

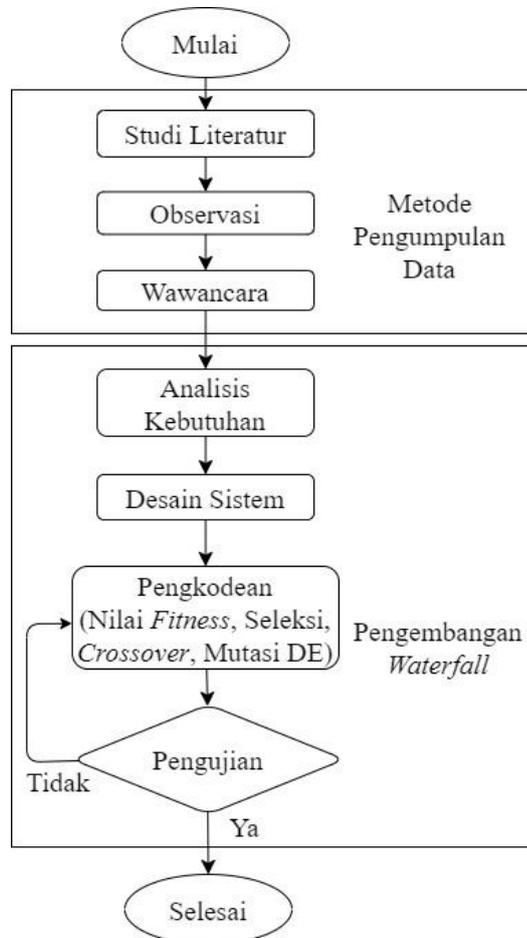
BAB III METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan metode *System Development Life Cycle* (SDLC) yang merupakan metode untuk mengembangkan sebuah sistem informasi. Kerangka kerja menjadi langkah awal dari SDLC dalam mengendalikan dan membuat rencana sistem. Model pengembangan SDLC salah satunya menggunakan *waterfall*. Model pengembangan *waterfall* prosesnya dilakukan secara berurutan (Anwardi dkk., 2020).

3.2 Alur Penelitian

Alur penelitian yang akan dilakukan secara garis besar dapat dilihat dari diagram alur pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

Dari diagram alir pada Gambar 3.1 secara rinci diuraikan sebagai berikut.

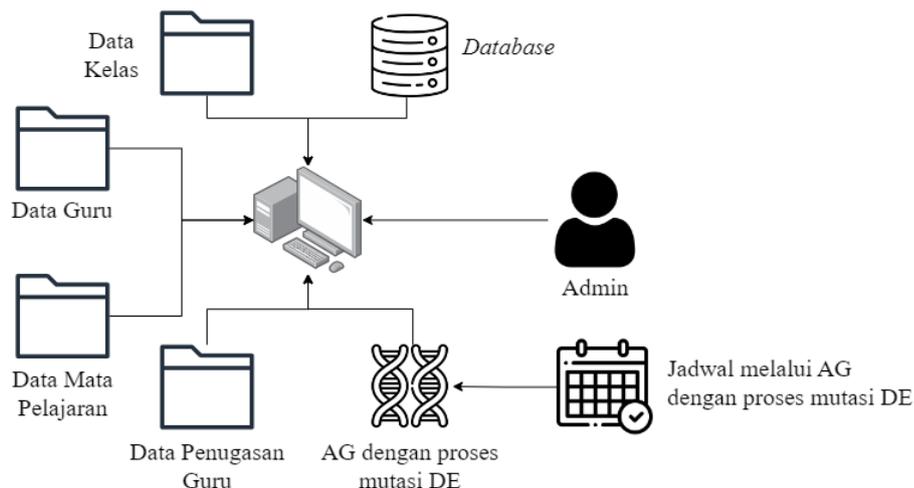
3.2.1 Metode Pengumpulan Data

Metode pengumpulan data dan informasi yang dilakukan untuk perancangan sistem ini sebagai berikut.

- Studi literatur yaitu proses pengumpulan data dengan membaca makalah, tesis, dan berbagai sumber relevan dengan topik yang dibahas.
- Observasi merupakan metode pengumpulan data dengan meninjau secara langsung terhadap objek yang akan diteliti. Data yang bersifat nyata dapat dilakukan riset pada tempat penelitian yaitu SMAN 2 Purwakarta. Dengan melihat bagaimana hasil dari penjadwalan yang dibuat oleh pihak kurikulum sekolah.
- Wawancara secara langsung dilakukan kepada kepala kurikulum sekolah SMAN 2 Purwakarta yang menentukan persebaran jadwal guru mengajar.

3.2.2 Analisis Kebutuhan

Perancangan dengan standar dan perangkat yang diperlukan. Dalam menganalisis kebutuhan diperlukan arsitektur sistem untuk mengetahui apa saja yang dibutuhkan pada penelitian ini. Pada Gambar 3.2 menunjukkan arsitektur sistem penjadwalan menggunakan Algoritma Genetika (GA) dengan proses mutasi *Differential Evolution* (DE).



Gambar 3.2 Arsitektur Sistem

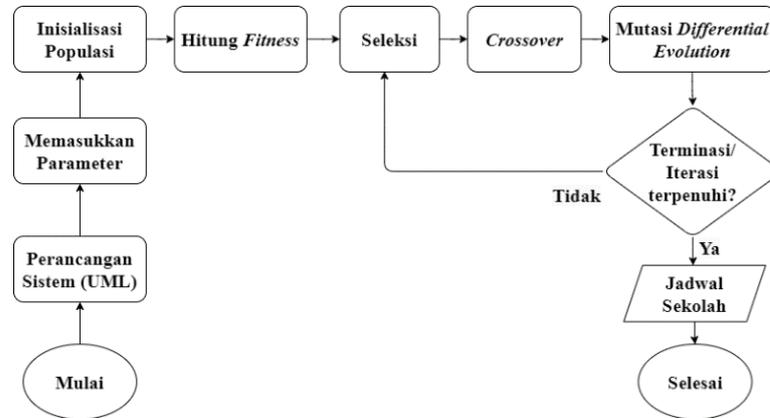
Admin memiliki peran penting dalam bagaimana sistem beroperasi. Sistem diakses oleh admin saat membuat jadwal. Pembuatan jadwal bergantung pada data kelas, data penugasan guru, data guru, dan data mata pelajaran. Data penjadwalan diaplikasikan menggunakan algoritma genetika dengan proses mutasi *differential* Meiliya Cahya Yustina, 2024

DESAIN SISTEM PENJADWALAN MATA PELAJARAN MENGGUNAKAN ALGORITMA GENETIKA DI SMAN 2 PURWAKARTA

evolution, sehingga menghasilkan jadwal yang dihasilkan secara otomatis. Data penjadwalan yang telah dibuat disimpan dalam *database* lokal.

3.2.3 Desain Sistem

Gambar 3.3 menggambarkan diagram alir yang memberikan gambaran awal perancangan sistem penjadwalan mata kuliah dengan menggunakan Algoritma Genetika proses mutasi *Differential Evolution*.



Gambar 3.3 Diagram Alir Algoritma Genetika Mutasi *Differential Evolution*

Gambaran dari algoritma genetika dengan proses mutasi *differential evolution* dapat diuraikan sebagai berikut:

- UML adalah model pengembangan sistem.
- Admin mengatur parameter penelitian, termasuk data guru, kelas, mata pelajaran, dan penugasan guru.
- Inisialisasi populasi merupakan tahap terpenting dalam proses algoritma genetika. Dengan membuat individu secara acak untuk mendapatkan hasil yang terbaik. Hasil dari penelitian ini menentukan waktu dan lokasi yang tepat untuk kegiatan belajar mengajar guru dan siswa. Satu individu berisi hari, jam mengajar, kelas, dan kode guru.
- Hitung *Fitness* dengan menentukan nilai setiap individu berdasarkan jumlah benturan guru. Parameter nilai *fitness* mengharuskan tidak ada jadwal guru yang berbenturan pada jam yang sama. Rumus untuk menghitung *fitness* secara sistematis dituliskan seperti persamaan (3.1):

$$F = \frac{1}{1 + (\text{jumlah benturan guru})} \dots \dots \dots (3.1)$$

- e. Seleksi dapat menggunakan metode *roulette-wheel* untuk menghitung nilai probabilitas masing-masing individu. Secara sistematis untuk menghitung probabilitas dituliskan seperti persamaan (3.2):

$$\text{Probabilitas} = \frac{\text{nilai fitness}}{\text{total nilai fitness}} \dots \dots \dots (3.2)$$

Nilai probabilitas akan dijumlahkan dalam menghitung nilai kumulatif. Perhitungan nilai kumulatif akan terus berlanjut hingga semua individu dihitung. Secara sistematis untuk menghitung kumulatif dituliskan seperti berikut:

$$\text{Kumulatif} = \text{gabungan nilai probabilitas individu} \dots \dots \dots (3.3)$$

Roulette-wheel ditentukan dengan menghitung nilai probabilitas dan nilai kumulatif berdasarkan jumlah peserta. Kromosom dengan nilai *fitness* yang lebih tinggi ditempatkan di potongan lingkaran *roulette-wheel* yang lebih besar daripada kromosom dengan nilai *fitness* yang lebih rendah.

- f. *Crossover* terjadi ketika dua individu bertukar kromosom berdasarkan *crossover rate*. Dalam penelitian ini, *crossover rate* adalah 0.9. Nilai *crossover rate* telah dimasukkan ke dalam prosedur pengkodean. Sebuah titik potong diperlukan saat menukar kromosom. Penelitian ini menggunakan *one point crossover* sebagai titik potong. *One point crossover* adalah jenis perkawinan silang yang membagi dua bagian dengan satu titik potong. Pendekatan *crossover* penelitian ini menghasilkan angka acak antara 1 sampai dengan n (panjang kromosom dikurangi 1).
- g. Mutasi dalam proses *Differential Evolution* melibatkan pembuatan individu baru dari individu yang sudah ada dengan menggunakan rumus yang berbeda dengan Algoritma Genetika (secara acak). Jika rumus mutasi dalam *Differential Evolution* V adalah mutasi baru, F adalah faktor skala 0,5 (nilai yang dimasukkan selama proses pengkodean), dan X_{r1}, X_{r2}, X_{r3} adalah kromosom yang dimutasi. Rumus mutasi *Differential Evolution* dinyatakan sebagai berikut:
- $$V = X_{r1} + F \times (X_{r2} - X_{r3}) \dots \dots \dots (3.4)$$
- h. Terminasi adalah titik di mana proses berakhir. Jika iterasi selesai, seluruh populasi telah dianalisis dan jadwal optimal telah ditentukan berdasarkan nilai

fitness. Mekanisme ini juga mempengaruhi jumlah populasi dan generasi maksimum yang dimasukkan dalam sistem. Semua masukkan akan diproses hingga selesai, dan sistem akan menentukan jadwal yang optimal.

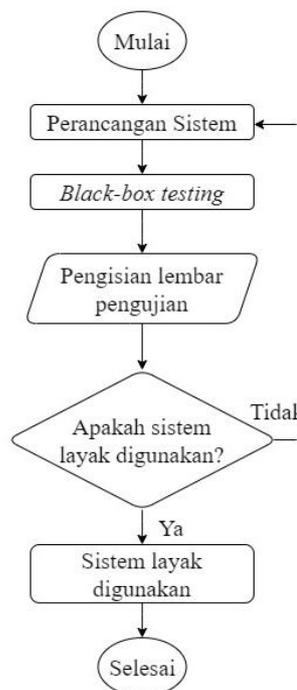
3.2.4 Pengkodean

PHP digunakan untuk pengkodean. *Framework* Laravel mendukung PHP, HTML, dan CSS, yang terhubung ke *database* dan berjalan di Laragon sebagai *server web* lokal. Pengkodean dilakukan di Visual Studio Code dengan konfigurasi Laravel. Proses Algoritma Genetika dengan mutasi *Differential Evolution* yang dihasilkan pada tahap desain sistem dikonversi ke dalam kode optimasi yang menghasilkan jadwal secara otomatis.

3.2.5 Pengujian

Pada penelitian ini menggunakan dua buah pengujian yaitu *black-box testing* dan SUS.

- a. *Black-box testing* dilakukan oleh penguji yang paham akan fungsi terkait *fullstack development*, namun tidak perlu memperhatikan mengenai detail sistem. Fungsi pengujian ini untuk memeriksa kelayakan sistem dalam penggunaannya. Pada Gambar 3.4 diuraikan mengenai prosedur dari pengujian *black-box*.



Gambar 3.4 Diagram Alir *Black-box Testing*

Meiliya Cahya Yustina, 2024

DESAIN SISTEM PENJADWALAN MATA PELAJARAN MENGGUNAKAN ALGORITMA GENETIKA DI SMAN 2 PURWAKARTA

Universitas Pendidikan Indonesia | repository.upi.edu | Perpustakaan.upi.edu

- b. Pengujian SUS dengan jumlah responden dalam pengisian kuesioner SUS diberikan pada 3 orang yang akan menggunakan sistem ini nantinya. Responden tersebut merupakan 1 kepala kurikulum dan 2 staf kurikulum untuk menguji *usability* pada sistem. Responden yang dipilih berkaitan dengan pembuatan jadwal. Pertanyaan dari pengujian ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Pertanyaan Pengujian SUS

No.	Pertanyaan
1.	Saya berpikir akan menggunakan sistem ini lagi
2.	Saya merasa sistem ini rumit untuk digunakan
3.	Saya merasa sistem ini mudah digunakan
4.	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini
5.	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya
6.	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini)
7.	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat
8.	Saya merasa sistem ini membingungkan
9.	Saya merasa tidak ada hambatan dalam menggunakan sistem ini
10.	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini

Skor dari setiap butir pertanyaan memiliki skala likert yang disediakan oleh SUS berdasarkan pada Tabel 3.2.

Tabel 3.2 Skala Likert Pengujian SUS

Skala Likert	Skor
Sangat Tidak Setuju	1
Tidak Setuju	2
Netral	3
Setuju	4
Sangat Setuju	5

Pedoman dalam menentukan skor kuesioner pengujian SUS yaitu sebagai berikut:

- 1) Kurangi satu skor pengguna untuk pertanyaan ganjil,
- 2) Kurangi lima skor pengguna untuk pertanyaan genap, dan
- 3) Kalikan jumlah dari skor pengguna dengan angka 2,5.

Untuk menentukan skor rata-rata, cari skor SUS masing-masing responden, kemudian jumlahkan seluruh skor pengguna dan bagi dengan jumlah total responden. Skor SUS ditentukan dengan menggunakan rumus berikut:

$$\bar{x} = \frac{\sum x}{n} \dots\dots\dots(3.5)$$

Keterangan:

\bar{x} = Skor rata-rata

$\sum x$ = Jumlah skor SUS

n = Jumlah responden

Setelah menghitung skor, parameter penilaian yaitu Skala Nilai dan Kategori yang dapat digunakan untuk menarik kesimpulan. Tabel 3.3 berisi informasi lebih lanjut mengenai hasil skor yang dicapai.

Tabel 3.3 Tabel Bobot Skor SUS

Total Skor SUS	Skala Nilai	Kategori
> 80.3	A	<i>Excellent</i>
68 – 80.3	B	<i>Good</i>
68	C	<i>Okay</i>
51-68	D	<i>Poor</i>
< 51	E	<i>Awful</i>

3.3 Karakteristik Objek Penelitian

Karakteristik objek dalam penelitian sistem penjadwalan mata pelajaran dengan algoritma genetika dan mutasi *differential evolution* dapat dijelaskan secara rinci pada Tabel 3.4. Tabel tersebut memuat informasi penting mengenai spesifikasi dan fitur-fitur yang digunakan dalam pengembangan sistem penjadwalan tersebut.

Tabel 3.4 Karakteristik Objek Penelitian

Karakteristik	Deskripsi
Jenis Sistem	Sistem penjadwalan berbasis perangkat lunak.
Tujuan	Mengoptimalkan jadwal mata pelajaran untuk siswa dan guru.
Fitur Utama	Algoritma genetika dengan proses mutasi <i>differential evolution</i> (Inialisasi populasi, hitung <i>fitness</i> , seleksi, <i>crossover</i> , mutasi dan iterasi).
Parameter	Mata pelajaran, kelas, guru, dan penugasan guru.
Data <i>Input</i>	Jumlah populasi, maksimal generasi.
<i>Output</i>	Jadwal mata pelajaran yang terbaik.
Batasan	Tidak boleh terjadi benturan jadwal antar guru dalam jam dan hari yang sama.