

Perancangan Aplikasi Penjadwalan Mata Kuliah (Studi Kasus : STMIK Provisi Semarang)

Rudy Hartadi¹, Arief Hidayat², Victor G Utomo³

^{1,3}Program Studi Teknik Informatika, ²Program Studi Sistem Informasi STMIK ProVisi Semarang
¹pg@rudhartadi.web.id, ²rifmillenia@gmail.com, ³victor.utomo@gmail.com

ABSTRAKSI - Jadwal mata kuliah adalah salah satu komponen penting dalam suksesnya pelaksanaan kegiatan belajar mengajar di perguruan tinggi, karena semua kegiatan dosen dan mahasiswa bergantung pada jadwal yang ada, sehingga jadwal harus disusun dengan benar pada awal semester. Penyusunan jadwal mata kuliah dengan cara manual menggunakan aplikasi *spreadsheet*, seperti yang dilakukan di STMIK ProVisi, membutuhkan ketelitian yang tinggi, karena harus memperhatikan kendala-kendala dalam penjadwalan (ruangan, dosen, kelompok mahasiswa, hari, dan waktu) agar tidak saling bertumbukan. Permasalahan dalam penyusunan jadwal dapat diselesaikan menggunakan aplikasi penjadwalan mata kuliah yang mampu membangkitkan jadwal secara otomatis menggunakan algoritma genetik serta menggunakan metode pengeditan jadwal tarik-lepas (*drag and drop*) melalui GUI (*Graphical User Interface*). Aplikasi penjadwalan yang dihasilkan dalam penelitian telah berhasil diuji coba untuk menjadwalkan mata kuliah pada program studi Teknik Informatika dan Sistem Informasi STMIK ProVisi Semarang.

Kata Kunci: *Algoritma Genetik, Drag and Drop, Penjadwalan Mata Kuliah.*

ABSTRACT - Schedule of courses is one important component in the successful implementation of teaching and learning in higher education, because all the activities of lecturers and students rely on the existing schedule, so the schedule should be drawn up correctly at the start of the semester. Course scheduling by hand using a spreadsheet application, such as is done in STMIK ProVisi, requiring high accuracy, because it must consider the scheduling constraints (room, lecturer, student groups, day, and time) in order not to collide with each other. Problems in scheduling can be solved using the course scheduling application that is able to automatically generate schedules using genetic algorithms and editing schedule using drag and drop method through the GUI (Graphical User Interface). The resulting scheduling applications in the study had been successfully tested to schedule courses in the department of Informatics Engineering and Information Systems STMIK ProVisi Semarang.

Keywords: *Genetic algorithms, Drag and Drop, Course Scheduling.*

1. Pendahuluan

Pembuatan jadwal kuliah adalah kegiatan rutin yang dilakukan oleh perguruan tinggi pada tiap tahun ajaran. Di perguruan tinggi, program penjadwalan merupakan salah satu hal penting dalam proses belajar mengajar, karena semua kegiatan dosen dan mahasiswa bergantung pada jadwal yang ada, sehingga harus disusun dengan benar dan diperbaiki pada awal tahun akademik, sehingga nantinya tidak mengganggu aktifitas belajar mengajar antar dosen dan mahasiswa (Ariani et al,2011:1). Pembuatan jadwal ini membutuhkan banyak waktu untuk menghasilkan jadwal yang baik dan adil (Jenal et al,2011:1).

Terdapat beberapa aspek yang harus diperhatikan dalam pembuatan jadwal, yaitu: tidak boleh adanya jadwal yang bertumbukan, misalnya ada seorang dosen yang dijadwalkan mengajar 2 kelas berbeda pada saat yang bersamaan atau mahasiswa dijadwalkan kuliah di 2 mata kuliah yang berlainan pada jam yang sama, ketersediaan waktu mengajar bagi dosen tidak tetap, keterbatasan ruangan, dan

distribusi mata kuliah yang merata untuk tiap minggunya.

Selain permasalahan-permasalahan tersebut, STMIK ProVisi Semarang mempunyai permasalahan unik lainnya, yaitu: adanya kelas pagi, sore, dan karyawan. Sebagian besar mahasiswa yang mengambil kelas sore adalah mereka yang bekerja di pagi hari, sehingga jadwal kelas sore tidak mungkin untuk dipindah ke kelas pagi. Sedangkan mahasiswa kelas pagi, yang sebagian besar belum bekerja, dapat dijadwalkan untuk mengikuti kelas di sore hari. Kelas karyawan diadakan di akhir pekan (Jum'at – Minggu), dan jadwal tidak mungkin untuk dipindahkan di hari lain.

Pembuatan jadwal kuliah di STMIK ProVisi Semarang dilakukan oleh ketua program studi. Pembuatan jadwal tersebut masih menggunakan cara manual dengan menggunakan perangkat lunak Microsoft Excel yang rentan akan kesalahan. Hal ini dibuktikan dengan revisi, sebelum rilis, yang harus dilakukan hingga 2 sampai 3 kali akibat kesalahan penyusunan jadwal. Masalah utama

yang ditemui dengan cara manual adalah sulit dalam melakukan pengeditan jadwal, karena harus melakukan *cut-copy-paste* data dari *cell* asal ke *cell* tujuan. Masalah lain adalah harus dilakukan pemeriksaan ulang antara 2 kaprodi agar jadwal dosen yang mengajar di program studi Teknik Informatika dan Sistem Informasi tidak bertumbukan.

2.1. Penjadwalan

Tugas penjadwalan tradisional berkaitan dengan pengalokasian yang tepat untuk sejumlah aktivitas ke sumber daya (atau sumber daya ke aktivitas) dari waktu ke waktu dengan memperhatikan prioritas, durasi, kapasitas, dan kendala-kendala ketidakcocokan (Bartak,1999:5). Menurut Pinedo (1995:1), penjadwalan adalah proses pengambilan keputusan yang berkenaan dengan pengalokasian sumber daya terbatas untuk tugas-tugas dari waktu ke waktu yang memiliki tujuan untuk mengoptimasi dari satu atau lebih tujuan.

Setiap tugas mungkin memiliki tingkat prioritas tertentu, kecepatan waktu mulai tertentu dan tanggal jatuh tempo tertentu. Tujuan juga dapat berupa berbagai macam bentuk. Tujuan penjadwalan organisasi yang satu mungkin untuk meminimalkan waktu yang dibutuhkan untuk menyelesaikan suatu tugas, sedang tujuan penjadwalan organisasi yang lain adalah untuk mengurangi jumlah tugas yang selesai setelah tanggal jatuh tempo. (Pinedo,1995:1)

2.2. Penjadwalan Mata Kuliah

Penjadwalan mata kuliah adalah kegiatan administratif yang paling utama di sebagian besar universitas. Penjadwalan mata kuliah dapat diklasifikasikan ke dalam dua kategori utama: jadwal kuliah dan jadwal ujian (Petrovic et al,2004:2).

Ada perbedaan yang signifikan antara jadwal kuliah dan ujian, yaitu: ada tepat satu ujian untuk setiap mata kuliah, sementara tiap mata kuliah mungkin perlu diadakan beberapa kali per minggu. Contoh lain yang menggambarkan perbedaan antara jadwal kuliah dan ujian diperoleh dengan mengamati bahwa ujian dapat dijadwalkan ke dalam satu ruangan maupun dibagi ke beberapa ruangan. Hal ini tentu tidak dapat diterima dalam penjadwalan mata kuliah, karena satu mata kuliah harus dijadwalkan persis ke dalam satu ruangan saja (Petrovic et al,2004:3).

Kendala dalam pembuatan jadwal yang valid melibatkan penjadwalan kelas, guru, dan

ruangan ke dalam sejumlah periode yang tetap, sedemikian rupa sehingga tidak ada guru, kelas, atau ruangan yang digunakan lebih dari sekali per periode. Kombinasi tertentu dari seorang guru, mata kuliah, ruangan, dan kelas disebut tuple. Tuple mungkin diperlukan lebih dari satu kali per minggu. Dengan demikian, masalah penjadwalan dapat diutarakan sebagai penjadwalan sejumlah tuple sehingga seorang guru, kelas atau ruang tidak muncul lebih dari sekali per periode (Abramson et al,1992:2).

Kendala dalam penjadwalan dapat dibedakan ke dua jenis, yaitu:

1. *Hard Constraints*, merupakan kendala-kendala yang harus dipatuhi dalam penjadwalan. Solusi yang diajukan tidak boleh melanggar *hard constraints*. *Hard constraint* yang umum dalam penjadwalan mata kuliah adalah sebagai berikut:
 - a. Seorang dosen atau mahasiswa hanya dapat memberi atau mengikuti kuliah untuk satu lokasi pada waktu tertentu.
 - b. Jumlah total sumber daya (dosen dan kelas) yang dibutuhkan dalam satu periode tertentu tidak lebih besar dari jumlah sumber daya (dosen dan kelas) yang tersedia.
2. *Soft Constraints* merupakan batasan-batasan dalam penjadwalan yang apabila dilanggar masih dapat menghasilkan solusi yang layak, tetapi sedapat mungkin harus dipenuhi. Kenyataannya, masalah penjadwalan mata kuliah biasanya tidak mungkin untuk memenuhi semua *soft constraints*. Kualitas jadwal dapat dinilai berdasarkan seberapa banyak *soft constraints* dapat dipenuhi. (Petrovic et al,2004:2-3)

2.3. Algoritma Genetik

1. Pengertian Algoritma Genetik

Algoritma genetik adalah algoritma yang berusaha menerapkan pemahaman mengenai evolusi alamiah pada tugas-tugas pemecahan masalah. Pendekatan yang diambil oleh algoritma ini adalah dengan menggabungkan secara acak berbagai pilihan solusi terbaik di dalam suatu kumpulan untuk mendapatkan generasi solusi terbaik berikutnya yaitu pada suatu kondisi yang memaksimalkan kecocokannya atau lazim disebut *fitness*. Generasi ini akan merepresentasikan perbaikan-perbaikan pada populasi awalnya. Dengan melakukan proses ini secara berulang, algoritma ini diharapkan dapat menyimulasikan proses evolusioner. Pada akhirnya, akan

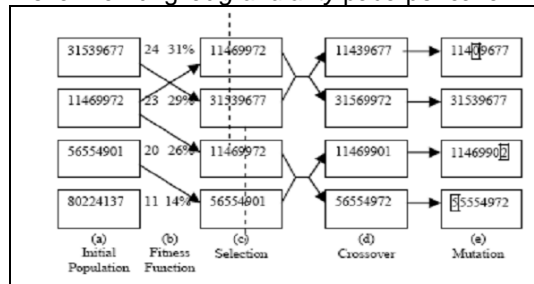
didapatkan solusi-solusi yang paling tepat bagi permasalahan yang dihadapi.

Untuk menggunakan algoritma genetik, solusi permasalahan direpresentasikan dengan menggunakan kromosom. Tiga aspek yang penting untuk penggunaan algoritma genetik adalah definisi *fitness function*, definisi dan implementasi representasi genetik serta definisi dan implementasi operasi genetik. Jika ketiga aspek diatas telah didefinisikan, algoritma genetik akan bekerja dengan baik.

2. Teknik Penggunaan Algoritma Genetik

Algoritma genetik dimulai dengan sekumpulan himpunan status yang dipilih secara acak, yang disebut populasi. Algoritma ini mengkombinasikan dua populasi induk. Setiap status atau individual direpresentasikan sebagai sebuah *string*.

Pada mutasi, tiap lokasi menjadi sasaran mutasi acak, dengan probabilitas independen yang kecil. Sebuah digit dimutasikan pada anak pertama, ketiga dan keempat. Algoritma genetik mengkombinasikan suatu kecenderungan menaik dengan eksplorasi acak di antara *thread* pencarian paralel. Keuntungan utamanya datang dari operasi *crossover*. Namun secara matematis dapat ditunjukkan bahwa bila posisi dari kode genetik dipermutasikan di awal dengan urutan acak, *crossover* tidak memberikan keunggulan. Secara intuisi, keuntungannya didapat dari kemampuan *crossover* untuk menggabungkan blok-blok huruf berukuran besar yang telah berevolusi secara independen untuk melakukan fungsi yang bermanfaat sehingga dapat menaikkan tingkat *granularity* pada pencarian.



Gambar 1. Contoh Penggunaan Algoritma Genetik (Sumber : Nugroho, 2008)

Teori dari algoritma genetik menjelaskan cara kerjanya menggunakan ide dari suatu *schema*, suatu *sub-string* di mana beberapa posisi tidak disebutkan. Dapat ditunjukkan bahwa, bila *fitness* rata-rata dari *schema* berada di bawah rerata, maka jumlah instansiasi dari *schema* di dalam populasi akan bertambah seiring bertambahnya waktu. Jelas sekali bahwa efek

ini tidak akan signifikan bila bit-bit yang bersebelahan sama sekali tidak berhubungan satu sama lain, karena akan ada beberapa blok kontinyu yang memberikan keuntungan yang konsisten. Algoritma genetik paling efektif dipakai bila setiap *schema* berkorespondensi menjadi komponen berarti dari sebuah solusi. Sebuah komponen yang baik cenderung bekerja dengan baik pada rancangan yang berbeda. Ini menunjukkan bahwa penggunaan algoritma genetik yang benar memerlukan rekayasa yang baik pada representasinya (Nugraha,2008:1-2).

Algoritma genetik dipilih karena kemudahan dalam pengaplikasian algoritma ke dalam sebuah solusi untuk mengatasi masalah penjadwalan mata kuliah.

2.4. Metode System Development Life Cycle (SDLC)

Metodologi pengembangan sistem informasi berarti suatu metode yang digunakan untuk melakukan pengembangan sistem informasi berbasis komputer. Metode yang paling umum digunakan adalah dengan siklus hidup pengembangan sistem (*System Development Life Cycle - SDLC*). SDLC merupakan metodologi klasik yang digunakan untuk mengembangkan, memelihara dan menggunakan sistem informasi. Metode ini menggunakan pendekatan sistem yang disebut pendekatan air terjun (*waterfall approach*), yang menggunakan beberapa tahapan dalam mengembangkan sistem (Supriyanto, 2007: 271). Berikut tahap – tahap pengembangan sistem dengan metode SDLC menurut Supriyanto (2007:271-275) :

1. Tahap Perencanaan Sistem (*System Planning*). Tahap perencanaan adalah tahap awal pengembangan sistem yang mendefinisikan perkiraan kebutuhan-kebutuhan sumber daya seperti perangkat fisik, manusia, metode (teknik dan operasi), dan anggaran yang sifatnya masih umum (belum detail/rinci).
2. Tahap Analisis Sistem (*System Analysis*). Tahap analisis sistem adalah tahap penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau diperbarui.
3. Tahap Perancangan / Desain Sistem (*System Design*). Tahap desain sistem adalah tahap setelah analisis sistem yang menentukan proses dan data yang diperlukan oleh sistem baru. Desain sistem dibedakan menjadi dua macam, yaitu desain sistem umum dan desain sistem terinci.

4. Tahap Penerapan / Implementasi Sistem (*System Implementation*). Tahap implementasi atau penerapan adalah tahap dimana desain sistem dibentuk menjadi suatu kode (program) yang siap untuk dioperasikan.
5. Tahap Pemeliharaan / Perawatan Sistem. Tahap pemeliharaan / perawatan sistem merupakan tahap yang dilakukan setelah tahap implementasi yang meliputi penggunaan sistem, audit sistem, penjagaan sistem, perbaikan sistem dan peningkatan sistem.

2.5. Penelitian Terdahulu

Penelitian tentang penjadwalan mata kuliah pernah dilakukan oleh Dian Ariani (Politeknik Elektronika Negeri Surabaya / PENS) pada tahun 2011, hasil dari penelitian tersebut adalah Aplikasi Penjadwalan Mata Kuliah menggunakan Algoritma *Particle Swarm Optimization* (PSO) yang diimplementasikan pada jurusan Teknik Informatika PENS. Ivan Nugraha (ITB) juga pernah melakukan penelitian serupa pada tahun 2008 yang menghasilkan contoh pengaplikasian Algoritma Genetik menggunakan perangkat lunak LabView untuk memberi solusi masalah penjadwalan pada jurusan elektro UNIKOM semester I tahun ajaran 2005/2006.

Penelitian yang penulis kerjakan sekarang selain menggunakan metode penyusunan jadwal secara otomatis, juga menggunakan metode tarik-lepas (*drag and drop*) melalui GUI (*Graphical User Interface*). Penggunaan metode *drag and drop* diharapkan dapat lebih memudahkan dalam melakukan perubahan jadwal, karena dapat memberikan tampilan menyeluruh mengenai jadwal mata kuliah yang ada dalam bentuk matrik.

2. Metode Pengembangan Sistem

Aplikasi Penjadwalan Mata Kuliah dirancang menggunakan metode pengembangan SDLC dengan adapun tahapan yang dilakukan meliputi: tahap perencanaan sistem, tahap analisis sistem, tahap perancangan sistem, tahap penerapan sistem, sedang tahap pemeliharaan sistem tidak dilakukan.

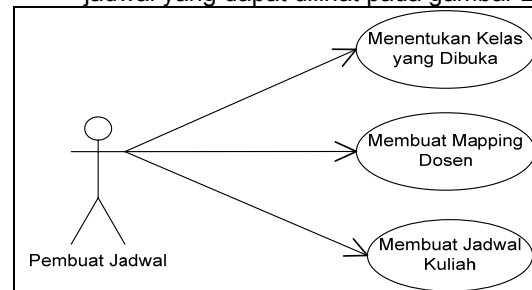
1. Kegiatan pada tahap perencanaan sistem adalah mengidentifikasi permasalahan yang dialami oleh Kaprogdi, yaitu : kesulitan dalam menyusun jadwal menggunakan aplikasi *spreadsheet* dan penggunaan data yang belum terintegrasi sehingga

membutuhkan validasi jadwal antar program studi.

2. Pada tahapan Analisa Sistem akan dianalisa lebih lanjut cara kerja sistem penjadwalan yang telah ada selama ini. Analisa akan dilakukan dengan menggambarkan cara kerja sistem lama dengan menggunakan diagram *use case* dan diagram aktivitas.

a. Diagram Use Case Sistem Lama

Diagram *use case* menggambarkan kegiatan yang dilakukan oleh pembuat jadwal yang dapat dilihat pada gambar 2.

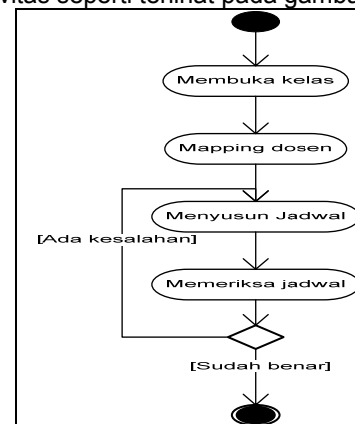


Gambar 2. Use Case pada Sistem Lama

Pada sistem lama, tahap pertama pembuatan jadwal adalah menentukan kelas yang akan dibuka. Setelah mengetahui kelas yang dibuka, dilakukan pemetaan untuk menentukan dosen pengampu untuk setiap kelas. Langkah selanjutnya adalah menempatkan kelas pada waktu dan ruang yang tepat. File yang digunakan untuk menyimpan mapping dosen dan jadwal kuliah terdiri dari dua file yang terpisah sehingga pemanfaatan pemanfaatan dosen untuk jadwal kuliah dilakukan secara manual.

b. Diagram Aktivitas Sistem Lama

Rincian lebih lanjut tentang pembuatan jadwal dijelaskan dengan diagram aktivitas seperti terlihat pada gambar 3.



Gambar 3. Diagram Aktivitas Sistem Lama

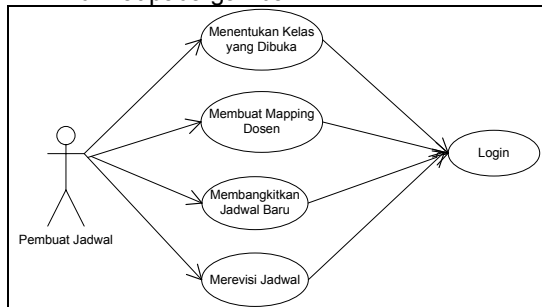
Berikut adalah keterangan lebih lanjut dari gambar 3.

- 1) Kelas yang akan dibuka pada suatu semester ditentukan terlebih dahulu. Kelas yang akan dibuka ditentukan oleh mata kuliah yang dilaksanakan pada semester tersebut dan jumlah mahasiswa tiap angkatan.
- 2) Pemetaan dosen menentukan dosen pengampu untuk masing-masing kelas.
- 3) Menyusun jadwal untuk setiap kelas.
- 4) Memeriksa ulang apakah ada jadwal yang salah. Jika ada kesalahan maka jadwal disusun ulang. Jika tidak ada lagi kesalahan, maka penyusunan jadwal telah selesai.

Berdasarkan analisa dari sistem lama dapat diambil beberapa kesimpulan

- 1) Pada sistem lama, pemetaan dosen yang terpisah dengan jadwal menimbulkan kemungkinan kesalahan pada proses penyusunan jadwal karena keliru dalam melihat pemetaan.
- 2) Kesalahan jadwal, baik karena kekeliruan melihat pemetaan atau kesalahan ruang/waktu harus diperiksa secara manual. Proses pemeriksaan ini pun masih rentan dengan kesalahan.
3. Tahap perancangan sistem, kegiatan yang dilakukan yaitu membuat *use case* sistem baru, diagram aktivitas, diagram kelas, rancangan basis data, rancangan antar muka dan rancangan penerapan algoritma genetik.

a. Diagram Use Case Sistem Baru. Pada sistem baru, alur proses penjadwalan berkembang dengan adanya penambahan fitur penjadwalan otomatis dari aplikasi Penambahan proses dapat dilihat pada gambar 4.

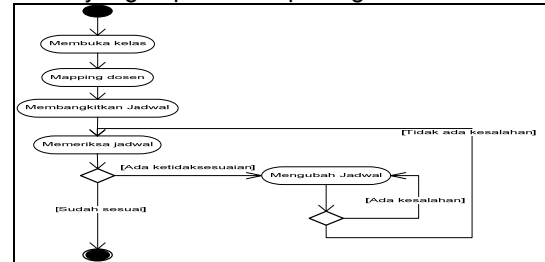


Gambar 4. Use Case Sistem Baru

Dua langkah awal pada sistem baru mirip dengan sistem lama yaitu penentuan kelas dan

pemetaan dosen. Perbedaannya, hasil penentuan kelas dan pemetaan dosen disimpan pada aplikasi penjadwalan. Langkah berikutnya pada sistem baru adalah pembangkitan jadwal baru. Pada langkah ini jadwal akan dibangkitkan secara otomatis dengan bantuan algoritma genetik. Setelah jadwal baru terbentuk, revisi tambahan mungkin harus dilakukan jika ternyata jadwal baru masih belum memenuhi syarat.

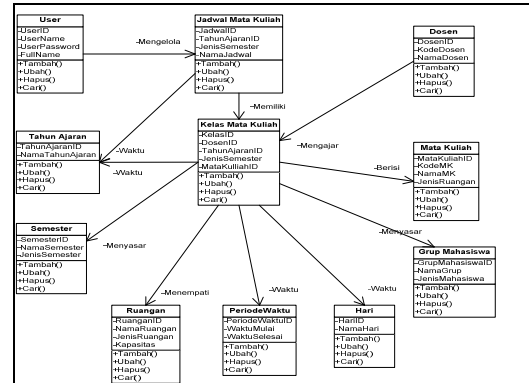
- b. Diagram Aktivitas Sistem Baru. Sistem baru yang dirancang dijelaskan secara lebih terperinci dengan diagram aktivitas yang dapat dilihat pada gambar 5.



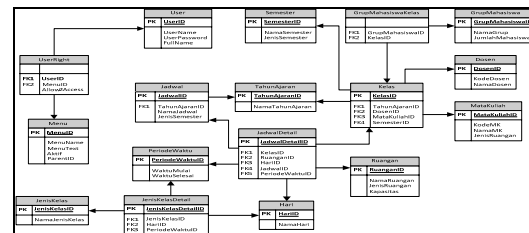
Gambar 5. Diagram Aktivitas Sistem Baru

Proses yang terdapat pada diagram aktivitas sistem baru dapat dijelaskan sebagai berikut:

- 1) Membuka kelas untuk semester bersangkutan.
- 2) Melakukan pemetaan dosen pengampu terhadap kelas yang dibuka.

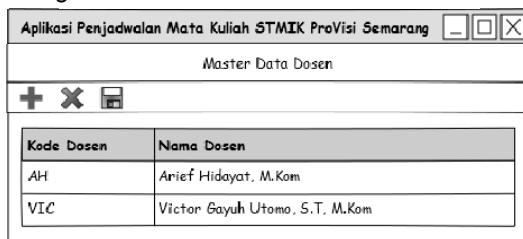


Gambar 6. Diagram Kelas Sistem Baru



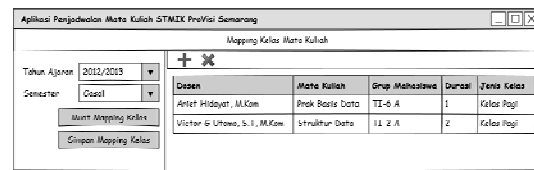
Gambar 7. Relasi Antar Tabel

- 3) Membangkitkan jadwal secara otomatis dengan bantuan program.
 - 4) Memeriksa jadwal untuk memeriksa ketidaksesuaian jadwal dengan kesediaan dosen. Jika sudah sesuai maka proses penjadwalan selesai. Jika belum sesuai maka perlu dilakukan penyesuaian dengan mengubah jadwal.
 - 5) Setelah diubah, jadwal diperiksa kesalahannya terkait tumbukan ruang/waktu, dosen dan grup mahasiswa. Jika ternyata terdapat kesalahan maka perubahan ditolak dan harus diulang hingga tidak terdapat kesalahan. Jika ternyata tidak terdapat kesalahan maka kembali diperiksa kesesuaian jadwal (proses 4).
- c. Diagram Kelas. Tahapan berikutnya pada perancangan sistem adalah merancang diagram kelas, yang dapat dilihat pada gambar 6.
 - d. Rancangan Tabel Basis Data. Rancangan basis data diwujudkan setelah melihat pemodelan pada diagram kelas. Rancangan basis data diwujudkan dalam bentuk tabel dan relasi antar tabel. Gambar 7 merupakan hubungan antar tabel aplikasi penjadwalan.
 - e. Rancangan antarmuka. Rancangan antar muka merupakan sketsa antar muka form-form yang terdapat dalam aplikasi penjadwalan mata kuliah. Rancangan antar muka Form Master Data Dosen digunakan untuk mengelola data dosen yang mengajar di STMIK ProVisi, yang ditunjukkan pada gambar 8.



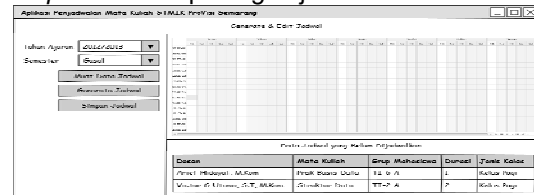
Gambar 8. Perancangan Form Master Data Dosen

Rancangan antar muka Form Mapping Kelas Mata Kuliah digunakan untuk mengelola data kelas mata kuliah pada tahun ajaran dan semester tertentu. Mapping Kelas Mata Kuliah merupakan relasi antara mata kuliah, dosen yang mengajar mata kuliah, dan grup mahasiswa yang mengikuti mata kuliah.



Gambar 9 Perancangan Form Mapping Kelas Mata Kuliah

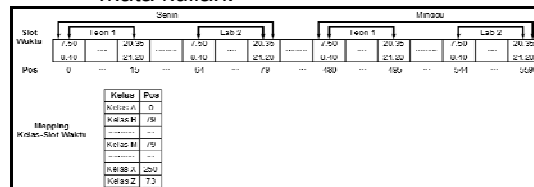
Rancangan antar muka Form Generate dan Edit Jadwal digunakan untuk membangkitkan jadwal serta melakukan pengeditan jadwal. Jadwal kelas mata kuliah yang telah memenuhi kriteria akan ditampilkan dalam *grid* jadwal (bagian kanan atas), sedang data jadwal kelas yang belum memenuhi kriteria akan ditampilkan di tabel kanan bawah. Proses pengeditan jadwal dapat dilakukan dengan cara *drag and drop* data kelas pada *grid* jadwal.



Gambar 10 Perancangan Form Generate & Edit Jadwal

- f. Rancangan Penerapan Algoritma Genetik. Siklus algoritma genetik adalah menentukan representasi individu dalam populasi, fungsi kecocokan (*fitness function*), melakukan seleksi terhadap individu dalam populasi, melakukan perkawinan silang (*crossover*), dan melakukan mutasi. Berikut adalah detail penerapan dari tiap tahapan siklus algoritma genetik untuk mencari solusi masalah penjadwalan mata kuliah:

- 1) Representasi Individu / Kromosom. Individu dalam penelitian ini merupakan representasi dari solusi jadwal mata kuliah. Jadwal direpresentasikan sebagai slot waktu yang menyimpan informasi data kelas. Gambar 10 menggambarkan slot waktu yang digunakan dalam representasi jadwal mata kuliah.



Gambar 10. Representasi Jadwal ke dalam Slot Waktu

Rumus untuk menentukan panjang slot waktu yang dibutuhkan yaitu:

$$\sum \text{Slot Waktu} = \sum \text{hari} \times \sum \text{ruangan} \times \sum \text{periode waktu}$$

Pada gambar 10, jadwal untuk hari Senin, ruangan Teori 1, dan periode waktu 07.50 - 8.40 berada pada posisi slot waktu 0. Jadwal untuk hari Senin, ruangan Lab 2, dan periode waktu 07.50 - 8.40 berada pada posisi slot waktu 64. Informasi mengenai lokasi suatu kelas dalam slot waktu disimpan dalam tabel pemetaan kelas – slot waktu untuk memudahkan dalam pencarian posisi.

Algoritma genetik menggunakan minimal 2 individu dalam mencari solusi terbaik. Kumpulan individu disebut sebagai populasi. Jumlah individu dalam populasi yang digunakan dalam penelitian ini adalah 100.

- 2) Representasi Fungsi Kecocokan (*Fitness Function*). Range nilai kecocokan untuk tiap kromosom adalah 0 sampai 1. Nilai kecocokan untuk tiap kromosom dihitung dengan rumus:

$$\text{fitness} = \frac{\sum \text{nilai kelas}}{\text{nilai maksimum}}$$

Nilai untuk tiap kelas mata kuliah ditentukan berdasarkan 5 kriteria, yaitu:

- a) Kelas mata kuliah harus dialokasikan pada ruangan yang kosong (tidak digunakan oleh kelas mata kuliah lain).
- b) Kelas mata kuliah yang membutuhkan komputer harus dialokasikan pada ruangan jenis lab.
- c) Kelas mata kuliah harus dialokasikan pada ruangan dengan kapasitas yang cukup untuk menampung jumlah mahasiswa dari kelas mata kuliah tersebut.
- d) Dosen hanya dapat mengajar pada satu kelas mata kuliah dalam satu waktu.
- e) Grup mahasiswa hanya dapat mengikuti satu kelas mata kuliah dalam satu waktu.

Nilai awal dari suatu kelas adalah 0. Tiap kriteria yang terpenuhi akan menambah nilai dari kelas, sehingga tiap kelas dapat memiliki nilai maksimum 5. Nilai maksimum yang digunakan sebagai pembagi dalam menentukan nilai *fitness* adalah $\sum \text{kelas} \times 5$.

- 3) Proses Seleksi. Proses seleksi dilakukan menggunakan metode *Elite Selection*. Cara kerja metode *elite selection* yaitu:

- a) Mengurutkan individu berdasarkan nilai *fitness*-nya.
- b) Beberapa individu dengan nilai terkecil kemudian dihapus dari populasi. Jumlah individu yang dihapus dalam penelitian ini didapat dari operasi pembangkitan bilangan acak.
- c) Individu yang dihapus kemudian diganti dengan individu baru.

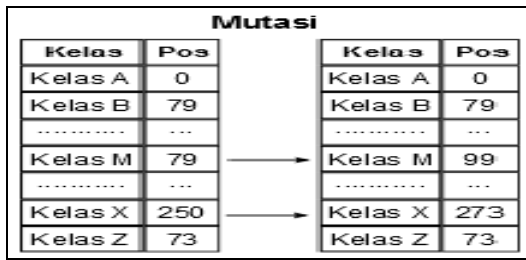
- 4) Persilangan (*Crossover*). Persilangan dilakukan dengan cara mengombinasikan data posisi kelas antara 2 individu/jadwal. Jumlah kelas yang mengalami persilangan adalah 2. Posisi kelas yang mengalami persilangan dipilih secara acak. Gambar 11 mengilustrasikan proses persilangan antara jadwal 1 dan jadwal 5.

Crossover				Hasil Crossover			
Kelas	Pos	Kelas	Pos	Kelas	Pos	Kelas	Pos
Kelas A	0	Kelas A	10	Kelas A	0	Kelas A	10
Kelas B	79	Kelas B	9	Kelas B	79	Kelas B	9
.....
Kelas M	79	Kelas M	63	Kelas M	63	Kelas M	79
.....
Kelas X	250	Kelas X	45	Kelas X	45	Kelas X	250
Kelas Z	73	Kelas Z	82	Kelas Z	73	Kelas Z	82
Jadwal 1				Jadwal 5			

Gambar 11. Ilustrasi Proses Persilangan

Kelas M dan kelas X terpilih sebagai kelas yang akan mengalami persilangan. Sebelum proses persilangan, kelas M dan kelas X pada jadwal 1 berada pada posisi slot waktu 79 dan 250. Setelah mengalami persilangan, posisi kelas M dan X ditukar dengan posisi kelas M dan X pada jadwal 5, yaitu: 63 dan 45.

- 5) Mutasi. Mutasi dilakukan dengan memindahkan posisi kelas mata kuliah dari posisi awal ke posisi acak. Individu / jadwal yang mengalami mutasi dipilih secara acak, sedang jumlah kelas yang dimutasi adalah 2. Gambar 12 mengilustrasikan proses mutasi dalam suatu jadwal.



Gambar 12. Ilustrasi Proses Mutasi

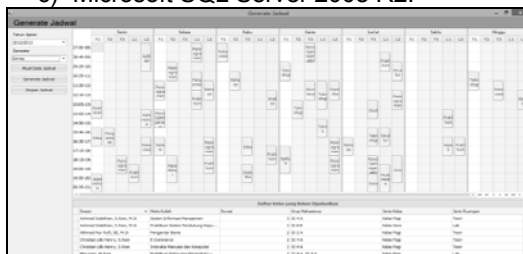
Kelas M dan kelas X terpilih sebagai kelas yang dimutasi. Sebelum dimutasi, kelas M berada pada posisi slot waktu 79, sedang kelas X berada pada posisi 250. Setelah dilakukan mutasi, posisi kelas M dan X menjadi 99 dan 273.

3. Penerapan sistem. Perangkat keras yang dibutuhkan untuk penerapan sistem yaitu:

- a) Perangkat komputer klien dengan spesifikasi minimum yaitu: prosesor Intel Pentium 4, kapasitas RAM sebesar 1 GB.
- b) Komputer server basis data, dengan spesifikasi minimum yaitu: prosesor Intel Dual Core dan kapasitas RAM sebesar 1 GB.

- c) Printer.
 - d) Jaringan intranet.
- Perangkat lunak yang dibutuhkan yaitu:

- a) Sistem operasi Windows XP atau Windows 7.
- b) Microsoft .Net Framework 4.
- c) Microsoft SQL Server 2008 R2.



Gambar 13. Form Generate dan Edit Jadwal

Pengujian dilakukan dengan metode *Black Box* yang dilakukan dengan cara memberikan masukan kepada sistem yang sudah berjalan dan mengamati hasil keluaran dari sistem apakah sudah sesuai dengan yang diharapkan.

4.1 Hasil Implementasi Rancangan

Sistem yang dihasilkan membutuhkan beberapa data master yang akan digunakan dalam proses penjawalan. Data yang dibutuhkan yaitu tahun ajaran, semester, ruangan, mata kuliah, grup mahasiswa, dosen, jenis kelas, pemetaan kelas mata kuliah dan ketidaktersediaan dosen.

Fitur utama dari sistem dimunculkan pada Form Generate dan Edit Jadwal digunakan untuk melakukan pembangkitan jadwal otomatis menggunakan algoritma genetik serta melakukan pengeditan jadwal dengan metode *drag and drop*, ditunjukkan pada gambar 13.

Form laporan jadwal matrik digunakan untuk menampilkan laporan jadwal dalam format matrik. Laporan jadwal matrik memuat semua informasi jadwal untuk semua hari, periode waktu, ruangan, dan kelas mata kuliah yang diajarkan, ditunjukkan pada gambar 14.



Gambar 14. Form Laporan Jadwal Matrik

Form laporan jadwal mengajar dosen digunakan untuk menampilkan laporan jadwal untuk tiap dosen, ditunjukkan pada gambar 15.



Gambar 15. Form Laporan Jadwal Mengajar Dosen

Form laporan cari jadwal kosong dosen dapat digunakan untuk mencari jadwal kosong dosen, sehingga memudahkan untuk mencari waktu rapat antar dosen atau waktu bimbingan bagi mahasiswa, ditunjukkan pada gambar 16.



Gambar 16. Form Laporan Cari Jadwal Kosong Dosen

4.2 Pembahasan

1. Pembangkitan Jadwal

Proses pembangkitan jadwal dilakukan menggunakan algoritma genetik, dengan tahapan yaitu: pendefinisian individu, penghitungan nilai *fitness*, seleksi individu terbaik, persilangan, dan mutasi. Individu dalam penelitian ini adalah solusi jadwal mata kuliah. Langkah yang dilakukan pada tahap pendefinisian jadwal adalah membangkitkan jadwal menggunakan data-data yang telah dimasukkan ke dalam aplikasi. Ada 2 prioritas yang digunakan dalam penentuan jadwal untuk tiap kelas mata kuliah, yaitu: kelas praktikum yang harus dijadwalkan di ruangan lab, dan dosen yang mempunyai data jadwal ketidakhadiran. Kelas yang mempunyai salah satu kriteria tersebut akan dicari jadwal terlebih dahulu. Penggunaan prioritas ini dimaksudkan untuk mempercepat proses pencarian solusi jadwal mata kuliah menggunakan algoritma genetik.

2. Pengeditan Jadwal

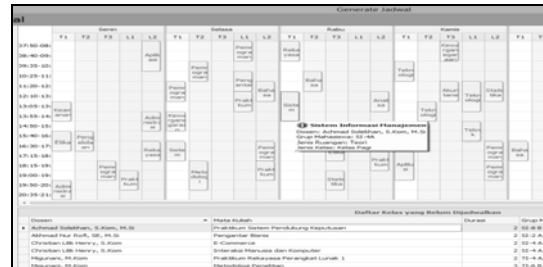
Pengeditan jadwal dilakukan apabila hasil dari pembangkitan jadwal masih belum sesuai dengan yang diharapkan atau masih ada kelas mata kuliah yang gagal dijadwalkan. Pengeditan jadwal dilakukan dengan metode *drag and drop*.

Kelas Mata Kuliah yang tidak berhasil dijadwalkan menggunakan metode pembangkitan jadwal akan ditampilkan dalam grid bagian kanan bawah pada form Generate dan Edit Jadwal. Gambar 17 menampilkan daftar kelas mata kuliah yang belum dijadwalkan.

Dosen	Mata Kuliah	Durasi	Grup Mahasiswa	Jenis Kelas	Jenis Ruangan
Achmad Sembahan, S.Kom, M.Si	Sistem Informasi Manajemen	3 SI-4 A		Kelas Pag	Teori
Achmad Sembahan, S.Kom, M.Si	Praktikum Sistem Pendukung Keputusan	2 SI-6 B		Kelas Sore	Lab
Ahmad Nur Rafi, SE, M.Si	Pengantar Bisnis	2 SI-2 A		Kelas Pag	Teori
Christian Lili Henry, S.Kom	E-Commerce	2 SI-4 A		Kelas Pag	Teori
Christian Lili Henry, S.Kom	Interaksi Manusia dan Komputer	2 SI-4 A		Kelas Pag	Teori

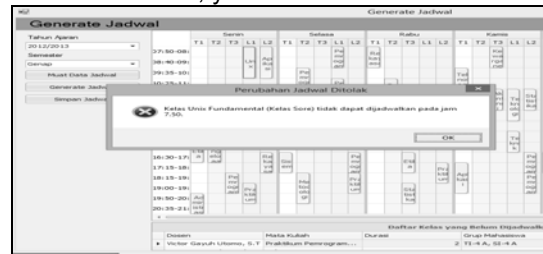
Gambar 17. Daftar Kelas Mata Kuliah yang Gagal Dijadwalkan Menggunakan Metode Pembangkitan Jadwal

Data Kelas tersebut dapat dipindah ke dalam grid jadwal (kanan atas) pada posisi yang sesuai. Gambar 18 menunjukkan kelas Sistem Informasi Manajemen yang telah di-*drag and drop* untuk dijadwalkan pada hari Rabu, jam 12.10-14.45, ruangan Teori 1.



Gambar 18. Penempatan kelas ke dalam grid jadwal

Pesan *error* akan diberikan ketika penempatan kelas mata kuliah pada grid jadwal tidak memenuhi persyaratan. Gambar 19 menunjukkan contoh pesan *error* yang disebabkan karena penempatan kelas mata kuliah untuk kelas sore di luar range periode waktu kelas sore, yaitu: 16.30-21.20.



Gambar 19. Pesan *Error* yang Muncul ketika Kelas Sore Ditempatkan di Luar *Range* Periode Waktu untuk Kelas Sore (16.30-21.20)

4.3 Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode *black box*, yaitu dengan cara memberikan sejumlah masukan ke aplikasi untuk mengetahui hasil keluaran yang dihasilkan, apakah sudah sesuai dengan fungsi aplikasi. Beberapa hasil pengujian dapat dilihat pada tabel 1.

Tabel 1. Hasil Pengujian Metode *Black Box*

No Kasus	Deskripsi	Prosedur Pengujian	Output yang Diharapkan	Hasil
UJ-01	Menjalankan aplikasi dan melakukan login	<ul style="list-style-type: none"> Klik 2x pada icon aplikasi Memasukkan nama pengguna dan kata sandi pada form login 	<ul style="list-style-type: none"> Muncul form login Berhasil login dan tampil daftar menu sesuai dengan hak akses pengguna 	<ul style="list-style-type: none"> Sesuai harapan Sesuai harapan
UJ-02	Mengelola Master Data Dosen	<ul style="list-style-type: none"> Menambah data dosen Mengedit data dosen 	<ul style="list-style-type: none"> Penambahan data dapat disimpan Pengeditan data dapat disimpan Penghapusan 	<ul style="list-style-type: none"> Sesuai harapan Sesuai harapan

		- Menghapus data dosen	data dapat disimpan	Sesuai harapan
UJ - 03	Mengelola Data Mapping Kelas	- Menambah data mapping kelas - Mengedit data mapping kelas - Menghapus data mapping kelas	Penambahan data dapat disimpan Pengeditan data dapat disimpan Penghapusan data dapat disimpan	Sesuai harapan Sesuai harapan Sesuai harapan
UJ - 04	Generate dan Edit Jadwal	- Membangkitkan jadwal - Mengedit jadwal dengan metode tarik-lepas	Berhasil membangkitkan jadwal menggunakan data master yang telah diinputkan Berhasil melakukan pengeditan dengan metode tarik-lepas	Sesuai yang diharapkan Sesuai yang diharapkan
UJ - 05	Mengelola data pengguna aplikasi	- Menambah pengguna - Mengedit data pengguna - Menghapus data pengguna	Penambahan data dapat disimpan Pengeditan data dapat disimpan Penghapusan data dapat disimpan	Sesuai harapan Sesuai harapan Sesuai harapan
UJ - 13	Mengelola hak akses pengguna aplikasi	- Mengedit hak akses pengguna	Berhasil menyimpan hak akses pengguna	Sesuai harapan

Dari hasil pengujian sistem, keluaran dari aplikasi penjadwalan yang dirancang telah sesuai dengan yang diharapkan, sehingga dapat disimpulkan bahwa perancangan aplikasi penjadwalan mata kuliah ini berhasil.

5.1 Kesimpulan

1. Aplikasi penjadwalan mata kuliah dapat mempermudah penyusunan jadwal kuliah untuk program studi Teknik Informatika dan Sistem Informasi.
2. Aplikasi penjadwalan mata kuliah dapat mempercepat dalam pembuatan berbagai laporan jadwal, yaitu: laporan jadwal mengajar bagi dosen, laporan jadwal kuliah bagi mahasiswa, dan laporan jadwal kosong dosen.

5.2 Saran

1. Penambahan fitur periode waktu yang fleksibel.
2. Integrasi dengan sistem akademik STMIK ProVisi, sehingga data-data master dapat diambil langsung dari database sistem akademik.
3. Perbaikan antarmuka grid jadwal pada form Generate dan Edit Jadwal, sehingga dapat memberikan informasi yang lebih informatif.

Daftar Pustaka :

- [1] Abramson, D. & Abela, J. 1992. *A Parallel Genetic Algorithm For Solving The School Timetabling Problem*. Australia : CSIRO & RMIT.
- [2] Ariani, D, Fahriza A, Prasetyaningrum I. 2011. *Optimasi Penjadwalan Mata Kuliah di Jurusan Teknik Informatika PENS dengan Menggunakan Algoritma Particle Swarm Optimization (PSO)*. PENS ITS, Surabaya.
- [3] Bartak, R. 1999. *Dynamic Constraint Models for Planning and Scheduling Problems* [Online]. Charles University, Prague. Tersedia: <http://ktiml.mff.cuni.cz/~bartak/downloads/ERCIM99-proceedings.pdf> [9 Januari 2012]
- [4] Jenal, R, Ismail, WR, Yuen LC, Oughalime, A. 2011. *A Cyclical Nurse Schedule Using Goal Programming*. Universiti Kebangsaan Malaysia, Selangor, Malaysia.
- [5] Nugraha, I. 2008. *Aplikasi Algoritma Genetik untuk Optimasi Penjadwalan Kegiatan Belajar Mengajar*. Institut Teknologi Bandung, Bandung.
- [6] Nugroho, A. 2010. *Rekayasa Perangkat Lunak, Berorientasi Objek dengan Metode USDP (Unified Software Development Process)*. Jakarta : Andi.
- [7] Petrovic, S & Burke, E. 2004. *University Time Tabling*. Nottingham : University of Nottingham.
- [8] Pinedo, M. 1995. *Scheduling - Theory, Algorithms, and Systems*. Prentice Hall, Englewood Cliffs.
- [9] Supriyanto, A. 2007. *Pengantar Teknologi Informasi*. Jakarta : Salemba Infotex.