successes  Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
Login	50	3153	705	5425	1123.50	0.00%	8.2/sec	110.98	1.07	13806.0
TOTAL	50	3153	705	5425	1123.50	0.00%	8.2/sec	110.98	1.07	13806.0

Include group name in label?  Save Table Data  Save Table Header

Gambar III.9.  
*Summary report load testing* halaman login

Hasil Identifikasi load testing pada halaman login:

- Samples (50): JMeter telah mengirimkan total 50 permintaan (request) ke server untuk proses Login.
- Average (3153 ms): Rata-rata waktu yang dibutuhkan server untuk memproses login adalah 3.153 milidetik (atau sekitar 3,15 detik).
- Min (705 ms) & Max (5425 ms): Pengguna yang paling cepat bisa login dalam waktu 0,7 detik (cepat) dan pengguna yang paling lambat harus menunggu hingga 5,4 detik (cukup lambat).
- Std. Dev. (1123.50): Standar Deviasi menunjukkan variasi data. Angka

1.123 ms ini cukup tinggi.

- e) Error % (0.00%): hasil cukup baik. Dari 50 percobaan, tidak ada yang gagal (failed). Semua permintaan berhasil diproses (HTTP Status 200 OK).
- f) Throughput (8.2/sec): Server mampu menangani sekitar 8,2 transaksi login per detik pada beban kerja ini.

## 2) Dashboard/Halaman Absensi

Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Time...
31	16:58:30.528	Load Testing 1...	Dashboard_ab...	685	✓	15662	399	77	37
32	16:58:31.042	Load Testing 1...	Dashboard_ab...	177	✓	15662	399	79	40
33	16:58:31.060	Load Testing 1...	Dashboard_ab...	197	✓	15662	399	81	40
34	16:58:31.082	Load Testing 1...	Dashboard_ab...	181	✓	15662	399	71	35
35	16:58:30.990	Load Testing 1...	Dashboard_ab...	693	✓	15662	399	86	43
36	16:58:30.704	Load Testing 1...	Dashboard_ab...	673	✓	15662	399	74	37
37	16:58:30.733	Load Testing 1...	Dashboard_ab...	665	✓	15662	399	66	32
38	16:58:30.709	Load Testing 1...	Dashboard_ab...	689	✓	15662	399	79	38
39	16:58:30.753	Load Testing 1...	Dashboard_ab...	697	✓	15662	399	80	39
40	16:58:31.304	Load Testing 1...	Dashboard_ab...	175	✓	15662	399	78	38
41	16:58:31.304	Load Testing 1...	Dashboard_ab...	184	✓	15662	399	78	38
42	16:58:31.320	Load Testing 1...	Dashboard_ab...	193	✓	15662	399	84	47
43	16:58:31.371	Load Testing 1...	Dashboard_ab...	171	✓	15662	399	79	40
44	16:58:30.950	Load Testing 1...	Dashboard_ab...	693	✓	15662	399	90	36
45	16:58:31.617	Load Testing 1...	Dashboard_ab...	181	✓	15662	399	74	38
46	16:58:30.651	Load Testing 1...	Dashboard_ab...	1186	✓	15662	399	72	38
47	16:58:31.693	Load Testing 1...	Dashboard_ab...	176	✓	15662	399	77	39
48	16:58:31.208	Load Testing 1...	Dashboard_ab...	695	✓	15662	399	93	49
49	16:58:31.728	Load Testing 1...	Dashboard_ab...	162	✓	15662	399	85	31
50	16:58:30.062	Load Testing 1-5	Dashboard_ab...	2192	✓	15662	399	68	31

Gambar III.10.  
Hasil *respon* load testing halaman absensi

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Dashboard_a...	50	375	162	2192	374.43	0.00%	22.0/sec	336.47	8.57	15662.8
TOTAL	50	375	162	2192	374.43	0.00%	22.0/sec	336.47	8.57	15662.8

Gambar III.11.  
*Summary report* load testing halaman absensi

Hasil Identifikasi load testing :

Secara umum, performa fitur ini sangat baik dan responsif.

- a) Average (375 ms): Rata-rata waktu muat halaman ini hanya 0,37 detik.

- b) Min (162 ms) & Max (2192 ms):Waktu tercepat sangat impresif (0,16 detik) dan waktu terlama mencapai 2,1 detik.
- c) Throughput (22.0/sec):Server mampu menangani 22 permintaan per detik untuk halaman ini.
- d) Error % (0.00%):Sama seperti tes sebelumnya, tidak ada kegagalan. Server stabil dalam menjawab permintaan.

### 3) Halaman Penggajian

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only  Errors  Successes  Configure

Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Time...
31	21:52:25.488	Load_testing 1...	Halaman_pen...	784	✓	15669	409	72	36
32	21:52:25.790	Load_testing 1...	Halaman_pen...	510	✓	15669	409	69	34
33	21:52:25.997	Load_testing 1...	Halaman_pen...	304	✓	15669	409	70	32
34	21:52:26.378	Load_testing 1...	Halaman_pen...	546	✓	15669	409	69	34
35	21:52:26.028	Load_testing 1...	Halaman_pen...	227	✓	15669	409	78	37
36	21:52:26.075	Load_testing 1...	Halaman_pen...	287	✓	15669	409	84	39
37	21:52:25.853	Load_testing 1...	Halaman_pen...	509	✓	15669	409	69	34
38	21:52:26.107	Load_testing 1...	Halaman_pen...	286	✓	15669	409	111	48
39	21:52:26.091	Load_testing 1...	Halaman_pen...	302	✓	15669	409	79	37
40	21:52:25.885	Load_testing 1...	Halaman_pen...	549	✓	15669	409	71	36
41	21:52:25.678	Load_testing 1...	Halaman_pen...	788	✓	15669	409	76	39
42	21:52:26.040	Load_testing 1...	Halaman_pen...	621	✓	15669	409	77	37
43	21:52:25.455	Load_testing 1...	Halaman_pen...	1289	✓	15669	409	76	38
44	21:52:25.964	Load_testing 1...	Halaman_pen...	730	✓	15669	409	66	32
45	21:52:25.151	Load_testing 1-1	Halaman_pen...	1669	✓	15669	409	159	115
46	21:52:25.710	Load_testing 1...	Halaman_pen...	1111	✓	15669	409	74	36
47	21:52:26.012	Load_testing 1...	Halaman_pen...	864	✓	15669	409	90	39
48	21:52:25.758	Load_testing 1...	Halaman_pen...	1261	✓	15669	409	1077	1037
49	21:52:25.536	Load_testing 1...	Halaman_pen...	1542	✓	15669	409	318	33
50	21:52:26.123	Load_testing 1...	Halaman_pen...	1505	✓	15669	409	1328	1039

Scroll automatically?  Child samples? No of Samples: 50 Latest Sample: 1505 Average: 539 Deviation: 975

Gambar III.12.  
Hasil respons load testing halaman penggajian

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only  Errors  Successes  Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
Halaman_pe...	50	539	237	1669	375.99	0.00%	20.2/sec	308.88	8.06	15669.1
TOTAL	50	539	237	1669	375.99	0.00%	20.2/sec	308.88	8.06	15669.1

Include group name in label?  Save Table Data  Save Table Header

Gambar III.13.  
Summary report load testing halaman penggajian

Hasil Identifikasi load testing :

- a) Average(539 ms / 0,5 detik): Rata-rata waktu muat halaman ini hanya 539 ms.
- b) Min (237 ms) & Max (1669 ms):Waktu tercepat sangat impresif (0,23 detik)

dan waktu terlalu lama mencapai 1,6 detik.

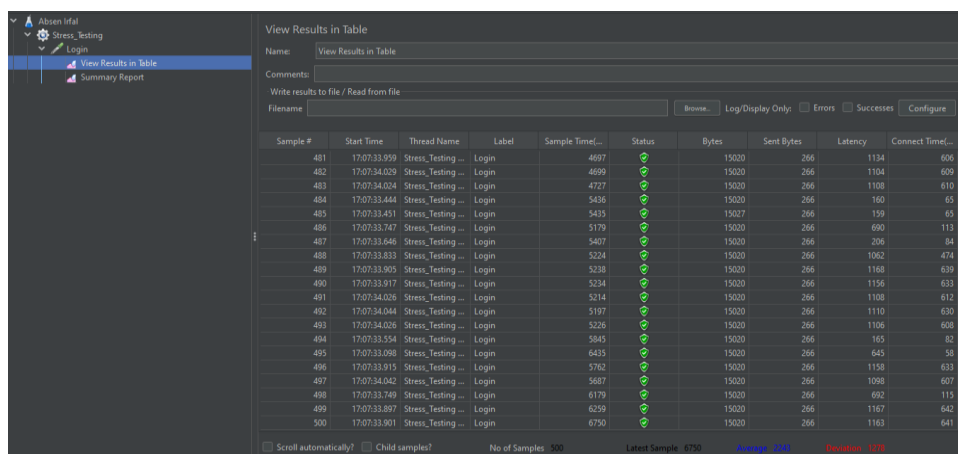
c) Throughput (20.2/sec): Kapasitas pemrosesan berada di angka 20,2 request per detik.

d) Error % (0.00%): Sama seperti tes sebelumnya, tidak ada kegagalan. Server stabil dalam menjawab permintaan.

## b. Stress Testing

Stress Testing (Pengujian Tekanan) adalah jenis pengujian performa di mana sistem (website/aplikasi) diberikan beban lalu lintas yang ekstrem, di luar batas kapasitas normal, hingga sistem tersebut akhirnya mencapai titik kerusakan (*breaking point*).

### 1) Login



Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Time...
481	17:07:33.999	Stress_Testing ...	Login	4897	Success	15020	266	1134	686
482	17:07:34.020	Stress_Testing ...	Login	4899	Success	15020	266	1104	609
483	17:07:34.024	Stress_Testing ...	Login	4727	Success	15020	266	1108	610
484	17:07:33.444	Stress_Testing ...	Login	5436	Success	15020	266	160	65
485	17:07:33.451	Stress_Testing ...	Login	5435	Success	15027	266	159	65
486	17:07:33.747	Stress_Testing ...	Login	5179	Success	15020	266	690	113
487	17:07:33.646	Stress_Testing ...	Login	5407	Success	15020	266	206	84
488	17:07:33.833	Stress_Testing ...	Login	5224	Success	15020	266	1082	474
489	17:07:33.895	Stress_Testing ...	Login	5238	Success	15020	266	1163	639
490	17:07:33.917	Stress_Testing ...	Login	5234	Success	15020	266	1156	623
491	17:07:34.026	Stress_Testing ...	Login	5214	Success	15020	266	1108	612
492	17:07:34.044	Stress_Testing ...	Login	5197	Success	15020	266	1110	630
493	17:07:34.026	Stress_Testing ...	Login	5226	Success	15020	266	1106	608
494	17:07:33.554	Stress_Testing ...	Login	5845	Success	15020	266	165	82
495	17:07:33.088	Stress_Testing ...	Login	6435	Success	15020	266	645	58
496	17:07:33.915	Stress_Testing ...	Login	5762	Success	15020	266	1158	633
497	17:07:34.042	Stress_Testing ...	Login	5887	Success	15020	266	1098	607
498	17:07:33.749	Stress_Testing ...	Login	6179	Success	15020	266	692	115
499	17:07:33.897	Stress_Testing ...	Login	6259	Success	15020	266	1167	642
500	17:07:33.901	Stress_Testing ...	Login	6790	Success	15020	266	1163	641

Gambar III.14.  
Hasil respons stresstestinghalaman login

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
Login	500	2243	264	6750	1278.73	0.00%	65.7/sec	963.74	17.07	15020.1
TOTAL	500	2243	264	6750	1278.73	0.00%	65.7/sec	963.74	17.07	15020.1

Gambar III.15.  
Summary report stress testing halaman login

Hasil Identifikasi load testing :

- a) Samples (Jumlah Sampel): 500, Artinya JMeter telah mengirimkan total 500 permintaan (requests) ke server untuk proses Login ini.
- b) Average (Rata-rata): 2243 ms (atau sekitar 2,24 detik).
- c) Min (264 ms) dan Max (6750 ms): artinya Permintaan tercepat dilayani dalam waktu kurang dari 0,3 detik dan Permintaan terlama dilayani dalam waktu hampir dari 7 detik.
- d) Error %: 0.00% : Tidak ada permintaan yang gagal (HTTP 500, Timeout, atau koneksi putus). Semua 500 permintaan berhasil diproses oleh server.
- e) Throughput 65.7 /sec : Server mampu menangani sekitar 65 hingga 66 transaksi login per detik pada beban ini.

2) Dashboard/Halaman Absensi

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename:  Browse Log/Display Only:  Errors  Successes  Configure

Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Time...
481	17:11:25.476	Stress_Testing...	Dashboard_Ab...	5323	✓	15630	399	161	0
482	17:11:24.003	Stress_Testing...	Dashboard_Ab...	6886	✓	15630	399	107	0
483	17:11:25.403	Stress_Testing...	Dashboard_Ab...	5407	✓	15630	399	221	0
484	17:11:25.632	Stress_Testing...	Dashboard_Ab...	5213	✓	15630	399	58	0
485	17:11:28.352	Stress_Testing...	Dashboard_Ab...	2656	✗	8800	399	53	0
486	17:11:28.357	Stress_Testing...	Dashboard_Ab...	2678	✓	15630	399	47	0
487	17:11:27.372	Stress_Testing...	Dashboard_Ab...	3699	✓	15630	399	40	0
488	17:11:27.809	Stress_Testing...	Dashboard_Ab...	3262	✓	15630	399	66	0
489	17:11:28.990	Stress_Testing...	Dashboard_Ab...	2195	✓	15630	399	91	0
490	17:11:28.559	Stress_Testing...	Dashboard_Ab...	3682	✓	15630	399	51	0
491	17:11:28.919	Stress_Testing...	Dashboard_Ab...	2360	✓	15630	399	60	0
492	17:11:25.511	Stress_Testing...	Dashboard_Ab...	5819	✓	15630	399	136	0
493	17:11:25.458	Stress_Testing...	Dashboard_Ab...	5880	✓	15630	399	175	0
494	17:11:25.610	Stress_Testing...	Dashboard_Ab...	5753	✓	15630	399	69	0
495	17:11:28.871	Stress_Testing...	Dashboard_Ab...	2698	✓	15630	399	64	0
496	17:11:28.873	Stress_Testing...	Dashboard_Ab...	2702	✓	15630	399	63	0
497	17:11:25.476	Stress_Testing...	Dashboard_Ab...	6180	✓	15630	399	183	0
498	17:11:27.620	Stress_Testing...	Dashboard_Ab...	4211	✓	15630	399	49	0
499	17:11:25.194	Stress_Testing...	Dashboard_Ab...	6207	✓	15630	399	85	0
500	17:11:28.597	Stress_Testing...	Dashboard_Ab...	3680	✓	15630	399	45	0

Scroll automatically?  Child samples? No of Samples: 500 Latest Sample: 3880 Successes: 499 Failures: 1/500

Gambar III.16.  
Hasil *respons stress testing* halaman absensi

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse Log/Display Only:  Errors  Successes  Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/s	Sent KB/sec	Avg. Bytes
Dashboard_A...	500	1969	148	6981	1586.80	0.20%	47.4/sec	722.43	18.46	15616.4
TOTAL	500	1969	148	6981	1586.80	0.20%	47.4/sec	722.43	18.46	15616.4

Gambar III.17.  
*Summary report stress testing* halaman absensi

Hasil Identifikasi load testing pada halaman dashboard :

a) Average (Rata-rata): 1960 ms (atau sekitar 1,96 detik).

b) Min (148 ms) dan Max (6981 ms): artinya Permintaan tercepat dilayani dalam waktu kurang dari 0,2 detik dan Permintaan terlama dilayani dalam waktu hampir dari 7 detik.

c) Error %: 0.02% : Dari 500 sampel, 0.20% berarti ada 1 request yang gagal.

d) Throughput 47.4 /sec : Server mampu menangani sekitar 47 transaksi login per detik pada beban ini.

### 3) Halaman Penggajian

The screenshot shows the 'View Results in Table' window in JMeter. The table displays individual test results for 500 samples. The columns include Sample #, Start Time, Thread Name, Label, Sample Time, Status, Bytes, Sent Bytes, Latency, and Connect Time. All samples show a status of 'Success' and a latency of approximately 3439 ms.

Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Time...
481	17:44:42.686	Thread Group ...	HTTP Request	6538	Success	15670	409	171	83
482	17:44:43.242	Thread Group ...	HTTP Request	6321	Success	15670	409	481	209
483	17:44:43.177	Thread Group ...	HTTP Request	6393	Success	15670	409	435	192
484	17:44:43.078	Thread Group ...	HTTP Request	6497	Success	15670	409	389	183
485	17:44:43.379	Thread Group ...	HTTP Request	6348	Success	15670	409	622	251
486	17:44:42.829	Thread Group ...	HTTP Request	6951	Success	15670	409	226	110
487	17:44:42.970	Thread Group ...	HTTP Request	6817	Success	15670	409	301	112
488	17:44:43.236	Thread Group ...	HTTP Request	6552	Success	15670	409	497	210
489	17:44:43.253	Thread Group ...	HTTP Request	7461	Success	15670	409	91	41
490	17:44:43.297	Thread Group ...	HTTP Request	6709	Success	15670	409	330	205
491	17:44:42.887	Thread Group ...	HTTP Request	7157	Success	15670	409	1900	1452
492	17:44:43.448	Thread Group ...	HTTP Request	6608	Success	15670	409	652	285
493	17:44:43.071	Thread Group ...	HTTP Request	6993	Success	15670	409	387	180
494	17:44:43.419	Thread Group ...	HTTP Request	6667	Success	15670	409	644	271
495	17:44:42.886	Thread Group ...	HTTP Request	7315	Success	15670	409	239	113
496	17:44:43.113	Thread Group ...	HTTP Request	7486	Success	15670	409	423	203
497	17:44:43.210	Thread Group ...	HTTP Request	7280	Success	15670	409	485	212
498	17:44:43.086	Thread Group ...	HTTP Request	7314	Success	15670	409	395	190
499	17:44:43.006	Thread Group ...	HTTP Request	7612	Success	15670	409	322	130
500	17:44:43.155	Thread Group ...	HTTP Request	7463	Success	15670	409	443	198

Gambar III.18.  
Hasil *respons stress testing* halaman penggajian

The screenshot shows the 'Summary Report' window in JMeter. The table provides a summary of the test results, including the number of samples, average, minimum, maximum, standard deviation, error percentage, throughput, and average bytes for each label.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
HTTP Request	500	3439	367	7612	1608.02	0.00%	61.1/sec	935.72	24.42	15670.6
TOTAL	500	3439	367	7612	1608.02	0.00%	61.1/sec	935.72	24.42	15670.6

Gambar III.19.  
*Summary report stress testing* halaman absensi

Hasil Identifikasi load testing :

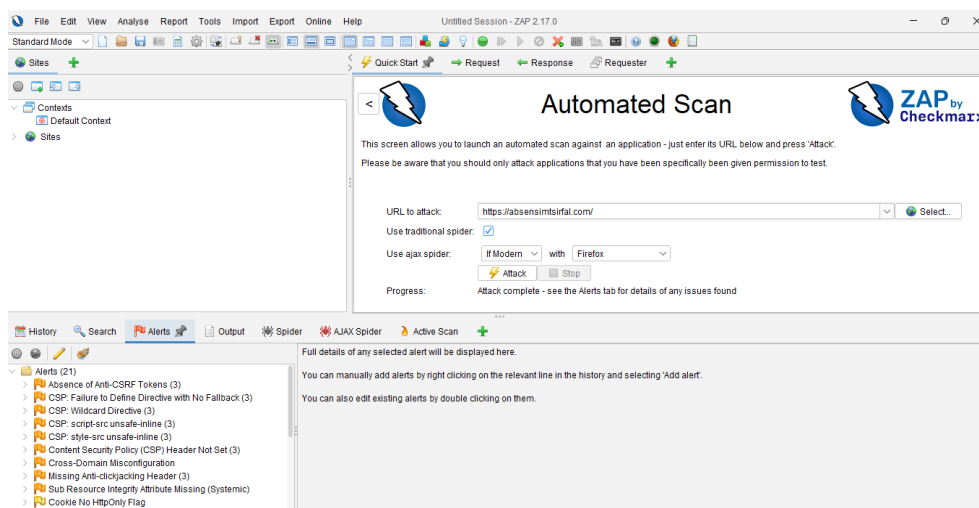
- Average (Rata-rata): 3439 ms (atau sekitar 3,44 detik).
- Min (367 ms) dan Max (7612 ms): artinya Permintaan tercepat dilayani dalam waktu kurang dari 0,4 detik dan Permintaan terlama dilayani dalam waktu hampir dari 8 detik.
- Error %: 0.00% : Tidak ada permintaan yang gagal (HTTP 500, Timeout,

atau koneksi putus). Semua 500 permintaan berhasil diproses oleh server.

- d) Throughput 61.1 /sec : Server mampu menangani sekitar 61 transaksi login per detik pada beban ini, dimana Server masih mampu menerima request dengan cukup cepat, tetapi penyelesaiannya yang lambat.

## 2. Pengujian Keamanan Website (Security Testing)

Tujuan dari Pengujian Keamanan Website adalah untuk memastikan bahwa website atau aplikasi web yang dibangun aman dari potensi ancaman atau serangan yang dapat merusak integritas, kerahasiaan, dan ketersediaan data serta sistem yang ada. Pada kali ini pengujian dilakukan dengan menggunakan *software OWASP ZAP*. *OWASP ZAP (Zed Attack Proxy)* adalah alat pendeteksi keamanan *open source* yang dirancang untuk membantu pengujian keamanan aplikasi web. Fungsi utama dari *OWASP ZAP* adalah untuk menemukan kerentanannya dalam aplikasi web dan membantu pengembang atau penguji dalam mengidentifikasi potensi risiko serta memperbaiki masalah yang ditemukan.



Gambar III. 20.

Hasil *scanning* pengujian keamanan *website absensimtsirfal.com*

Dari hasil scan pengujian keamanan website <https://absensimtsirfal.com> maka dihasilkan report sebagai berikut:

Tabel III.9. hasil *scanning* pengujian keamanan *website*

Peringatan (Alert)	URL	Risiko (Risk)	Deskripsi (Description)
Absence of Anti-CSRF Tokens	<a href="https://absensimtsirfal.com/">https://absensimtsirfal.com/</a>	Medium (sedang)	No Anti-CSRF tokens were found in a HTML submission form.
CSP: Failure to Define Directive with No Fallback	<a href="https://absensimtsirfal.com/">https://absensimtsirfal.com/</a>	Medium (sedang)	The Content Security Policy fails to define one of the directives that has no fallback. Missing/excluding them is the same as allowing anything.
CSP: script-src unsafe-inline	<a href="https://absensimtsirfal.com/">https://absensimtsirfal.com/</a>	Medium (sedang)	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data injection attacks.
Content Security Policy (CSP) Header Not Set	<a href="https://absensimtsirfal.com/robots.txt">https://absensimtsirfal.com/robots.txt</a>	Medium (sedang)	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks.
Cross-Domain Misconfiguration	<a href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css</a>	Medium (sedang)	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain
Missing Anti-clickjacking Header	<a href="https://absensimtsirfal.com/">https://absensimtsirfal.com/</a>	Medium (sedang)	The response does not protect against 'ClickJacking'

<b>Peringatan (Alert)</b>	<b>URL</b>	<b>Risiko (Risk)</b>	<b>Deskripsi (Description)</b>
			attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.
Sub Resource Integrity Attribute Missing	<a href="https://absensimtsirfal.com/sitemap.xml">https://absensimtsirfal.com/sitemap.xml</a>	Medium (sedang)	The integrity attribute is missing on a script or link tag served by an external server. The integrity tag prevents an attacker who have gained access to this server from injecting a malicious content.
Cookie No Http Only Flag	<a href="https://absensimtsirfal.com/">https://absensimtsirfal.com/</a>	Rendah (low)	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
Cookie without SameSite Attribute	<a href="https://absensimtsirfal.com/">https://absensimtsirfal.com/</a>	Rendah (low)	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
Cross-Domain JavaScript Source File Inclusion	<a href="https://absensimtsirfal.com/robots.txt">https://absensimtsirfal.com/robots.txt</a>	Rendah (low)	The page includes one or more script files from a third-party domain.
Server Leaks Information via "X-Powered-By" HTTP Response Header	<a href="https://absensimtsirfal.com/">https://absensimtsirfal.com/</a>	Rendah (low)	The web/application server is leaking information via one or more "X-Powered-

<b>Peringatan (Alert)</b>	<b>URL</b>	<b>Risiko (Risk)</b>	<b>Deskripsi (Description)</b>
Field(s)			By" HTTP response headers. Access to such information may facilitate attackers identifying other frameworks/components your web application is reliant upon and the vulnerabilities such components may be subject to.
Strict-Transport-Security Header Not Set	<a href="https://absensimsirf.al.com/sitemap.xml">https://absensimsirf.al.com/sitemap.xml</a>	Rendah (low)	HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797
Timestamp Disclosure - Unix	<a href="https://absensimsirf.al.com/sitemap.xml">https://absensimsirf.al.com/sitemap.xml</a>	Rendah (low)	A timestamp was disclosed by the application/web server. - Unix
X-Content-Type-Options Header Missing	<a href="https://absensimsirf.al.com/">https://absensimsirf.al.com/</a>	Rendah (low)	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type

1. Anti-CSRF Tokens Missing (Medium Risk)
  - a. Deskripsi: Tidak ditemukan token Anti-CSRF pada formulir pengiriman HTML.
  - b. Risiko: Tanpa token ini, aplikasi rentan terhadap serangan Cross-Site Request Forgery (CSRF), yang dapat mengeksploitasi sesi pengguna untuk melakukan tindakan yang tidak sah.
2. Content Security Policy (CSP) Issues (Medium Risk)
  - a. Deskripsi: Beberapa masalah terkait Content Security Policy (CSP) ditemukan, seperti kegagalan untuk mendefinisikan arahan yang tidak memiliki fallback, penggunaan wildcard, dan penggunaan script-src unsafe-inline.
  - b. Risiko: CSP yang tidak tepat dapat menyebabkan Cross-Site Scripting (XSS) dan serangan data injection.
3. CORS Misconfiguration (Medium Risk)
  - a. Deskripsi: Konfigurasi CORS yang salah memungkinkan permintaan lintas domain dari domain pihak ketiga yang tidak sah.
  - b. Risiko: Ini bisa mengarah pada kebocoran data atau penyalahgunaan API yang tidak terautentikasi.
4. Clickjacking Protection Missing (Medium Risk)
  - a. Deskripsi: Tidak ada perlindungan terhadap serangan Clickjacking.
  - b. Risiko: Penyerang dapat menyembunyikan elemen berbahaya di dalam iframe, yang dapat mengarahkan pengguna untuk melakukan tindakan yang tidak sah.
5. Missing Subresource Integrity (SRI) Attribute (Medium Risk)

- a. Deskripsi: Atribut integrity hilang pada tag skrip atau link yang disajikan oleh server eksternal.
  - b. Risiko: Penyerang yang berhasil mendapatkan akses ke server eksternal dapat menyisipkan konten berbahaya.
6. Cookies Without HttpOnly Flag (Low Risk)
- a. Deskripsi: Cookie tidak disetel dengan flag HttpOnly, yang memungkinkan akses ke cookie melalui JavaScript.
  - b. Risiko: Jika ada skrip berbahaya, cookie yang tidak terlindungi bisa dicuri, berpotensi menyebabkan session hijacking.
7. Cookies Without SameSite Attribute (Low Risk)
- a. Deskripsi: Cookie tidak memiliki atribut SameSite, yang memungkinkan pengiriman cookie melalui permintaan lintas situs.
  - b. Risiko: Ini dapat memudahkan serangan CSRF dan Cross-Site Script Inclusion.
8. Cross-Domain JavaScript Source File Inclusion (Low Risk)
- a. Deskripsi: Halaman menyertakan file skrip dari domain pihak ketiga.
  - b. Risiko: Menggunakan skrip dari pihak ketiga dapat memperkenalkan potensi celah keamanan, seperti skrip berbahaya yang dapat dieksekusi.
9. Server Leaking Information via "X-Powered-By" Header (LowRisk)
- a. Deskripsi: Server mengungkapkan informasi tentang teknologi yang digunakan melalui header X-Powered-By.
  - b. Risiko: Penyerang dapat memanfaatkan informasi ini untuk mencari kerentanannya.
10. Strict-Transport-Security Header Missing (Low Risk)
- a. Deskripsi: Header Strict-Transport-Security (HSTS) tidak disetel.

- b. Risiko: Tanpa HSTS, situs web rentan terhadap serangan man-in-the-middle, di mana penyerang dapat memaksa komunikasi untuk menggunakan HTTP yang tidak aman.
11. Timestamp Disclosure - Unix (Low Risk)
- a. Deskripsi: Stempel waktu (timestamp) dibocorkan oleh server web.
  - b. Risiko: Mengetahui timestamp dapat memberikan informasi tentang konfigurasi atau operasi server, yang dapat digunakan oleh penyerang.
12. X-Content-Type-Options Header Missing (Low Risk)
- a. Deskripsi: Header X-Content-Type-Options tidak disetel ke 'nosniff'.
  - b. Risiko: Ini memungkinkan browser untuk menebak tipe konten, yang bisa menyebabkan interpretasi yang salah terhadap konten dan potensi serangan MIME sniffing.

### **3.4. Tahap Implementasi**

#### **3.4.1 Tahap Penerapan IPTEK**

Tahap implementasi merupakan fase krusial di mana rancangan sistem (desain) dan kode program (konstruksi) yang telah dinyatakan lulus uji, diletakkan ke dalam lingkungan operasional yang sebenarnya. Tujuan utama tahap ini adalah menjamin transisi sistem penggajian dari metode manual berbasis kertas/Excel menuju sistem informasi terintegrasi berbasis web yang dilengkapi kecerdasan buatan (*Machine Learning*).

#### **A. Kebutuhan Hardware dan Software**

##### **1. Perangkat Keras (*Hardware*)**

Tabel berikut merincikan spesifikasi fisik perangkat yang dibutuhkan untuk menjalankan sistem.

Tabel III.10.  
Spesifikasi Kebutuhan Perangkat Keras (Hardware)

Lingkungan	Perangkat / Komponen	Spesifikasi Minimum	Rekomendasi	Fungsi Utama
<b>Server</b> ( <i>Cloud Hosting</i> )	<b>Processor</b> ( <b>vCPU</b> )	1 Core @ 2.0 GHz	2 Core @ 2.4 GHz	Menjalankan proses komputasi algoritma <i>Linear Regression</i> .
	<b>Memory</b> ( <b>RAM</b> )	1 GB	2 GB	Menangani permintaan akses ( <i>request</i> ) dari banyak guru secara bersamaan.
	<b>Storage</b>	5 GB	10 GB (SSD)	Menyimpan file sistem, database, dan arsip PDF slip gaji.
	<b>Bandwidth</b>	50 GB / Bulan	Unmetered	Jalur lalu lintas data saat pengunduhan slip gaji.
<b>Client</b> ( <i>Administrator</i> )	<b>Perangkat</b>	PC Desktop / Laptop	PC / Laptop (Core i3)	Digunakan untuk input data absensi dan verifikasi gaji.
	<b>RAM</b>	4 GB	8 GB	Agar lancar menjalankan browser dan Excel bersamaan.
	<b>Monitor</b>	Resolusi 1366x768	Resolusi 1920x1080	Memaksimalkan tampilan tabel data gaji yang lebar.
<b>Client</b> ( <i>Guru &amp; Kepsek</i> )	<b>Perangkat</b>	Smartphone / Tablet	Smartphone 4G	Mengakses dashboard dan mengunduh slip gaji secara mobile.
<b>Jaringan</b>	<b>Internet</b>	5 Mbps (Stabil)	10 Mbps (Fiber Optic)	Koneksi utama untuk sinkronisasi data ke server.

## 2. Perangkat Lunak (Software)

Tabel berikut merincikan lisensi dan jenis perangkat lunak yang harus terinstal agar aplikasi dapat berjalan tanpa *error*.

Tabel III.11.  
Spesifikasi Kebutuhan Perangkat Lunak (Software)

Kategori	Nama Perangkat Lunak	Versi Minimum	Keterangan / Fungsi
----------	----------------------	---------------	---------------------

<b>Sistem Operasi Server</b>	Linux / Windows Server	Ubuntu 20.04 LTS / Windows Server 200	OS standar untuk <i>Web Hosting</i> yang stabil dan aman.
<b>Web Server</b>	Apache / Nginx	Apache 2.4	Melayani permintaan HTTP/HTTPS dan menjalankan skrip PHP.
<b>Bahasa Pemrograman</b>	PHP	Versi 7.4 atau 8.x	Bahasa utama aplikasi. Wajib versi 7.4+ untuk dukungan library matematika.
<b>Basis Data (DBMS)</b>	MySQL / MariaDB	MySQL 5.7	Tempat penyimpanan data relasional (Tabel Guru, Gaji, Absen).
<b>Management Tool</b>	phpMyAdmin	Versi 5.x	Antarmuka grafis untuk mempermudah pengelolaan database.
<b>Web Browser (Client)</b>	Google Chrome / Edge	Versi Terbaru	Aplikasi pengakses sistem. Chrome direkomendasikan karena <i>engine</i> V8-nya cepat.
<b>Document Viewer</b>	Adobe Reader / Browser	-	Untuk membuka file output Slip Gaji & Laporan (PDF).
<b>Text Editor (Dev)</b>	Visual Studio Code	-	Alat bantu jika diperlukan perbaikan kode ( <i>maintenance</i> ) di sekolah.

### 3. Instalasi dan Konfigurasi Sistem

Proses ini melibatkan pemindahan kode dari lingkungan pengembangan (localhost) ke server produksi (live server).

#### 1) Migrasi Database:

- a) Mengimpor skema database (db\_penggajian.sql) yang telah dirancang pada tahap ERD ke phpMyAdmin di hosting.
- b) Memastikan relasi *Foreign Key* antara tabel tbl\_guru, tbl\_absensi, dan tbl\_penggajian aktif untuk mencegah *orphan data* (data yatim/tidak valid).

#### 2) Deployment Source Code:

- a) Mengunggah seluruh file proyek ke direktori public\_html melalui FTP.
- b) Mengonfigurasi file config/database.php menyesuaikan *username* dan *password* database server sekolah.

### 4. Inisialisasi Data Awal (Data Seeding)

Ini adalah langkah paling krusial untuk fitur Machine Learning. Algoritma Linear Regression tidak dapat bekerja jika database kosong.

- a) Tindakan: Peneliti bersama Admin TU melakukan input data historis manual (rekap gaji 12 bulan ke belakang) ke dalam tabel `tbl_penggajian`.
- b) Tujuan: Membentuk *Dataset Latih* (\$Training Set\$) awal. Sehingga, pada hari pertama sistem diluncurkan, fitur "Prediksi Anggaran" sudah langsung bisa menampilkan grafik tren dan angka estimasi, tanpa harus menunggu data terkumpul setahun lagi.

## B. Arsitektur IPTEK

Arsitektur sistem menggambarkan struktur konseptual dan logika bagaimana perangkat lunak diorganisasikan. Dalam pengembangan "Sistem Penggajian Online dengan Prediksi Anggaran Berbasis Machine Learning" ini, arsitektur yang digunakan adalah Three-Tier Architecture (Arsitektur Tiga Lapis) yang diimplementasikan menggunakan pola desain Model-View-Controller (MVC). Pemilihan arsitektur ini didasarkan pada kebutuhan sistem yang harus dapat diakses melalui internet (*Web-Based*) namun tetap menjaga keamanan data gaji dan efisiensi proses perhitungan algoritma.

### 1. Arsitektur Tiga Lapis (Three-Tier Architecture)

Sistem ini membagi fungsi aplikasi menjadi tiga lapisan fisik yang terpisah namun saling terintegrasi:

#### a) Presentation Tier (Lapis Presentasi / Client Side)

- 1) Fungsi: Merupakan antarmuka (*User Interface*) yang berhadapan langsung dengan pengguna (Administrator, Guru).
- 2) Teknologi: HTML5, CSS3, JavaScript, dan Bootstrap Framework.
- 3) Perangkat: Web Browser (Chrome/Edge) pada PC atau Smartphone

pengguna.

- 4) Peran: Menerima input data absensi dan menampilkan output grafik prediksi anggaran.

b) Application Tier (Lapis Aplikasi / Server Side)

- 1) Fungsi: Merupakan "otak" dari sistem yang memproses logika bisnis. Di sinilah Algoritma Linear Regression dijalankan.
- 2) Teknologi: PHP (Hypertext Preprocessor) Versi 8.x.
- 3) Peran: Menerima *request* dari Client, melakukan perhitungan gaji, menjalankan rumus regresi, dan mengirimkan hasilnya kembali ke Client. Lapisan ini juga menangani autentikasi keamanan (*Login Session*).

c) Data Tier (Lapis Data / Database Server)

- 1) Fungsi: Tempat penyimpanan data persisten.
- 2) Teknologi: MySQL / MariaDB.
- 3) Peran: Menyimpan tabel guru, gaji, absensi, dan config. Lapisan ini hanya bisa diakses oleh *Application Tier*, tidak bisa diakses langsung oleh *Client* demi keamanan.

2. Arsitektur Perangkat Lunak (Pola MVC)

Untuk memudahkan pengembangan dan pemeliharaan kode program, struktur perangkat lunak disusun menggunakan pola MVC:

a) Model (Manajemen Data):

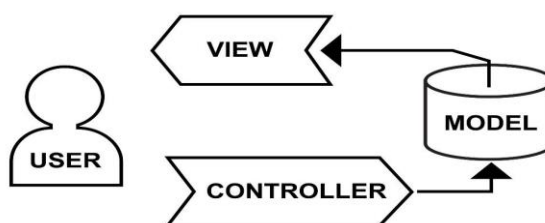
- 1) Bertanggung jawab menangani koneksi ke database.
- 2) Contoh: File `GajiModel.php` berisi fungsi `getRiwayatGaji()` yang mengambil data raw dari database untuk kebutuhan prediksi.

b) View (Tampilan):

- 1) Bertanggung jawab menampilkan data ke layar pengguna.
- 2) Contoh: File dashboard\_kepek.php yang berisi kode grafik (Chart.js) untuk memvisualisasikan hasil prediksi.

c) Controller (Logika & Algoritma):

- 1) Bertanggung jawab menghubungkan Model dan View serta memproses algoritma.
- 2) Contoh: File Prediksi Controller. php berisi logika matematika  $\Sigma x$ ,  $\Sigma y$  untuk menghitung prediksi anggaran sebelum dikirim ke View.



Gambar III.21.

*Model View Controller*

### 3.4.2. Tahap Penerimaan IPTEK

Teknologi yang canggih tidak akan berguna jika sumber daya manusia (SDM) menolak menggunakannya, untuk itu tahap penerimaan IPTEK ini berfokus pada pendekatan persuasif dan edukatif kepada seluruh staf MTs Irsyadul Athfal.

#### A. Sosialisasi Perubahan Alur Kerja (Change Management)

Peneliti melakukan presentasi di hadapan Kepala Madrasah dan Guru untuk menjelaskan manfaat sistem baru.

1. Poin Penekanan: Menjelaskan bahwa sistem ini bukan untuk "memantau" guru secara kaku, melainkan untuk transparansi (guru bisa cek hitungan gaji sendiri) dan kepastian anggaran (sekolah tidak defisit).
2. Target: Mengurangi resistensi (penolakan) dari staf yang sudah nyaman dengan

cara lama.

## B. Pelatihan Pengguna Berbasis Peran (Role-Based Training)

Pelatihan dilakukan secara spesifik sesuai dengan aktor pada Use Case Diagram:

### 1. Pelatihan Admin Tata Usaha (Operator Teknis)

- a) Fokus: Cara input data master guru dan input absensi jam mengajar (Variabel \$x\$).
- b) Penekanan: Admin diajarkan bahwa kesalahan input jam mengajar akan menyebabkan *Garbage In, Garbage Out* (Gaji salah, Prediksi juga salah).

### 2. Pelatihan Bendahara (Validator Keuangan)

- a) Fokus: Verifikasi hasil hitung sistem, manajemen tabel `tbl_config` (ubah tarif transport/jabatan), dan cetak laporan PDF.
- b) Penekanan: Cara membaca output Linear Regression (Intercept & Slope) yang diterjemahkan menjadi bahasa awam untuk laporan ke Yayasan.

### 3. Pelatihan Kepala Madrasah (Eksekutif)

- a) Fokus: Cara login ke Dashboard Pimpinan.
- b) Penekanan: Cara membaca grafik visualisasi prediksi untuk pengambilan keputusan strategis (misal: apakah dana cukup untuk merekrut guru baru bulan depan?).

## C. Strategi Parallel Run (Uji Jalan Berdampingan)

Untuk meminimalisir risiko kegagalan sistem, diterapkan strategi Parallel Run selama 1 bulan pertama.

Mekanisme: Proses penggajian dilakukan dua kali, yaitu menggunakan cara lama (Excel) dan sistem baru (Web).

Tujuan: Membandingkan hasil akhir. Jika angka gaji di Web sama persis dengan Excel, maka sistem dinyatakan valid dan aman. Jika ada selisih, dilakukan *debugging* pada rumus kode program.

#### D. Serah Terima dan Dokumentasi (Handover)

Setelah masa pendampingan selesai dan sistem berjalan lancar, dilakukan serah terima resmi:


1. Penyerahan Akun Super-Admin: Hak akses penuh database diserahkan ke sekolah.
2. Buku Panduan (User Manual): Dokumen cetak & PDF yang berisi tutorial langkah-demi-langkah bergambar untuk solusi jika Admin lupa caranya.
3. Pakta Integritas: Peneliti menjamin tidak menyimpan *backdoor* atau akses tersembunyi demi keamanan data privasi guru.

Tabel III.22.  
Ringkasan Tabel Implementasi

Aspek	Kegiatan Detail	Tujuan / Luaran
Teknologi	Instalasi Server & HTTPS	Sistem aman dan dapat diakses online 24 jam.
Data	Input Historis 12 Bulan (Seeding)	Algoritma ML langsung cerdas sejak hari pertama.
Manusia	Pelatihan Administrator	SDM mampu mengoperasikan fitur tanpa bantuan peneliti.
Proses	<i>Parallel Run</i> (Web vs Excel)	Membuktikan akurasi hitungan 100% valid.

Tahapan selanjutnya dilakukan pengujian keterimaan oleh pengguna, apakah sistem mudah digunakan atau tidak, berikut dokumen uji keterimaan aplikasi dibawah.

Tabel III.13  
User Acceptance Testing

DOKUMEN USER ACCEPTANCE TESTING					
Nama Proyek	: Sistem Penggajian Online Dengan Prediksi Anggaran Berbasis Machine Learning Pada Mts Irsyadul Athfal Depok				
Studi Kasus / Mitra	: MTs Irsyadul Athal				
Manajer Proyek	: Muslimah				
Proses Pengujian					
No	Use Case	Hasil Uji [Berhasil/Gagal]	Nama Penguji	Tanggal Pengujian	Catatan Pengujian
1	<p>Usecase Uji : Login</p> <p>Deskripsi : Menginput usemame dan password yang valid ke form login</p> <p>Kasus Pengujian Usemame : admin Password : 123</p> <p>Hasil yang diharapkan : -Sistem memvalidasi akun dan jika login berhasil akan masuk kedalam halaman dashboard sesuai hak akses untuk admin -Jika login tidak berhasil maka tidak akan masuk kedalam halaman dashboard admin dan menampilkan display pesan kesalahan</p>	Berhasil	Subhan Jayani	26 Januari 2026	
2	<p>Usecase Uji : Absensi Guru</p> <p>Deskripsi : Menginput usemame dan password yang valid ke form login</p> <p>Kasus Pengujian Usemame : randani Password : 123</p> <p>Hasil yang diharapkan : -Jika login berhasil akan masuk kedalam halaman dashboard dan memvalidasi lokasi melalui Gps serta menampilkan informasi lokasi kemudian guru dapat menginput absen masuk -Jika login tidak berhasil maka tidak akan masuk kedalam halaman dashboard guru dan menampilkan display pesan kesalahan</p>	Berhasil	Ramdani	26 Januari 2026	
3	<p>Usecase Uji : Hitung gaji</p> <p>Deskripsi : User memilih menu penggajian dan menekan tombol proses gaji untuk periode bulan tertentu</p> <p>Kasus Pengujian Usemame : admin Password : 123</p> <p>Hasil yang diharapkan : -Sistem menghitung total gaji secara otomatis berdasarkan komponen gaji</p>	Berhasil	Marpuah	26 Januari 2026	

### 3.5. Uraian Tugas

Pengembangan "Sistem Penggajian Online dengan Prediksi Anggaran Berbasis Machine Learning pada MTs Irsyadul Athfal Depok" merupakan proyek yang kompleks yang melibatkan manajemen data, rekayasa perangkat lunak, dan analisis statistik. Oleh karena itu, proses pengembangan dikelompokkan menjadi tiga peran fungsional utama. Setiap peran memiliki tanggung jawab spesifik, *tools* yang digunakan dan *output* yang dihasilkan.

#### 3.5.1. Project Manager/Sistem Analis

Dalam pengembangan sistem ini yang bertugas sebagai Project Manager/Sistem Analis adalah **Muslimah**, yang mempunyai peran sebagai inisiator dan perancang utama. Tugas utamanya adalah menjembatani kesenjangan komunikasi antara kebutuhan pihak sekolah (mitra) dengan solusi teknis yang akan dibangun.

1. Fokus Utamanya adalah perencanaan, analisis kebutuhan dan desain logika.
2. Rincian Tugas & Kegiatan
  - A. Analisis Kebutuhan (Requirement Engineering)
    - 1) Melakukan wawancara mendalam (*in-depth interview*) dengan Kepala Madrasah dan Bendahara untuk membedah komponen gaji (Gaji Pokok, Tunjangan Jabatan, Transport, Potongan).
    - 2) Mengidentifikasi masalah bisnis (contoh: keterlambatan rekapitulasi) dan menentukan solusi digital yang tepat.
    - 3) Menentukan variabel-variabel data historis yang valid untuk dijadikan dataset prediksi (Variabel X: Waktu/Jam Mengajar, Variabel Y: Total Pengeluaran).

## B. Manajemen Jadwal & Risiko

- 1) Menyusun Timeline penelitian (Gantt Chart) dari tahap pengumpulan data hingga sidang skripsi.
- 2) Melakukan analisis risiko (contoh: risiko data hilang) dan merencanakan mitigasinya (fitur backup database).

## C. Perancangan Sistem (System Modeling)

- 1) Menerjemahkan kebutuhan bisnis menjadi diagram teknis: Use Case Diagram, Activity Diagram, dan Sequence Diagram.
- 2) Merancang antarmuka (Mockup/Wireframe) aplikasi agar mudah digunakan (User Friendly) oleh guru yang awam teknologi.
- 3) Tools/Alat Bantu: Microsoft Visio / StarUML (Desain Diagram), Trello (Manajemen Tugas), Figma (Desain Antarmuka).
- 4) Output (Luaran): Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL), Dokumen Desain Sistem (UML), dan Prototipe Desain UI.

### 3.5.2. Db Administrator/Programmer

Dalam pengembangan sistem ini yang bertugas sebagai Db Administrator/Programmer adalah **Muhammad Rizki Beurata**, mempunyai peran sebagai eksekutor teknis. Bertanggung jawab menerjemahkan desain yang dibuat oleh Analis Sistem menjadi baris kode yang berfungsi dan struktur data yang efisien.

1. Fokus Utamanya adalah pengkodean (Coding), manajemen database, dan implementasi algoritma.
2. Rincian Tugas & Kegiatan

#### A. Database Engineering (Peran DBA)

- 1) Merancang skema database (ERD) dan melakukan normalisasi tabel hingga Third Normal Form (3NF) untuk mencegah redundansi data guru.
- 2) Melakukan data seeding (Input Data Awal) seperti memasukkan data gaji riil 12 bulan terakhir ke tabel database sebagai data latih (Training Data)
- 3) Mengatur keamanan database, termasuk enkripsi password menggunakan Bcrypt/Argon2.

#### B. Backend Development (Peran Programmer)

- 1) Membangun logika CRUD (Create, Read, Update, Delete) untuk data guru dan absensi menggunakan bahasa PHP.
- 2) Implementasi Machine Learning  
Mengonversi rumus matematika Linear Regression (Metode Kuadrat Terkecil) ke dalam fungsi algoritma PHP untuk menghitung nilai Slope ( $b$ ) dan Intercept ( $a$ ) secara otomatis.

#### C. Frontend Development

- 1) Mengimplementasikan desain UI menggunakan HTML, CSS, dan Bootstrap Framework agar responsif diakses melalui Smartphone guru.
- 2) Mengintegrasikan library Chart.js untuk memvisualisasikan hasil prediksi anggaran dalam bentuk grafik garis.
- 3) Tools/Alat Bantu yang digunakan adalah Visual Studio Code (Text Editor), XAMPP (Web Server Lokal), MySQL/MariaDB (DBMS), Git (Version Control).
- 4) Output (Luaran) seperti Source Code Aplikasi Final, File Database (.sql), dan Sistem yang terinstal di server (*Live Deployment*).

### 3.5.3. Pengujian Sistem

Yang bertugas sebagai Db Administrator/Programmer adalah **Erhan Agung Wijaya**, berperan sebagai penjaga kualitas (*Gatekeeper*). Bertanggung jawab memastikan sistem berjalan tanpa error dan hasil prediksi anggaran yang akurat.

1. Fokus Utamanya validasi fungsional, uji akurasi algoritma dan uji performa.
2. Rincian Tugas & Kegiatan

#### A. Pengujian Fungsional (*Black Box Testing*)

- 1) Membuat dokumen Skenario Uji (*Test Cases*).
- 2) Menguji validasi input (contoh: memastikan input "Gaji Pokok" tidak bisa diisi huruf).
- 3) Menguji alur kerja (workflow) contohnya: Apakah setelah absensi diinput, nominal gaji otomatis berubah?

#### B. Validasi Algoritma (*Accuracy Testing*)

- 1) Melakukan perhitungan manual regresi linear menggunakan microsoft excel.
- 2) Membandingkan hasil hitungan Excel dengan hasil sistem website.

#### C. Pengujian Performa & Keamanan (*Non-Functional Testing*)

- 1) Melakukan *Stress Test* menggunakan *Apache JMeter* untuk melihat ketahanan server saat diakses banyak guru dan *OWASP ZAP* untuk menguji keamanan *website*.
- 2) Memastikan data gaji guru A tidak bisa diintip oleh guru B (Uji Hak Akses/Session).
- 3) Tools/Alat Bantu: Aplikasi *Apache Jmeter* dan *OWASP ZAP*.
- 4) Output (Luaran): Dokumen Laporan Pengujian (*Test Report*), Tabel hasil pengujian sistem *scanning*.