

BAB III

PERANCANGAN SISTEM

3.1 UML (*Unified Modelling Language*)

Dalam melakukan perancangan sistem mengacu pada pendekatan berorientasi objek sehingga digunakanlah UML (*Unified Modelling Language*). Berdasarkan komponen proses metode UML dibagi menjadi 6 fase (Utomo, 2003), yaitu :

1. Pemodelan Bisnis

Merupakan langkah untuk mengidentifikasi kemampuan sistem yang diinginkan oleh pengguna. Aplikasi yang diinginkan pengguna (guru) adalah aplikasi yang dapat menyusun jadwal secara otomatis. Kendala yang dialami selama ini adalah penyusunan jadwal biasanya dengan cara manual yaitu dengan mengisi satu persatu jadwal kelas yang kosong dengan mata pelajaran setelah itu baru guru yang mengampu.

2. *Requirement* (Kebutuhan)

Kebutuhan yang mendasar dalam membangun aplikasi ini adalah sebuah komputer *server* yang menampung data-data guru, mata pelajaran, dan jadwal. Komputer *server* ini juga digunakan sebagai mesin pengolah jadwal.

3. Analisis Perancangan

Merupakan langkah menguraikan cara untuk merealisasikan sistem yaitu dengan melakukan perancangan aplikasi. Algoritma yang digunakan untuk membangun sistem ini untuk sementara

adalah Algoritma Genetika, tidak menutup kemungkinan akan ada tambahan Algoritma lain untuk membangun aplikasi ini.

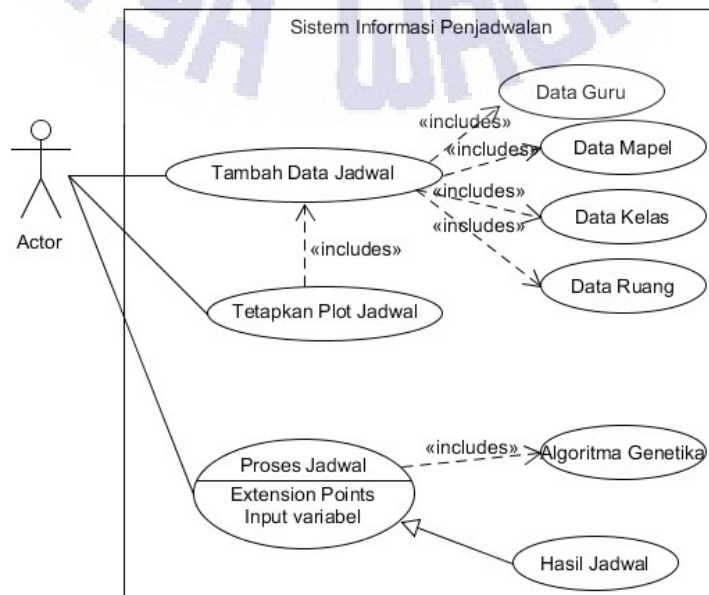
4. Implementasi

Merupakan langkah pembuatan program (kode-kode program) dengan menggunakan bahasa pemrograman tertentu. Pada tahap implementasi ini hal yang dilakukan pertama adalah mengumpulkan data guru yang mengampu. Kemudian data guru tersebut diplotkan ke kelas dimana guru tersebut mengajar. Barulah jadwal dibuat dengan menyusun data-data tersebut.

Akan tetapi masih ada kendala dalam langkah Mengolah jadwal, karena pada tahap tersebut membutuhkan alur yang panjang untuk menciptakan hasil jadwal jadi yang relevan dan tanpa kesalahan.

5. Pengujian

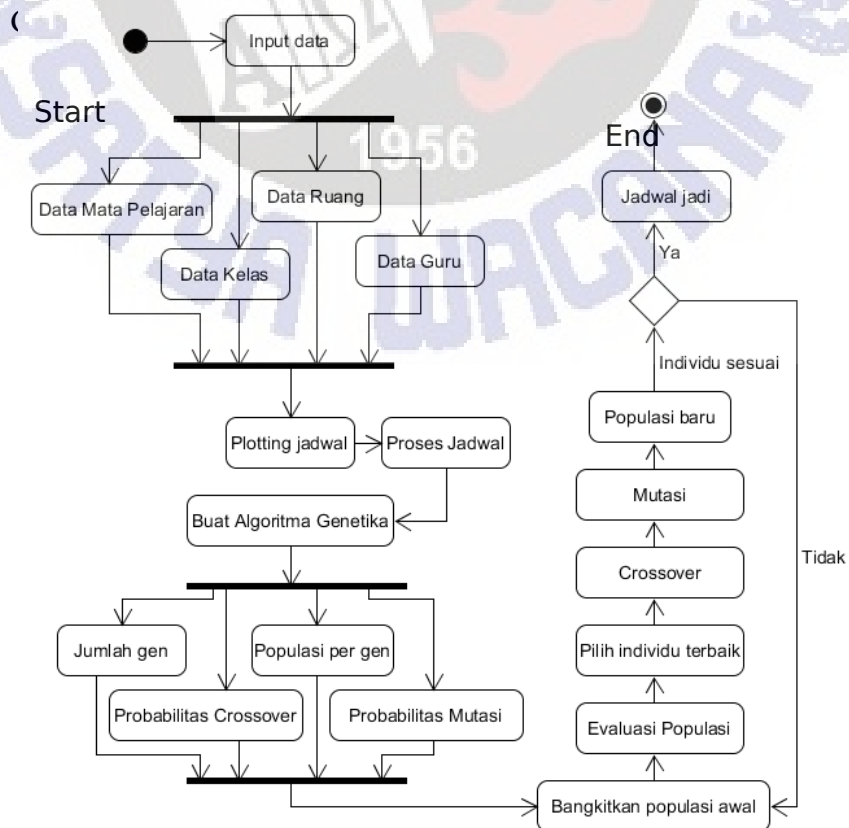
Langkah untuk melakukan verifikasi sistem secara keseluruhan. Untuk aplikasi pada tahap pengujian ini akan dilakukan dengan cara membuat simulasi jadwal sederhana yang langsung dapat dipakai. Kalau masih ada beberapa jadwal guru atau mata pelajaran yang bertabrakan, maka akan ada perbaikan di sisi



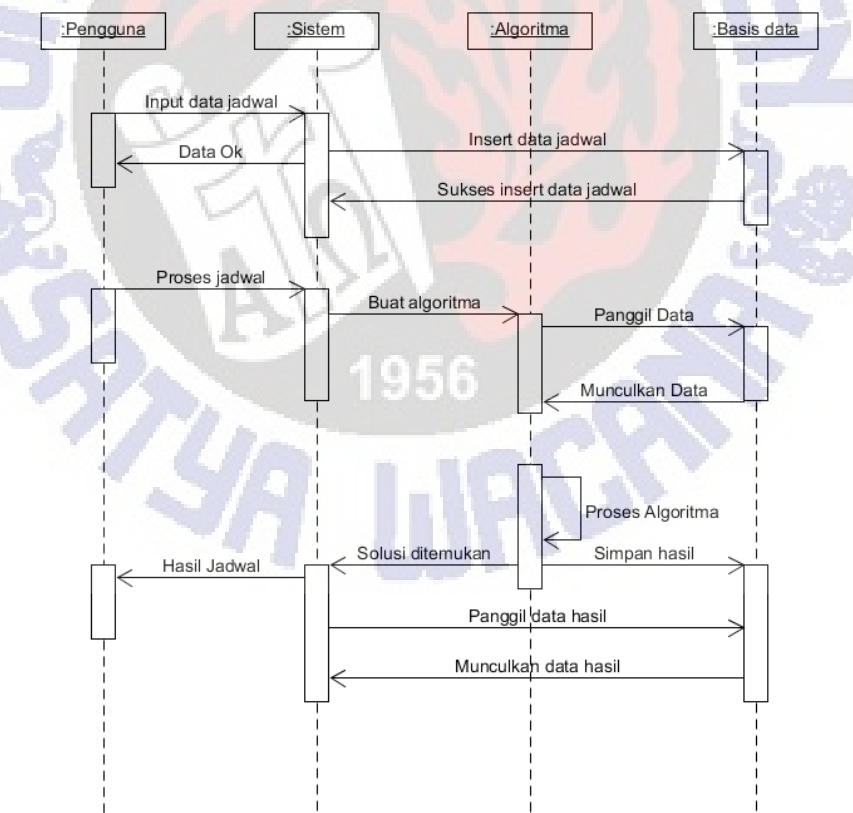
Penggun

Gambar 3.1. *Use Case Diagram Skema*

Dapat diketahui pada *Use Case Diagram* di atas bahwa hanya ada satu tokoh yaitu pengguna. Karena keseluruhan aplikasi menggunakan kecerdasan buatan sederhana yaitu Algoritma Genetika. Pengguna hanya diminta mengisi beberapa data dan kemudian menekan tombol sebagai pemicu untuk dimulainya proses pembuatan jadwal. Sistem kemudian akan menampilkan hasil yang telah diproses di dalam Algoritma



Aplikasi ini memiliki alur yang sederhana sebenarnya, akan tetapi alur tersebut sangatlah panjang terutama pada saat pemilihan populasi pada proses Algoritma Genetika. Dapat dilihat pada *Activity Diagram* di atas bahwa proses dari membangkitkan populasi awal sampai populasi baru terbentuk merupakan proses yang panjang, dan itu pun populasi baru yang terbentuk belum tentu memiliki individu yang sesuai dengan kriteria perhitungan. Dengan begitu populasi baru yang nantinya terseleksi adalah populasi yang benar-benar terpilih dari proses panjang pemilihan populasi.



Pengguna menekan tombol untuk memicu ke proses Algoritma, Sistem memanggil data yang dibutuhkan dari *Database* dan kemudian data-data tersebut diolah dalam Algoritma sampai

menghasilkan sebuah jadwal. Hasil jadi tersebut langsung ditampilkan langsung dari Sistem ke Pengguna.

3.2 Perancangan UI (*User Interface*)

3.2.1 Halaman Awal

Pada halaman awal ditampilkan nama untuk aplikasi yaitu *Skema*, serta ada beberapa *text label* yang berfungsi sebagai *button* yang berfungsi memanggil *form* lain.



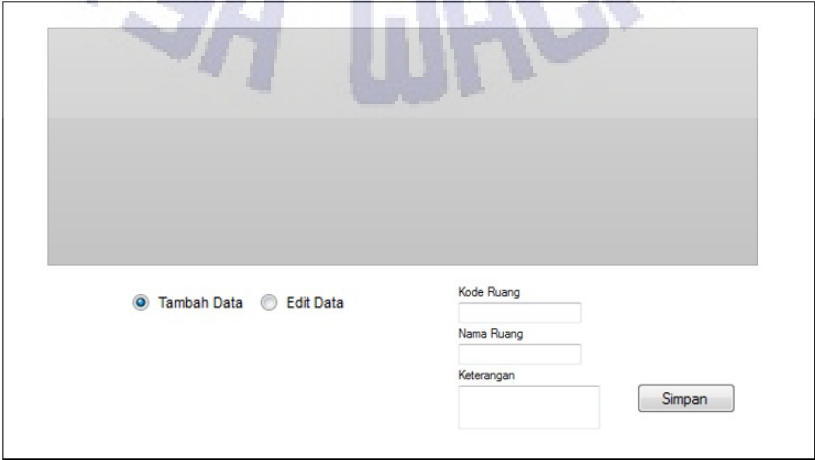
3.2.2 *Form* Mata Pelajaran

Pada *form* Mata Pelajaran ada empat buah tabulasi yang berisi untuk mengatur pembuatan jadwal.

Keterangan :

- Tabulasi sejumlah empat di atas untuk memilih menu yang diinginkan.
- *DataGridView* di sebelah kiri digunakan untuk menampilkan data dari *database*. Apabila salah satu baris diklik kanan maka akan muncul *ContextMenuStrip* dengan pilihan hapus berfungsi untuk menghapus satu baris data yang dipilih.
- *RadioButton* di sebelah kanan untuk memilih antara menambah data baru atau merubah data yang sudah ada. Untuk merubah data diharuskan memilih baris data yang akan dirubah terlebih dahulu di *DataGridView*.
- *TextBox* kode mapel, nama mapel, dan jam digunakan untuk mengisi data dan juga untuk menampung data yang akan dirubah.
- *Button* simpan sesuai namanya digunakan untuk menyimpan data baru atau menyimpan data yang akan dirubah.

3.2.3 **Form Ruang**



The screenshot shows a web form titled "Form Ruang". It features a large gray rectangular area at the top, likely for a table or list. Below this area, there are two radio buttons: "Tambah Data" (selected) and "Edit Data". To the right of these buttons are three text input fields labeled "Kode Ruang", "Nama Ruang", and "Keterangan". At the bottom right of the form is a "Simpan" (Save) button.

karena di dalam *form* Ruang hanya membutuhkan fungsi CRUD (*Create, Read, Update, and Delete*).

3.2.4 *Form Jadwal*

Genetika. *DataGridView* di kanan berfungsi untuk menampilkan hasil jadwal yang telah diproses.

3.3 Algoritma Genetika

3.3.1 Kondisi Awal

Aplikasi penjadwalan otomatis ini mempunyai kondisi dimana:

- Guru bisa mengampu lebih dari satu mata pelajaran di beberapa kelas.
- Satu ruang kelas bisa ditempati kelas yang berbeda.
- Menggunakan jadwal lima hari kerja.

Kemudian ada hal yang harus diperhatikan sebagai berikut:

- Satu ruangan atau laboratorium hanya bisa digunakan satu kelas saja, akan tetapi bengkel dan lapangan dapat digunakan oleh banyak kelas.

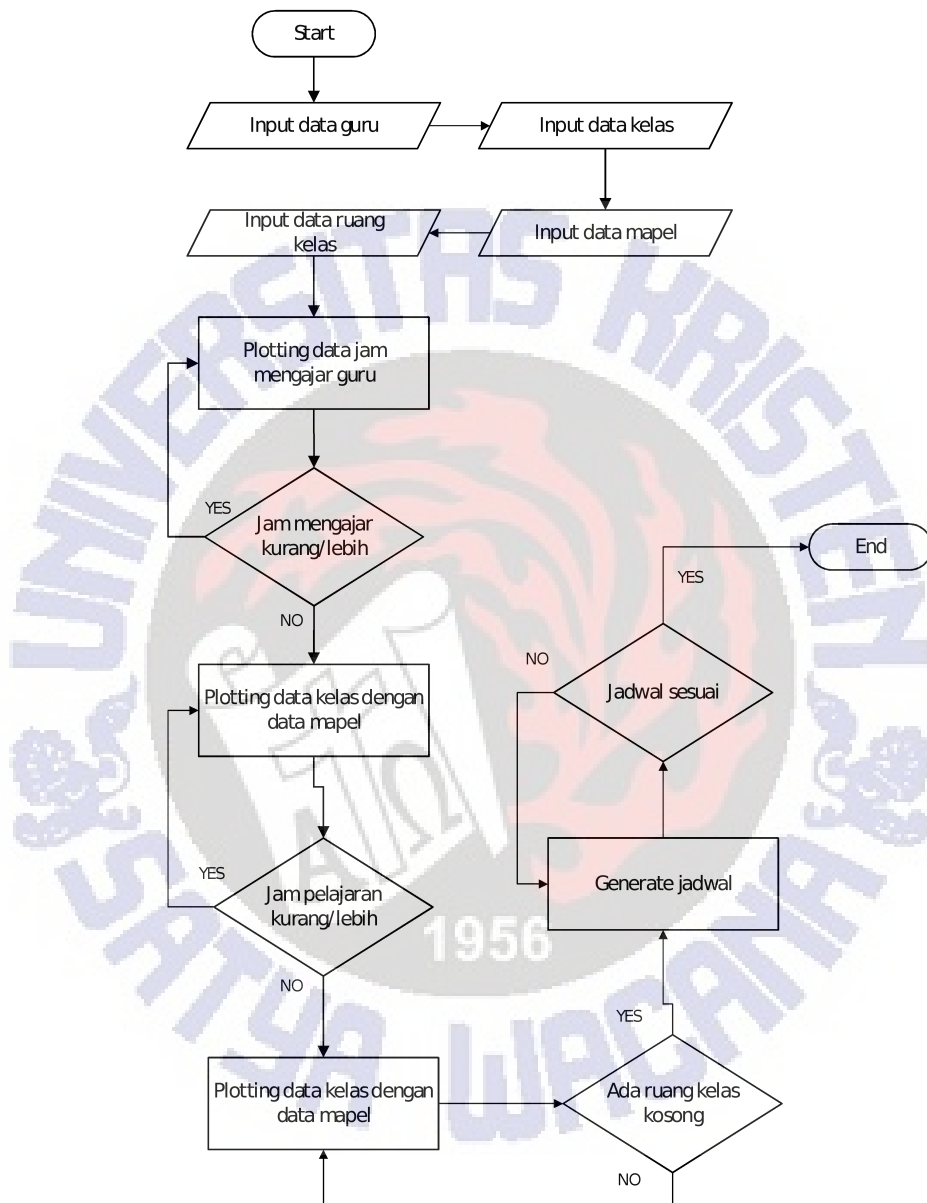
- b. Satu Guru tidak mungkin berada di sebuah jadwal untuk dua mata pelajaran atau satu mata pelajaran di dua kelas.
- c. Setiap kelas memiliki jam pelajaran dan mata pelajaran yang berbeda.

Dari beberapa analisis di atas didapat analisis data mentah yang digunakan untuk mendukung algoritma yang akan dipakai.

Adapun algoritma sederhana sebagai berikut :

Pseudo code 3.1. Garis besar aplikasi

```
//Pseudo code SKEMA  
Mulai  
Mengumpulkan data Kelas  
Mengumpulkan data mata pelajaran  
Mengumpulkan data guru pengampu  
Menentukan jumlah jam setiap kelas  
Membuat data jadwal kosong setiap kelas  
Membuat data mengajar guru setiap hari  
Memasukkan data mata pelajaran ke data kelas  
Mengolah jadwal  
Jadwal jadi  
Selesai
```

Gambar 3.8 Flowchart alur proses jadwal

3.3.2 Tahapan Algoritma Genetika

Proses genetika yang dilakukan pada tahapan penelitian

ini adalah sebagai berikut:

- a. Mengambil data yang dibutuhkan Algoritma Genetika.
- b. Memilih kromosom untuk dilakukan *crossover* dan mutasi, setelah itu hasilnya dimasukkan ke dalam populasi.
- c. Membuat jadwal setiap data kromosom baru dengan menggunakan proses seleksi sebelumnya.
- d. Menyeleksi nilai *fitness*, dan jika *fitness* belum didapat maka proses akan diulang kembali.
- e. Mengambil kromosom dengan nilai *fitness* terbaik sebagai solusi terbaik dan menjadikannya sebagai jadwal yang telah tersusun.

Hal-hal yang dilakukan pada proses inisialisasi Algoritma Genetika adalah sebagai berikut:

- a. Membuat daftar tabel data yang digunakan.
- b. Melakukan setting parameter Algoritma Genetika dari *database*.
- c. Mendata matakuliah yang ditawarkan dari database sebagai data kromosom.
- d. Membentuk kromosom sebagai populasi awal untuk penentuan *fitness*.

3.3.3 Proses Penjadwalan

Model Algoritma Genetika yang akan digunakan untuk melakukan optimasi adalah sebagai berikut:

- a. Seleksi

Pada seleksi, dilakukan penilaian atas nilai *fitness*. Akibatnya, *fitness* yang memiliki kualitas kromosom terbaik memiliki kesempatan pada generasi berikutnya. Seleksi yang digunakan adalah seleksi roulette wheel.

Dalam pelaksanaan seleksi ini perlu dipertimbangkan jumlah populasi agar populasi tidak terlalu banyak dan memakan waktu yang lama, dan populasi juga tidak terlalu sedikit yang akan mengakibatkan kemiripan kromosom.

b. *Crossover* (Kawin Silang)

Crossover yang digunakan adalah penyilangan satu titik dengan permutasi. Pemilihan kromosom ditentukan oleh probabilitas. Banyak gen yang ditukar tergantung pada penentuan parameter awal. Dalam melakukan *crossover*, setiap dua kromosom akan menghasilkan dua offspring yang baru sebagai gen terbaik.

c. Mutasi

Mutasi dilakukan setelah operasi *crossover* selesai. Teknik mutasi ini dilakukan dengan menukar gen secara random. Dalam proses ini perlu diperhatikan tingkat mutasi dan tingkat probabilitas terjadi mutasi. Jika mutasi terlalu besar kemungkinan hilangnya kromosom terbaik. Tetapi, jika mutasi terlalu sedikit, kromosom akan lama untuk menemukan solusi yang optimal.

d. Penentuan *Fitness*

Penentuan fitness pada dasarnya adalah pemberian nilai tersendiri yang menentukan tercapai tidaknya suatu proses algoritma genetika. Apa yang dilakukan proses ini adalah proses pembuatan jadwal sesuai kromosom yang dipilih dengan memprosesnya dengan menghitung seberapa dekat dengan nilai.