

Rancang Bangun Aplikasi Penjadwalan Praktikum Di Laboratorium Komputer Universitas Dian Nuswantoro Dengan Pendekatan Algoritma Genetika

Ajib Susanto, Hasan Yahya
Program Studi Teknik Informatika, Fakultas Ilmu Komputer
Universitas Dian Nuswantoro Semarang
Email : ajibsusanto@gmail.com

Abstract

In Dian Nuswantoro University the schedule of computer practicum is usually done manually by a coordinator of practicum schedule which need good human resource, much time and enough energy. Since the colliding and overlapping the schedule often occurred, it needs the arrangement the schedule better. The approach can be used to overcome the problem is algorithm genetics approach. It is a computational approach which is able to solve the problem through model of biology process from evolution as the result the arrangement of practicum will be appropriated with the laboratory specification moreover the colliding or overlapping of the schedule will not occur.

Keywords : schedule arrangement, Genetics Approach.

PENDAHULUAN

Kegiatan belajar mengajar dalam suatu kampus merupakan aktifitas akademis yang paling penting. Dimana dosen dan mahasiswa saling berbagi pengetahuan mengenai suatu ilmu dalam mata kuliah di waktu yang tepat dan ruangan yang tepat pula. Idealnya seperti itu, akan tetapi kenyataan di lapangan tidaklah mudah. Penjadwalan kegiatan belajar mengajar dalam suatu instansi pendidikan adalah hal yang kompleks. Ada banyak aspek yang harus dilibatkan misalnya jadwal dosen untuk mengajar, tidak boleh adanya jadwal mata kuliah yang bentrok pada saat yang bersamaan, tidak boleh adanya bentrok ruang kelas yang dipakai, dll. [1]

Penjadwalan kegiatan belajar mengajar dalam suatu instansi pendidikan adalah hal yang kompleks. Ada banyak aspek yang harus dilibatkan misalnya jadwal dosen untuk mengajar, tidak boleh adanya jadwal mata kuliah yang bentrok pada saat yang bersamaan, tidak boleh adanya bentrok ruang kelas yang dipakai, dan lain lain. [1]

Udinus (Universitas Dian Nuswantoro) adalah salah satu universitas swasta terkemuka di Semarang. Keseriusan pengelola mengaplikasikan slogan *cyber university* membuat Udinus dikenal memiliki program studi Teknologi Informasi yang baik. [2] Hal ini didukung fasilitas laboratorium komputer yang memadai untuk praktikum.

Penjadwalan praktikum yang diimplementasikan oleh laboratorium komputer Udinus masih dilakukan secara manual. Dimana tiap-tiap fakultas memberikan daftar mata kuliah praktikum, nama dosen, dan waktu perkuliahan (reguler / malam) ke koordinator penjadwalan di laboratorium komputer. Koordinator akan menyatukan jadwal dari tiap-tiap fakultas, dan disesuaikan dengan spesifikasi lab masing-masing. Proses-proses tersebut sangatlah tidak maksimal karena memakan banyak

resource, baik waktu maupun sumber daya manusia. Disamping itu, resiko terjadinya tabrakan antar jadwal mata kuliah juga masih tergolong besar.

Salah satu faktor yang menyebabkan kurang efektifnya penjadwalan tersebut adalah karena tiap-tiap fakultas tidak memberikan jadwal pada waktu yang sama. Seringkali beberapa fakultas terlambat dalam memberikan daftar mata praktikum kuliah. Biasanya daftar mata kuliah praktikum diserahkan ke koordinator 3-2 minggu sebelum perkuliahan dimulai, sehingga pihak laboratorium dapat menyesuaikan *software* yang sesuai dengan mata kuliah praktikum yang ada.

Apabila terjadi keterlambatan dalam penyerahan daftar mata kuliah praktikum, pihak koordinator akan merombak kembali jadwal kuliah yang sudah dibuat, pada proses ini biasanya kesalahan terjadi, faktor *human error* berupa ketidakteknelitian dapat menyebabkan terjadinya jadwal tabrakan atau penempatan mata kuliah yang tidak sesuai dengan spesifikasi lab. Kesalahan ini akan mengganggu proses perkuliahan, terutama di minggu-minggu awal perkuliahan berlangsung, beberapa jadwal harus diganti di tengah-tengah jalannya perkuliahan.

Suatu pendekatan yang cukup efektif untuk solusi penjadwalan praktikum ini adalah Algoritma Genetika. Algoritma genetika (GA) adalah pencarian heuristik yang meniru proses evolusi alam. Heuristik ini secara rutin digunakan untuk menghasilkan solusi yang berguna untuk masalah optimasi dan pencarian. Algoritma genetik termasuk kelas yang lebih besar dari algoritma evolusioner (EA), yang menghasilkan solusi untuk masalah optimasi menggunakan teknik yang terinspirasi oleh evolusi alam, seperti warisan, mutasi, seleksi, dan *crossover*. [3]

Bagaimana mengkodekan solusi dari masalah menjadi kromosom merupakan masalah utama ketika menggunakan algoritma genetik. Masalah ini telah diteliti dari banyak aspek, seperti pemetaan karakter dari ruang genotipe ke ruang fenotipe ketika individu diterjemahkan ke dalam solusi, dan sifat metamorfosis ketika individu dimanipulasi oleh operator genetik. [4]

TINJAUAN PUSTAKA

Penjadwalan

Penjadwalan (*scheduling*) merupakan salah satu kegiatan penting dalam perusahaan. Penjadwalan adalah pengaturan waktu dari satu kegiatan operasi, yang mencakup kegiatan mengalokasikan fasilitas, peralatan maupun tenaga kerja, dan menentukan urutan pelaksanaan bagi suatu kegiatan operasi. Dalam suatu perusahaan industri, penjadwalan diperlukan antara lain dalam mengalokasikan tenaga operator, mesin dan peralatan produksi, urutan proses, jenis produk, dan pembelian material. Dalam suatu lembaga pendidikan, penjadwalan diperlukan antara lain untuk mengalokasikan ruang kelas, peralatan mengajar, tenaga pengajar, staf administrasi, dan pendaftaran penerimaan mahasiswa baru atau ujian. Demikian pula, dalam kegiatan perhotelan, penjadwalan diperlukan dalam pengaturan kamar hotel, ruang seminar/resepsi, menu makanan, ataupun secara *entertainment*. [5]

Penjadwalan bertujuan meminimalkan waktu proses, waktu tunggu langganan, dan tingkat persediaan, serta penggunaan yang efisien dari fasilitas, tenaga kerja, dan peralatan. Penjadwalan biasanya disusun dengan mempertimbangkan berbagai keterbatasan yang ada. Terlepas dari jenis perusahaannya, setiap perusahaan perlu untuk melakukan penjadwalan sebaik mungkin agar dapat memperoleh utilisasi yang maksimum dari sumber daya produksi dan aset lain yang dimilikinya. Penjadwalan yang baik akan memberikan dampak positif yaitu rendahnya biaya operasi dan waktu pengiriman, yang pada akhirnya dapat meningkatkan kepuasan pelanggan. [5]

Algoritma Genetika

Algoritma ini ditemukan di Universitas Michigan, Amerika Serikat oleh John Holland (1975) melalui sebuah penelitian dan dipopulerkan oleh salah satu muridnya, David Goldberg (1989). Dimana mendefinisikan algoritma genetik ini sebagai metode algoritma pencarian berdasarkan pada mekanisme seleksi alam dan genetik alam. [6]

Algoritma genetik adalah algoritma yang berusaha menerapkan pemahaman mengenai evolusi alamiah pada tugas-tugas pemecahan-masalah (*problem solving*). 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 mensimulasikan proses evolusioner. [6]

Pada akhirnya, akan didapatkan solusi-solusi yang paling tepat bagi permasalahan yang dihadapi. Untuk menggunakan algoritma genetik, solusi permasalahan direpresentasikan sebagai kromosom. Tiga aspek yang penting untuk penggunaan algoritma genetik:

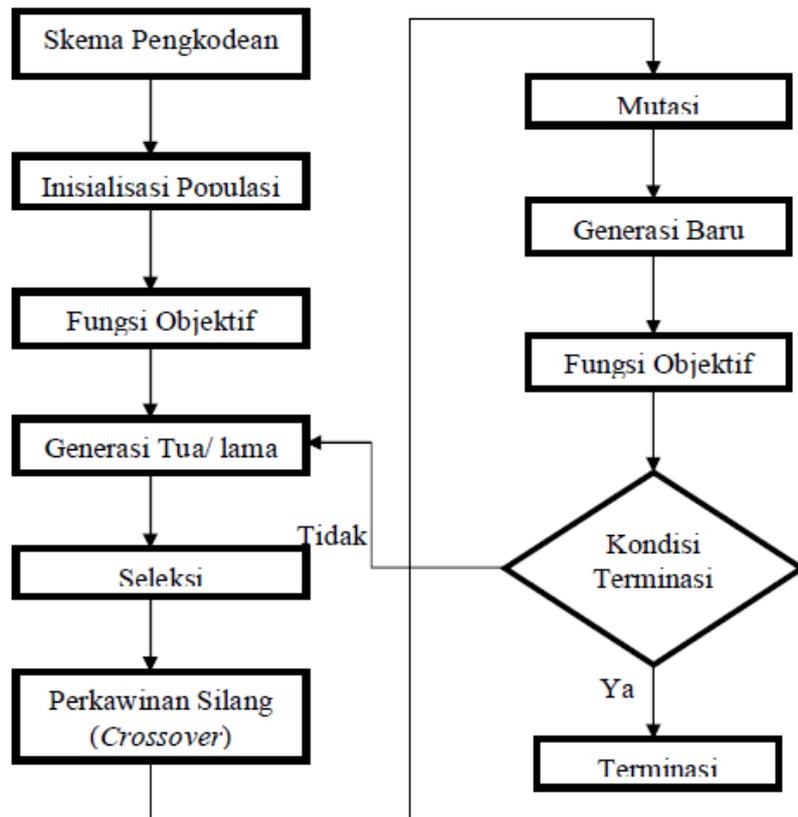
1. Definisi fungsi *fitness*
2. Definisi dan implementasi representasi genetik
3. Definisi dan implementasi operasi genetik

Jika ketiga aspek di atas telah didefinisikan, algoritma genetik akan bekerja dengan baik. Tentu saja, algoritma genetik bukanlah solusi terbaik untuk memecahkan segala masalah. Sebagai contoh, metode tradisional telah diatur untuk mencari penyelesaian dari fungsi analitis convex yang “berperilaku baik” yang variabelnya sedikit. Pada kasus-kasus ini, metode berbasis kalkulus lebih unggul dari algoritma genetik karena metode ini dengan cepat menemukan solusi minimum ketika algoritma genetik masih menganalisa bobot dari populasi awal.

Untuk problem-problem ini pengguna harus mengakui fakta dari pengalaman ini dan memakai metode tradisional yang lebih cepat tersebut. Akan tetapi, banyak persoalan realistis yang berada di luar golongan ini. Selain itu, untuk persoalan yang tidak terlalu rumit, banyak cara yang lebih cepat dari algoritma genetik. Jumlah besar dari populasi solusi, yang merupakan keunggulan dari algoritma genetik, juga harus mengakui kekurangannya dalam dalam kecepatan pada sekumpulan komputer yang dipasang secara seri-fitness function dari tiap solusi harus dievaluasi. Namun, bila tersedia komputer-komputer yang paralel, tiap prosesor dapat mengevaluasi fungsi yang terpisah pada saat yang bersamaan. Karena itulah, algoritma genetik sangat cocok untuk perhitungan yang paralel.

Struktur Umum Algoritma Genetik

Algoritma genetik memberikan suatu pilihan bagi penentuan nilai parameter dengan meniru cara reproduksi genetik, pembentukan kromosom baru serta seleksi alami seperti yang terjadi pada makhluk hidup. Algoritma Genetik secara umum dapat diilustrasikan dalam diagram alir berikut ini:



Gambar 2. Diagram Alur Algoritma Genetik

Golberg (1989) mengemukakan bahwa algoritma genetik mempunyai karakteristik-karakteristik yang perlu diketahui sehingga dapat terbedakan dari prosedur pencarian atau optimasi yang lain, yaitu:

1. Algoritma genetik dengan pengkodean dari himpunan solusi permasalahan berdasarkan parameter yang telah ditetapkan dan bukan parameter itu sendiri.
2. Algoritma genetik pencarian pada sebuah solusi dari sejumlah individu-individu yang merupakan solusi permasalahan bukan hanya dari sebuah individu.
3. Algoritma genetik informasi fungsi objektif (*fitness*), sebagai cara untuk mengevaluasi individu yang mempunyai solusi terbaik, bukan turunan dari suatu fungsi.
4. Algoritma genetik menggunakan aturan-aturan transisi peluang, bukan aturan-aturan deterministik.

Variabel dan parameter yang digunakan pada algoritma genetik adalah:

1. Fungsi *fitness* (fungsi tujuan) yang dimiliki oleh masing-masing individu untuk menentukan tingkat kesesuaian individu tersebut dengan kriteria yang ingin dicapai.
2. Populasi jumlah individu yang dilibatkan pada setiap generasi.
3. Populasi jumlah individu yang dilibatkan pada setiap generasi.
4. Probabilitas terjadinya mutasi pada setiap individu.
5. Jumlah generasi yang akan dibentuk yang menentukan lama penerapan algoritma genetik.

Secara umum struktur dari suatu algoritma genetik dapat mendefinisikan dengan langkah-langkah sebagai berikut:

1. Membangkitkan populasi awal

Populasi awal ini dibangkitkan secara random sehingga didapatkan solusi awal. Populasi itu sendiri terdiri atas sejumlah kromosom yang merepresentasikan solusi yang diinginkan.

2. Membentuk generasi baru

Untuk membentuk generasi baru, digunakan operator reproduksi/ seleksi, crossover dan mutasi. Proses ini dilakukan berulang-ulang sehingga didapatkan jumlah kromosom yang cukup untuk membentuk generasi baru dimana generasi baru ini merupakan representasi dari solusi baru. Generasi baru ini dikenal dengan istilah anak (offspring).

3. Evaluasi solusi

Pada tiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang dinamakan fitness. Nilai fitness suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai fitness setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Bila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah 2. Beberapa kriteria berhenti sering digunakan antara lain: berhenti pada generasi tertentu, berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai fitness tertinggi tidak berubah, berhenti dalam n generasi tidak didapatkan nilai fitness yang lebih tinggi.

Fungsi *fitness* tersebut sebagai berikut:

$$Fitness = \frac{1}{1 + Penalti} \quad (1)$$

dimana :

$$penalti = \sum Bp \sum Np$$

Dari persamaan diatas nilai fitness ditentukan oleh nilai *penalty*. *Penalty* tersebut menunjukkan jumlah pelanggaran kendala pada suatu kromosom. Semakin tinggi nilai *fitness* akan semakin besar kemungkinan kromosom tersebut terpilih ke generasi berikutnya. Jadi nilai *penalty* berbanding terbalik dengan nilai *fitness*, semakin kecil nilai *penalty* (jumlah pelanggaran) semakin besar nilai fitnessnya.

Jadi fungsi *fitness* :

$$Fitness = \frac{1}{1 + \sum Bp \sum Np} \quad (2)$$

Keterangan :

Bp : Bobot Pelanggaran

Np : Indikator Pelanggaran

Pengkodean

Pengkodean adalah suatu teknik untuk menyatakan populasi awal sebagai calon solusi suatu masalah ke dalam suatu kromosom sebagai suatu kunci pokok persoalan ketika menggunakan algoritma genetik.

Berdasarkan jenis symbol yang digunakan sebagai nilai suatu gen, metode pengkodean dapat diklasifikasikan sebagai berikut:

1. Pengkodean biner merupakan cara pengkodean yang paling umum digunakan karena adalah yang pertama kali digunakan dalam algoritma genetik oleh Holland. Keuntungan pengkodean ini adalah sederhana untuk diciptakan dan mudah dimanipulasi. Pengkodean biner memberikan banyak kemungkinan untuk kromosom walaupun dengan jumlah nilai-nilai yang mungkin terjadi pada suatu

gen yang sedikit (0 dan 1). Di pihak lain, pengkodean biner sering tidak sesuai untuk banyak masalah dan kadang pengoreksian harus dilakukan setelah operasi *crossover* dan mutasi.

2. Pengkodean bilangan riil adalah suatu pengkodean bilangan dalam bentuk riil. Masalah optimalisasi fungsi dan optimalisasi kendala lebih tepat jika diselesaikan dengan pengkodean bilangan riil karena struktur topologi ruang genotif untuk pengkodean bilangan riil identik dengan ruang fenotifnya, sehingga mudah membentuk operator genetik yang efektif dengan cara memakai teknik yang dapat digunakan yang berasal dari metode konvensional.
3. Pengkodean bilangan bulat adalah metode yang mengkodekan bilangan dalam bentuk bilangan bulat. Pengkodean ini baik digunakan untuk masalah optimisasi kombinatorial.
4. Pengkodean struktur data adalah model pengkodean yang menggunakan struktur data. Pengkodean ini digunakan untuk masalah kehidupan yang lebih kompleks seperti perencanaan jalur robot, dan masalah pewarnaan *Graph*.

Operator Genetik

Algoritma genetik merupakan proses pencarian yang heuristik dan acak sehingga penekanan pemilihan operator yang digunakan sangat menentukan keberhasilan algoritma genetik dalam menemukan solusi optimum suatu masalah yang diberikan. Hal yang harus diperhatikan adalah menghindari terjadinya konvergensi *premature*, yaitu mencapai solusi optimum yang belum waktunya, dalam arti bahwa solusi yang diperoleh adalah hasil optimum lokal.

Operator genetik yang digunakan setelah proses evaluasi tahap pertama membentuk populasi baru dari generasi sekarang. Operator-operator tersebut adalah operator seleksi, *crossover* dan mutasi.

a. Seleksi

Seleksi bertujuan memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling *fit*. Langkah pertama dalam seleksi ini adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap seleksi berikutnya. [7]

Kemampuan algoritma genetik untuk memproduksi kromosom yang lebih baik secara progresif tergantung pada penekanan selektif (*selective pressure*) yang diterapkan ke populasi. Penekanan selektif dapat diterapkan dalam dua cara. Cara pertama adalah membuat lebih banyak kromosom anak yang dipelihara dalam populasi dan memilih hanya kromosom-kromosom terbaik bagi generasi berikut. Walaupun orang tua dipilih secara acak, metode ini akan terus menghasilkan kromosom yang lebih baik berhubungan dengan penekanan selektif yang diterapkan pada individu anak tersebut.

Cara lain menerapkan penekanan selektif adalah memilih orang tua yang lebih baik ketika membuat keturunan baru. Dengan metode ini, hanya kromosom sebanyak yang dipelihara dalam populasi yang perlu dibuat bagi generasi berikutnya. Walaupun penekanan selektif tidak diterapkan ke level keturunan, metode ini akan terus menghasilkan kromosom yang lebih baik, karena adanya penekanan selektif yang diterapkan ke orangtua.

Ada beberapa metode untuk memilih kromosom yang sering digunakan antara lain adalah seleksi roda rolet (*roulette wheel selection*), seleksi ranking (*rank selection*) dan seleksi turnamen (*tournament selection*).

Dalam penelitian ini, metode yang digunakan adalah seleksi roda rolet (*roulette wheel selection*). Pada seleksi ini, orang tua dipilih berdasarkan *fitness* mereka. Lebih baik kualitas suatu kromosom, lebih besar peluangnya untuk terpilih. Probabilitas suatu individu terpilih untuk *crossover* sebanding dengan *fitness*-nya. Cara penyeleksian ini merupakan peniruan dari permainan roda rolet.

b. Crossover

Crossover (perkawinan silang) bertujuan menambah keanekaragaman string dalam populasi dengan penyilangan antar-string yang diperoleh dari sebelumnya. Beberapa jenis crossover tersebut adalah:

1. *Crossover* 1-titik

Pada crossover dilakukan dengan memisahkan suatu string menjadi dua bagian dan selanjutnya salah satu bagian dipertukarkan dengan salah satu bagian dari string yang lain yang telah dipisahkan dengan cara yang sama. Proses yang demikian dinamakan operator crossover satu titik seperti diperlihatkan pada gambar berikut:

Tabel 1 :Contoh *Crossover* 1-titik

Kromosom Orangtua 1	11001011
Kromosom Orangtua 2	11011111
Keturunan	11001111

2. *Crossover* 2-titik

Proses crossover ini dilakukan dengan memilih dua titik crossover. Kromosom keturunan kemudian dibentuk dengan barisan bit dari awal kromosom sampai titik crossover pertama disalin dari orangtua pertama, bagian dari titik crossover pertama dan kedua disalin dari orangtua kedua, kemudian selebihnya disalin dari orangtua pertama lagi.

Tabel 2 : Contoh *Crossover* 2-titik

Kromosom Orangtua 1	11001011
Kromosom Orangtua 2	11011111
Keturunan	11011111

3. *Crossover* seragam

Crossover seragam menghasilkan kromosom keturunan dengan menyalin bit-bit secara acak dari kedua orangtuanya.

Tabel 3 : Contoh *Crossover* seragam

Kromosom Orangtua 1	11001011
Kromosom Orangtua 2	11011111
Keturunan	11011111

c. Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Operasi *crossover* yang dilakukan pada kromosom dengan tujuan untuk memperoleh kromosom-kromosom baru sebagai kandidat solusi pada generasi mendatang dengan *fitness* yang lebih baik, dan lama-kelamaan menuju solusi optimum yang diinginkan. Akan tetapi, untuk mencapai hal ini, penekanan selektif juga memegang peranan yang penting. Jika dalam proses pemilihan kromosom-kromosom cenderung pada kromosom yang memiliki

fitness yang tinggi saja, konvergensi *premature*, yaitu mencapai solusi yang optimal lokal sangat mudah terjadi.

Untuk menghindari konvergensi *premature* tersebut dan tetap menjaga perbedaan (*diversity*) kromosom-kromosom dalam populasi, selain melakukan penekanan selektif yang lebih efisien, operator mutasi juga dapat digunakan. Proses mutasi dalam sistem biologi berlangsung dengan mengubah isi *allele* gen pada suatu *locus* dengan *allele* yang lain. Proses mutasi ini bersifat acak sehingga tidak selalu menjamin bahwa setelah proses mutasi akan diperoleh kromosom dengan *fitness* yang lebih baik.

Operator mutasi merupakan operasi yang menyangkut satu kromosom tertentu. Beberapa cara operasi mutasi diterapkan dalam algoritma genetik menurut jenis pengkodean terhadap *phenotype*, antara lain:

1. Mutasi dalam Pengkodean Biner

Mutasi pada pengkodean biner merupakan operasi yang sangat sederhana. Proses yang dilakukan adalah menginversi nilai bit pada posisi tertentu yang terpilih secara acak (atau menggunakan skema tertentu) pada kromosom, yang disebut *inverse* bit.

Tabel 4 : Contoh Mutasi Pada Pengkodean Biner

Kromosom sebelum mutasi	1 0 0 1 0 1 1 1
Kromosom setelah mutasi	1 0 0 1 0 0 1 1

2. Mutasi dalam Pengkodean Permutasi

Proses mutasi yang dilakukan dalam pengkodean biner dengan mengubah langsung bit-bit pada kromosom tidak dapat dilakukan pada pengkodean permutasi karena konsistensi urutan permutasi harus diperhatikan. Salah satu cara yang dapat dilakukan adalah dengan memilih dua posisi (*locus*) dari kromosom dan kemudian nilainya saling dipertukarkan.

Tabel 5 : Contoh Mutasi Pada Pengkodean Permutasi

Kromosom sebelum mutasi	1 2 3 4 6 5 8 7 9
Kromosom setelah mutasi	1 2 7 4 6 5 8 3 9

3. Mutasi dalam Pengkodean Nilai

Mutasi pada pengkodean nilai hampir sama dengan yang dilakukan pada pengkodean biner, tetapi yang dilakukan bukan menginversi nilai bit. Penerapannya bergantung pada jenis nilai yang digunakan. Sebagai contoh untuk nilai riil, proses mutasi dapat dilakukan seperti yang dilakukan pada pengkodean permutasi, dengan saling mempertukarkan nilai dua gen pada kromosom.

4. Mutasi dalam Pengkodean Pohon

Mutasi dalam pengkodean pohon dapat dilakukan antara lain dengan cara mengubah operator (+, -, *, /) atau nilai yang terkandung pada suatu verteks pohon yang dipilih. Atau, dapat juga dilakukan dengan memilih dua verteks dari pohon dan saling mempertukarkan operator atau nilainya.

Tidak setiap gen selalu dimutasi tetapi mutasi dikontrol dengan probabilitas tertentu yang disebut dengan *mutation rate* (probabilitas mutasi) dengan notasi *P_m*. Jenis operator mutasi antara lain:

1. Mutasi Terarah

Mutasi terarah tergantung dari informasi gen. Informasi gen tersebut berupa nilai pelanggaran gen (*violation gen*). Ini berarti bahwa setiap gen mempunyai peluang yang berbeda untuk terjadi mutasi. Gen yang mempunyai nilai pelanggaran yang lebih besar maka gen tersebut

mempunyai peluang untuk terjadi mutasi. Mutasi ini menghubungkan nilai pelanggaran relatif (nilai pelanggaran suatu gen dibagi dengan nilai pelanggaran total suatu kromosom) dengan probabilitas terjadinya mutasi dari suatu gen pada kromosom. Hubungan tersebut dinyatakan secara matematis sebagai berikut:

$$nr(i) = \frac{n(i)}{1 + n_{total}} \quad (3)$$

$$pm(i) = (1 + nr(i))^2 pm \quad (4)$$

Keterangan persamaan:

$nr(i)$: nilai pelanggaran relative gen ke-i

n_{total} : nilai pelanggaran total kromosom

2. Mutasi Biasa

Mutasi biasa tidak tergantung dari informasi gen. Setiap gen mempunyai peluang yang sama untuk terjadi mutasi.

Parameter Genetik

Pengoperasian algoritma genetik dibutuhkan 4 parameter [8] yaitu:

1. Probabilitas Persilangan (*Crossover Probability*)

Menunjukkan kemungkinan crossover terjadi antara 2 kromosom. Jika tidak terjadi crossover maka keturunannya akan sama persis dengan kromosom orangtua, tetapi tidak berarti generasi yang baru akan sama persis dengan generasi yang lama. Jika probabilitas crossover 100% maka semua keturunannya dihasilkan dari crossover. Crossover dilakukan dengan harapan bahwa kromosom yang baru akan lebih baik. [8]

2. Probabilitas Mutasi (*Mutation Probability*)

Menunjukkan kemungkinan mutasi terjadi pada gen-gen yang menyusun sebuah kromosom. Jika tidak terjadi mutasi maka keturunan yang dihasilkan setelah *crossover* tidak berubah. Jika terjadi mutasi bagian kromosom akan berubah. Jika probabilitas 100%, semua kromosom dimutasi. Jika probabilitasnya 0%, tidak ada yang mengalami mutasi. [8]

3. Jumlah Individu

Menunjukkan jumlah kromosom yang terdapat dalam populasi (dalam satu generasi). Jika hanya sedikit kromosom dalam populasi maka algoritma genetik akan mempunyai sedikit variasi kemungkinan untuk melakukan *crossover* antara orangtua karena hanya sebagian kecil dari *search space* yang dipakai. Sebaliknya jika terlalu banyak maka algoritma genetik akan berjalan lambat. [8]

4. Jumlah Populasi

Menentukan jumlah populasi atau banyaknya generasi yang dihasilkan, digunakan sebagai batas akhir proses seleksi, persilangan dan mutasi. [8]

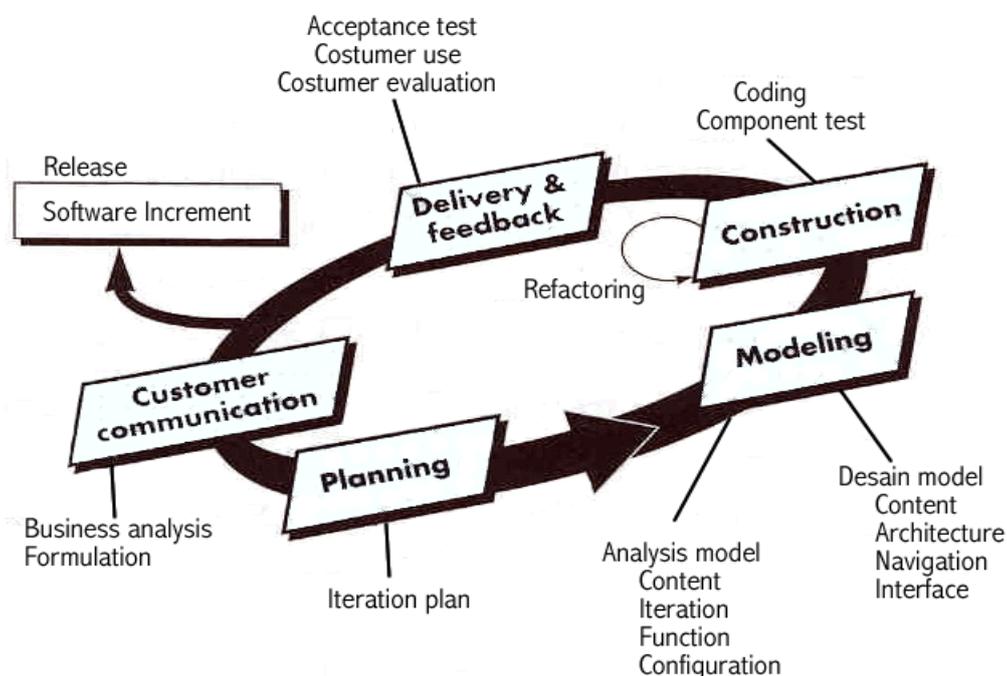
METODOLOGI PENELITIAN

Perancangan bangun sistem ini akan menggunakan pengembangan perangkat lunak dengan model *web engineering*. Model tersebut dipilih karena aplikasi penjadwalan ini berbasis *web*, sehingga pendekatan penelitian yang digunakan akan lebih tepat jika menggunakan *web engineering*. Karena *web engineering* digunakan untuk menciptakan aplikasi *web* yang berkualitas tinggi. Rekayasa *web* adalah

proses yang digunakan untuk menciptakan aplikasi *web* yang berkualitas tinggi. Rekayasa *web* mengadaptasi rekayasa perangkat lunak dalam hal konsep dasar yang menekankan pada aktifitas teknis dan manajemen. Namun demikian adaptasi tidak secara utuh, tapi dengan perubahan dan penyesuaian. Rekayasa *web* gabungan antara *web publishing* (suatu konsep yang berasal dari *printed publishing*) dan aktifitas rekayasa perangkat lunak. Dikatakan demikian karena desain sebuah aplikasi *web* menekankan pada desain grafis, desain informasi, teori *hypertext*, desain sistem dan pemrograman.

Tahapan Pengembangan Web engineering

Di dalam Pengembangan atau kegiatan *developing* aplikasi penjadwalan ini terkait dengan model *Web Engineering*[9] maka urutan proses atau tahapan-tahapan yang harus dilalui di dalam aplikasi *web* adalah :



Gambar 1 : Tahapan *Web Engineering* (Pressman,2005).

1. Customer Communication
Komunikasi dengan pengguna dimulai dengan wawancara dengan koordinator penjadwalan untuk mempelajari alur kerja pengolahan data serta para pelaku (pengguna) di dalam aplikasi penjadwalan baik itu mahasiswa, dosen maupun laboran untuk mendefinisikan permasalahan yang terjadi.
2. Web Engineering Planning/Perencanaan
Tahap perencanaan yang dilakukan pada penelitian ini adalah tahap penggabungan permintaan dan informasi dari pengguna dan perencanaan teknis serta respon dari pengguna. Perencanaan teknis dilakukan dengan mengidentifikasi perangkat lunak yang diperlukan.
3. Web Engineering Modelling
 - a. Analysis Modelling
Proses analisis merupakan lanjutan dari tahap komunikasi dengan pengguna.

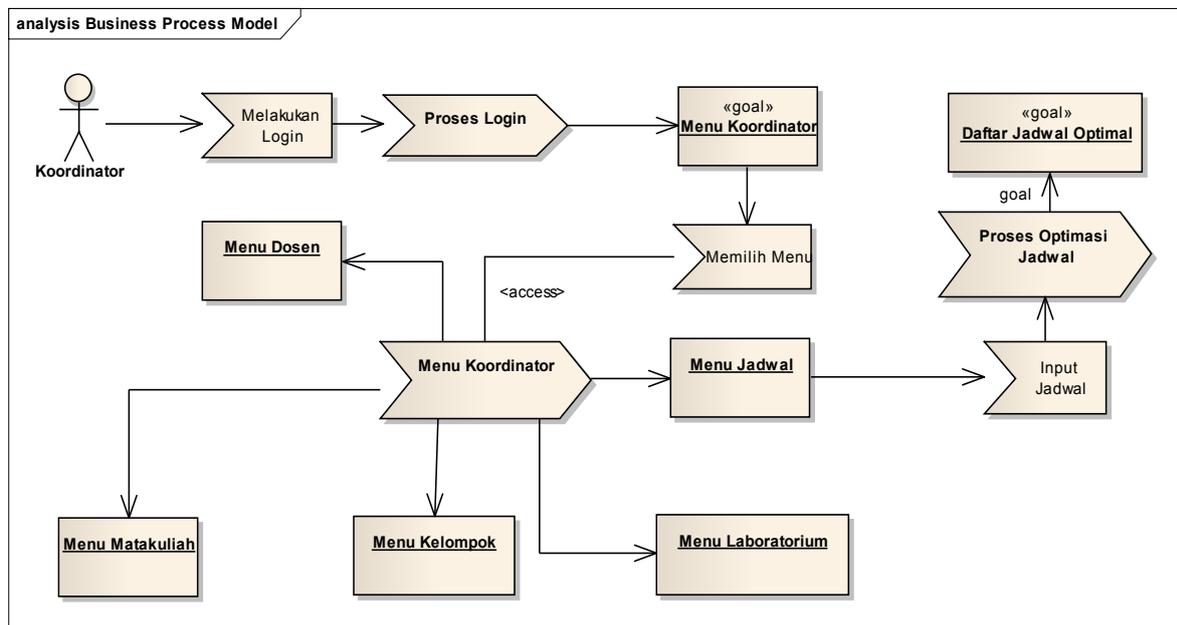
- 1) Analisis Isi (*Content*), dilakukan dengan merumuskan kebutuhan dari sistem pengolahan data penjadwalan beserta permasalahannya.
 - 2) Analisis Interaksi, dilakukan dengan mengidentifikasi interaksi pengguna dengan sistem berdasarkan hak akses pengguna.
 - 3) Analisis Fungsional, dilakukan dengan mengidentifikasi operasi-operasi yang akan dijalankan dan proses-proses yang terpisah dengan sistem tetapi sangat penting bagi pengguna.
 - 4) Analisis Konfigurasi, dilakukan dengan mengidentifikasi lingkungan dan infrastruktur yang tepat untuk aplikasi yang akan dibuat.
- b. Design Modelling
- Tahap desain yang dilakukan yaitu :
- 1) Desain Antarmuka (*interface*)
Desain antarmuka dilakukan dengan memeriksa kumpulan informasi yang telah dilakukan pada tahap analisis, membuat sketsa antarmuka dari aplikasi *web*, memetakan obyektif pengguna ke dalam antarmuka yang spesifik.
 - 2) Desain Estetika
Desain estetika dilakukan dengan merancang tampilan halaman dengan kombinasi warna, teks dan gambar yang sesuai dengan isi dan tujuan aplikasi *web*.
 - 3) Desain Isi (*content*)
Desain isi dilakukan dengan merancang *content* dari aplikasi *web*. Desain-desain tersebut dirancang berdasarkan kebutuhan informasi yang telah diidentifikasi pada tahap analisis. Sedangkan desain basisdata yang dilakukan yaitu : desain model konseptual, desain model *logic* dan desain model fisik.
 - 4) Desain Navigasi
Desain navigasi untuk aplikasi berbasis *web* memiliki aturan atau hak otorisasi untuk setiap pengguna sesuai dengan alur kerja sistem.
 - 5) Desain Arsitektur
Desain arsitektur untuk aplikasi berbasis *web* difokuskan pada struktur *hypermedia* secara keseluruhan dari aplikasi *web*. Struktur arsitektur berkaitan erat dengan tujuan dari pengembangan situs, *content* yang disediakan dan pengguna yang mengunjungi situs (*end-user*).
4. Web Engineering Construction
- a. Implementasi (coding)
Implementasi dilakukan dengan mengaplikasikan halaman *web* dalam bentuk PHP secara keseluruhan berdasarkan hasil proses perancangan isi pada aktivitas *non technical* maupun *technical member*, begitu juga dengan implementasi isi dibuat dalam bentuk PHP dengan kombinasi teknik *Ajax*.
 - b. Pengujian (Component Test)
Proses pengujian dilakukan untuk mengetahui kemungkinan terjadinya kesalahan seperti kesalahan pada skrip atau *form*, navigasi ataupun tampilan.
5. Web Engineering Delivery and Evaluation
- Serah terima dan respon dilakukan dengan menyebarkan kuisisioner kepada responden untuk mendapatkan penilaian dari setiap kriteria pada evaluasi. Kriteria evaluasi yang dinilai adalah *usability* (kegunaan), *functionality* (fungsional), serta *reliability*. *Efficiency* dan kegiatan *maintenance* pada penelitian ini tidak dilakukan karena sistem belum dilakukan instalasi.

WebApp Requirements Analysis

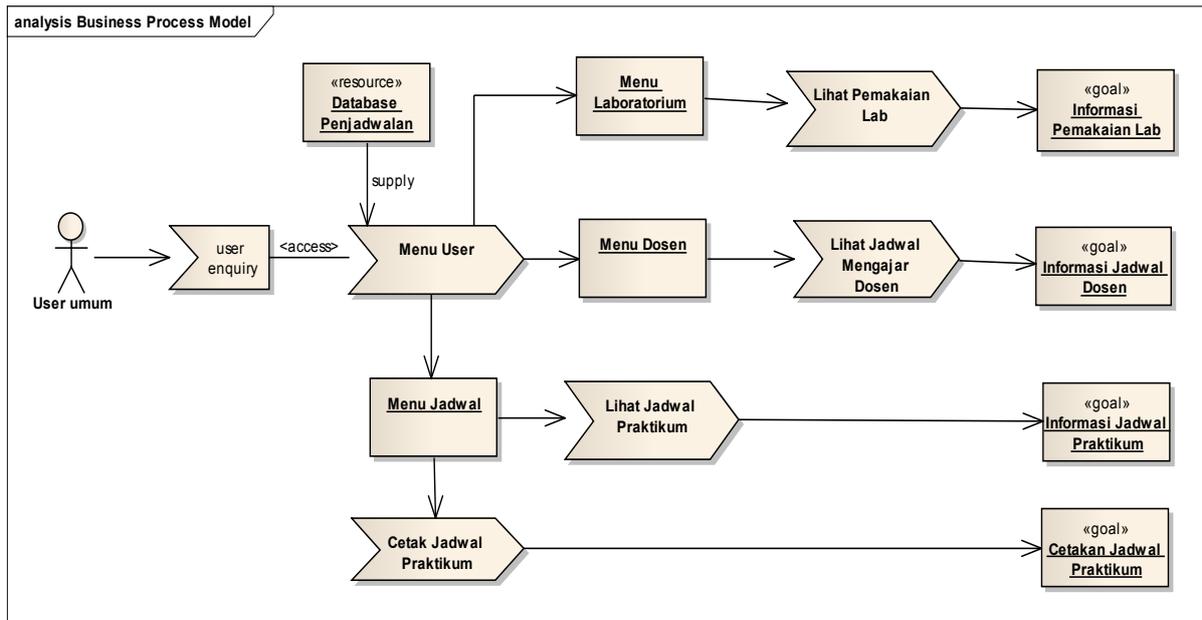
Tugas-tugas dari analisa kebutuhan di dalam aplikasi *web* adalah sebagai berikut :

1. Formulation
Mengidentifikasi tujuan objek-objek dari aplikasi *web* dan mengelompokkan *user* sesuai dengan kebutuhan masing-masing *user*.
2. Requirements Gathering
Di dalam tahap ini tim *web engineer* melakukan komunikasi dengan pihak laboratorium, yaitu kepala laboratorium dan *stakeholder* terkait pengembangan *web*, selain itu pada tahap ini harus membuat daftar dari konten dan kebutuhan-kebutuhan secara fungsional.
3. Analysis modeling
Tahap ini merupakan tahap analisa model yang digunakan di dalam *web engineering* yang meliputi 4 model, yaitu :
 1. Content model
 2. Iteration model
 3. Functional model
4. Configuration model

PEMBAHASAN Model Proses Bisnis



Gambar 3 : Model Proses Bisnis pada Koordinator

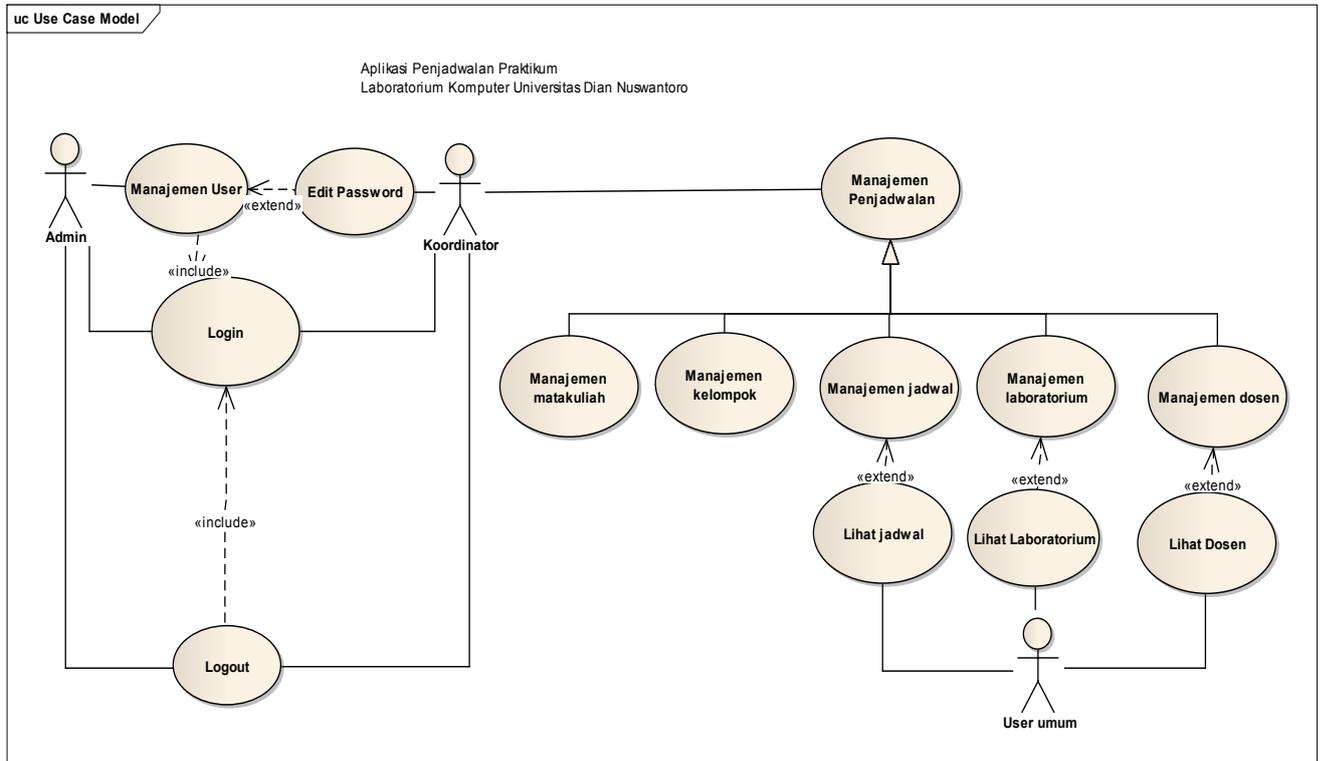


Gambar 4 : Model Proses Bisnis pada user umum

Model proses bisnis gambar 3 menerangkan proses bisnis koordinator penjadwalan, dimana koordinator harus terlebih dahulu melakukan login sebelum menggunakan aplikasi penjadwalan praktikum.

Gambar 4 menerangkan proses bisnis pada user umum, dimana user dapat melihat informasi pemakaian tiap-tiap laboratorium, jadwal dosen mengajar praktikum, dan jadwal praktikum keseluruhan yang ada di laboratorium komputer Universitas Dian Nuswantoro.

Diagram model Use-case

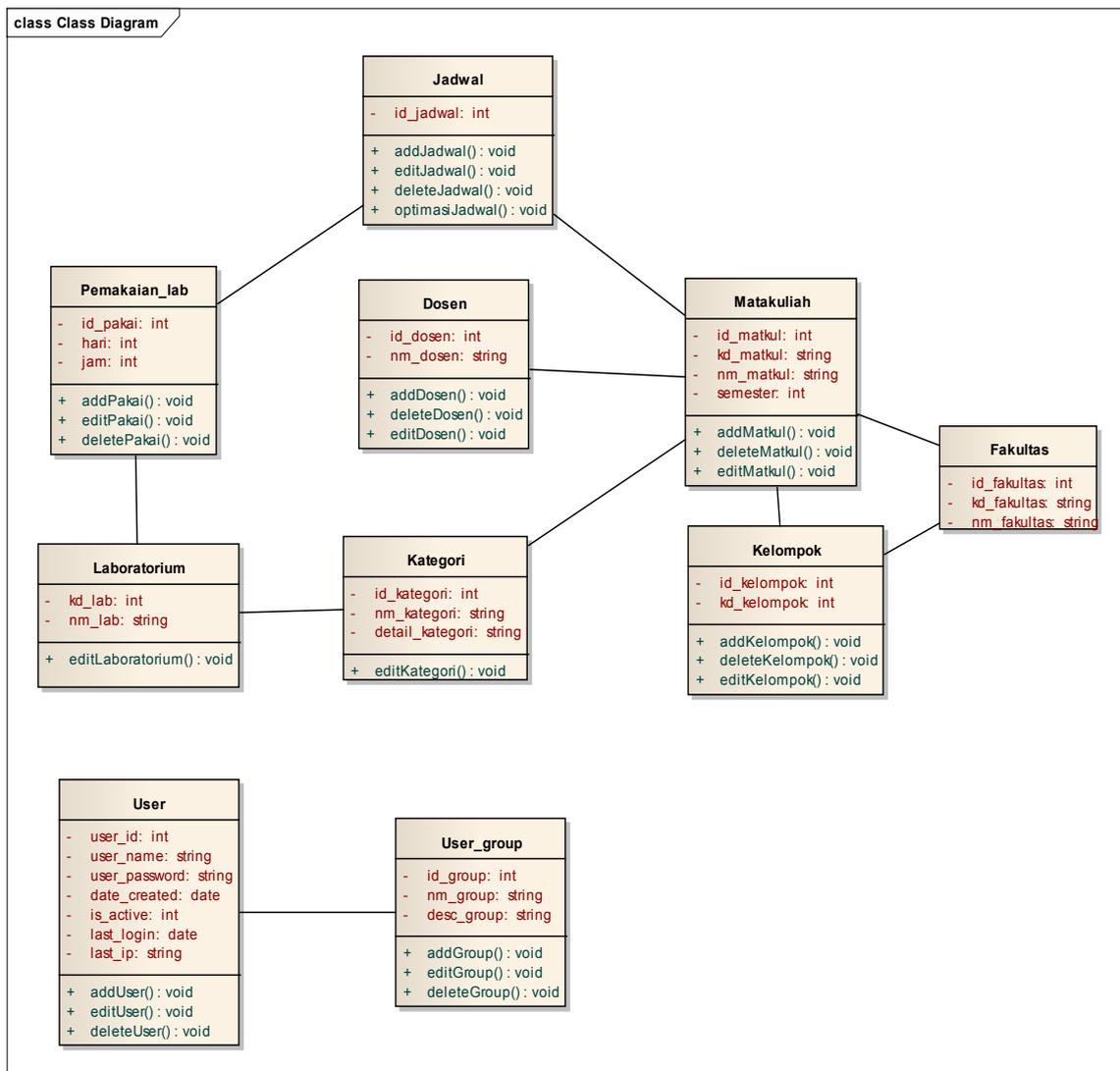


Gambar 5 : Diagram Use Case

Gambar di atas menggambarkan interaksi *user* dengan sistem aplikasi penjadwalan praktikum laboratorium komputer Universitas Dian Nuswantoro, dimana koordinator harus melakukan *login* terlebih dahulu sebelum dapat mengakses menu-menu yang ada di dalam aplikasi penjadwalan praktikum.

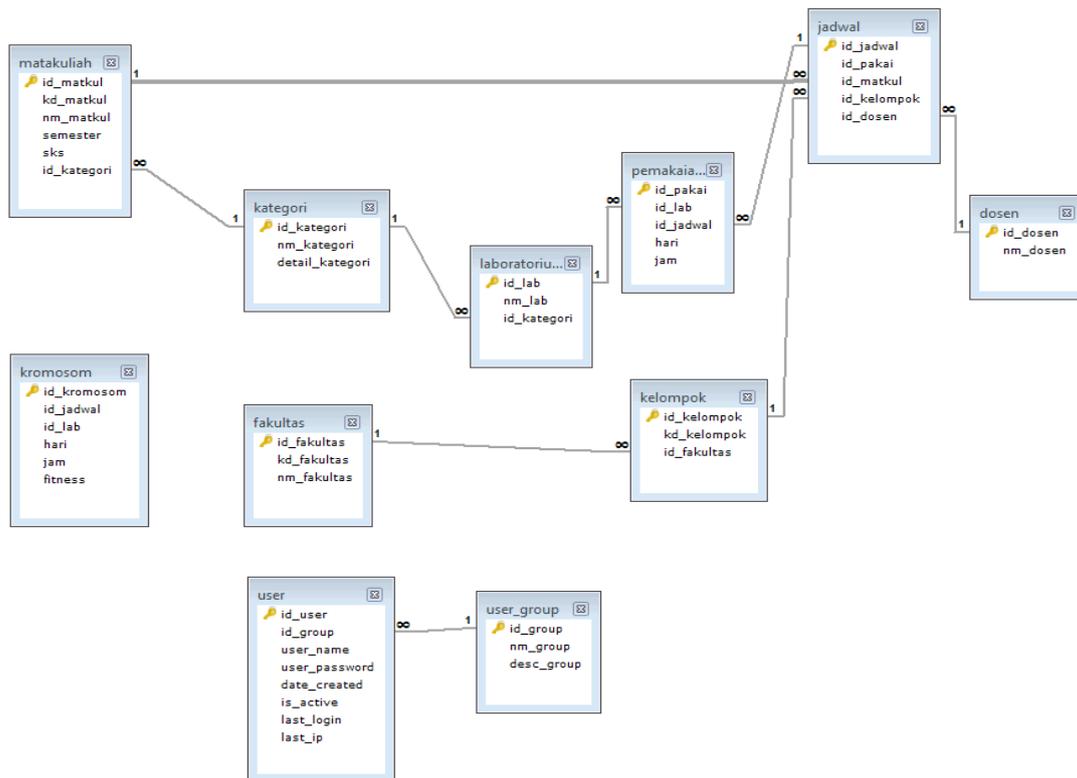
Diagram Class

Diagram *class* di sini menggambarkan hubungan antar *class* dan hubungan *class* dengan *class* pendukungnya. Gambar diagram *class* dapat dilihat pada gambar berikut :



Gambar 6 : Diagram Class

Relasi Tabel



Gambar 7 : Ralasi Antar Tabel

Optimasi menggunakan Algoritma Genetika

Evaluasi *Fitness*

Faktor-faktor yang mempengaruhi evaluasi *fitness* terhadap hasil solusi adalah sebagai berikut :

1. Kesesuaian kategori matakuliah dengan spesifikasi laboratorium.
2. Adanya tabrakan / tidak terhadap jadwal matakuliah yang lain.

Rumus *fitness* yang digunakan adalah sebagai berikut :

$$Fitness = \frac{1}{1 + B1xF1 + B2xF2}$$

dengan :

F1 = Kesesuaian spesifikasi lab dengan matakuliah

F2 = Terjadinya tabrakan dengan jadwal yang lain

B1 = Bobot kesesuaian lab

B2 = Bobot jika terjadi tabrakan

Dari rumus nilai *fitness* di atas dapat terlihat bahwa yang mempengaruhi besar nilai *fitness* adalah harga FN karena harga BN akan tetap selama proses. Jika harga FN semakin besar maka nilai *Fitness* akan semakin kecil. Karena diinginkan solusi yang memiliki nilai *Fitness* yang besar, maka program ini diharapkan tidak terlalu banyak memunculkan faktor-faktor pengaruh ini dalam solusi yang ditawarkan.

Kondisi Selesai

Terdapat tiga kondisi selesai yang dapat menghentikan proses algoritma pemrograman ini, yaitu:

1. Jika setelah beberapa generasi berturut-turut nilai *fitness* terbaik dari populasi tidak mengalami perubahan kembali

2. Jika jumlah generasi atau iterasi maksimum telah tercapai.
3. Jika nilai fitness terbaik minimal telah tercapai.

Jika salah satu kondisi di atas telah diperoleh maka iterasi akan dihentikan. dan jika salah satu kondisi selesai ini belum tercapai maka program akan mengulang kembali proses ini / iterasi dari langkah keempat yaitu evaluasi fitness terhadap populasi baru tadi.

Evaluasi Program

Berikut ini adalah contoh penjadwalan praktikum berdasarkan berdasarkan sampel yang ditentukan.

Tabel 5 :Tabel Jadwal

id_jadwal	id_pakai	id_matkul	id_kelompok	id_dosen
3	0	1	1	6
4	0	1	4	13
5	0	6	1	1
7	0	9	4	10
9	0	6	4	12
...dan seterusnya				

Tabel di atas masih menggunakan kode dari masing-masing matakuliah, kelompok, dan dosen. id_pakai masih belum diisi karena jadwal belum di proses, sehingga belum ada pemakaian laboratorium. Keterangan dari tabel di atas dapat dilihat di tabel 6

Tabel 6 : Data Jadwal Praktikum

Nama Matakuliah	Dosen	Kelompok
DASAR PEMROGRAMAN	IBNU UTOMO WM, S.KOM	1101
DASAR PEMROGRAMAN	SUHARNAWI, M.KOM	1102
JARINGAN KOMPUTER	ABDUSSALAM, S.KOM	1101
BAHASA ASSEMBLY	HANNY HARYANTO, M.T	1102
JARINGAN KOMPUTER	AGUSTINUS T, S.KOM	1102
...dan seterusnya		

Dengan menggunakan data sampel diatas, proses penjadwalan akan dilakukan sesuai dengan inputan user pada proses algoritma genetika.

Langkah selanjutnya adalah menjalankan algoritma genetika

1. Menentukan nilai *fitness* minimal
2. Membuat generasi baru
3. Melakukan evaluasi terhadap populasi
4. Menyusun populasi
5. Jika nilai fitness terbaik kurang dari *fitness* minimal maka lakukan mutasi
6. Looping

Pada percobaan pertama yang telah dilakukan, generasi dengan nilai *fitness* terbaik akan dimasukkan ke tabel kromosom. Seperti pada table 7.

Tabel 7 : Tabel Kromosom

id_kromosom	id_jadwal	id_lab	hari	jam	Fitness
1	3	7	1	4	1
2	4	6	1	7	1
3	5	5	4	5	1
4	7	1	3	8	1
5	9	9	4	5	1
... dan seterusnya					

Setelah ditemukan kromosom dengan *fitness* terbaik, maka dilakukan pengisian pada tabel pemakaian_lab dan *update* pada tabel jadwal.

Tabel 8 : Tabel Pemakaian_lab

id_pakai	id_lab	id_jadwal	hari	jam
1	7	3	1	4
2	6	4	1	7
3	5	5	4	5
4	1	7	3	8
5	9	9	4	5
... dan seterusnya				

Tabel 9 : Tabel jadwal setelah dilakukan optimasi

id_jadwal	id_pakai	id_matkul	id_kelompok	id_dosen
3	1	1	1	6
4	2	1	4	13
5	3	6	1	1
7	4	9	4	10
9	5	6	4	12
...dan seterusnya				

Setelah dilakukan proses optimasi dengan menggunakan algoritma genetika, jadwal matakuliah praktikum berdasarkan sampel di atas menjadi sebagai berikut.

Tabel 10 : Jadwal Praktikum Setelah Optimasi

Nama Matakuliah	Dosen	Kelompok	Lab	Hari/Jam
DASAR PEMROGRAMAN	IBNU UTOMO WM, S.KOM	1101	D.2.H	Senin 12.30 – 14.10
DASAR PEMROGRAMAN	SUHARNAWI, M.KOM	1102	D.2.G	Senin 18.30 – 20.10
JARINGAN KOMPUTER	ABDUSSALAM, S.KOM	1101	D.2.E	Kamis 14.10 – 15.50
BAHASA ASSEMBLY	HANNY HARYANTO, M.T	1102	D.2.A	Rabu 20.10 – 21.50
JARINGAN KOMPUTER	AGUSTINUS T, S.KOM	1102	D.2.J	Selasa 14.10 – 15.0
...dan seterusnya				

Implementasi Sistem

Halaman Utama Koordinator

Sistem Penjadwalan Laboratorium Universitas Dian Nuswantoro

Matakuliah Dosen Kelompok Laboratorium Jadwal User

Tambah Matakuliah Cari Matakuliah Cari

Kode Matakuliah	Nama Matakuliah	Semester	Kategori	Aksi
E1102003	DASAR PEMROGRAMAN	1	Pemrograman	
E123502	PEMROGRAMAN INTERNET	1	Pemrograman	
E123309	PERANCANGAN TERBANTU KOMPUTER	1	Lainnya	
E123704	SIMULASI KOMPUTER BISNIS	1	Lainnya	
E1102042	PEMROGRAMAN INTERNET 1	1	Pemrograman	
E123510	JARINGAN KOMPUTER	1	Jaringan	
E123308	PRAK. PEMROGRAMAN DAN DASAR KOMPUTER	1	Pemrograman	
E1102022	PEMROGRAMAN BERORIENTASI OBYEK	1	Pemrograman	
E1102015	BAHASA ASSEMBLY	1	Pemrograman	
E123911	PRAK. PEMROGRAMAN DAN DASAR KOMPUTER	1	Pemrograman	

Page 1 of 2 pages | Total 19 data ditemukan

Aplikasi Penjadwalan Praktikum Universitas Dian Nuswantoro

Gambar 8 : Halaman Utama Koordinator

Sistem Penjadwalan Laboratorium Universitas Dian Nuswantoro

Matakuliah Dosen Kelompok Laboratorium Jadwal User

Tambah Matakuliah Cari Matakuliah Cari

Kode Matakuliah
 Nama Matakuliah
 Semester Ganjil
 SKS
 Kategori Pemrograman
 Tambah

Pengaturan Akun
 Ubah Password
 Logout

Aplikasi Penjadwalan Praktikum Universitas Dian Nuswantoro

Gambar 9 : Halaman Input Mata Kuliah

Sistem Penjadwalan Laboratorium Universitas Dian Nuswantoro

Matakuliah Dosen Kelompok Laboratorium Jadwal User

Tambah Jadwal Cari Jadwal Cari OPTIMASI JADWAL

Nama Matakuliah DASAR PEMROGRAMAN
 Nama Dosen IBNU UTOMO WM, S.KOM
 Kode Kelompok 1101
 Lab D.2.A
 Hari Selasa
 Jam 08.40 - 10.20
 Simpan

Pengaturan Akun
 Ubah Password
 Logout

Aplikasi Penjadwalan Praktikum Universitas Dian Nuswantoro

Gambar 10 : Halaman Input Jadwal

Sistem Penjadwalan Laboratorium Universitas Dian Nuswantoro

Matakuliah Dosen Kelompok Laboratorium Jadwal

Tambah Jadwal OPTIMASI JADWAL CETAK JADWAL

Daftar Jadwal

Nama Matakuliah	Dosen	Kelompok	Lab	Hari / Jam	Aksi
DASAR PEMROGRAMAN	IBNU UTOMO WM, S.KOM	1101	Belum ada Lab	Belum ada Jadwal	
DASAR PEMROGRAMAN	SUHARNAWI, M.KOM	1102	Belum ada Lab	Belum ada Jadwal	
JARINGAN KOMPUTER	ABDUSSALAM, S.KOM	1101	Belum ada Lab	Belum ada Jadwal	
SIMULASI KOMPUTER BISNIS	WELLIA SHINTA SARI, M.KOM	1101	Belum ada Lab	Belum ada Jadwal	
BAHASA ASSEMBLY	HANNY HARYANTO, S.KOM, M.T	1102	Belum ada Lab	Belum ada Jadwal	
PERANCANGAN TERBANTU KOMPUTER	AGUSTINUS TJAHYONO WIDIGDYO, S.KOM	1102	Belum ada Lab	Belum ada Jadwal	
JARINGAN KOMPUTER	AGUSTINUS TJAHYONO WIDIGDYO, S.KOM	1102	Belum ada Lab	Belum ada Jadwal	
PEMROGRAMAN BERORIENTASI OBYEK	IMANUEL HARKESPAN, S.KOM	1101	Belum ada Lab	Belum ada Jadwal	
SIMULASI KOMPUTER BISNIS	ARIES SETIAWAN, S.KOM	1102	Belum ada Lab	Belum ada Jadwal	
PERANCANGAN TERBANTU KOMPUTER	DE ROSAL IGN MOSES S, S.KOM	1101	Belum ada Lab	Belum ada Jadwal	

Page 1 of 1 pages | Total 10 data ditemukan

Pengaturan Akun
 Ubah Password
 Logout

Aplikasi Penjadwalan Praktikum Universitas Dian Nuswantoro

Gambar 11 : Halaman Jadwal Sebelum Optimasi

Sistem Penjadwalan Laboratorium Universitas Dian Nuswantoro

Matakuliah Dosen Kelompok Laboratorium Jadwal User

Tambah Jadwal Cari Jadwal Cari OPTIMASI JADWAL

Daftar Jadwal

Nama Matakuliah	Dosen	Kelompok	Lab	Hari / Jam	Aksi
DASAR PEMROGRAMAN	IBNU UTOMO WM, S.KOM	1101	D.2.A	Selasa - 08.40 - 10.20	
DASAR PEMROGRAMAN	SUHARNAWI, M.KOM	1102	D.2.A	Selasa - 07.00 - 08.40	
JARINGAN KOMPUTER	ABDUSSALAM, S.KOM	1101	D.2.J	Kamis - 08.40 - 10.20	
SIMULASI KOMPUTER BISNIS	WELLIA SHINTA SARI, M.KOM	1101	D.2.E	Kamis - 12.30 - 14.10	
BAHASA ASSEMBLY	HANNY HARYANTO, S.KOM, M.T	1102	D.2.K	Rabu - 10.20 - 12.00	
PERANCANGAN TERBANTU KOMPUTER	AGUSTINUS TJAHYONO WIDIGDYO, S.KOM	1102	D.2.D	Jumat - 14.10 - 15.50	
JARINGAN KOMPUTER	AGUSTINUS TJAHYONO WIDIGDYO, S.KOM	1102	D.2.E	Rabu - 12.30 - 14.10	
PEMROGRAMAN BERORIENTASI OBYEK	IMANUEL HARKESPAN, S.KOM	1101	D.2.H	Senin - 07.00 - 08.40	
SIMULASI KOMPUTER BISNIS	ARIES SETIAWAN, S.KOM	1102	D.2.E	Selasa - 20.10 - 21.50	
PERANCANGAN TERBANTU KOMPUTER	DE ROSAL IGN MOSES S, S.KOM	1101	D.2.E	Rabu - 16.20 - 18.00	

Page 1 of 1 pages | Total 10 data ditemukan

Pengaturan Akun
Ubah Password
Logout

Aplikasi Penjadwalan Praktikum Universitas Dian Nuswantoro

Gambar 12 : Halaman Jadwal Sesudah Optimasi

KESIMPULAN

Kesimpulan yang dapat diberikan dari hasil implementasi sebagai berikut:

1. Dengan menggunakan aplikasi penjadwalan praktikum, proses penjadwalan menjadi lebih cepat dan efektif. Tidak ada lagi jadwal tabrakan, dan belum adanya jadwal karena keterlambatan penyerahan data.
2. Hasil dari proses optimasi penjadwalan praktikum ini mempermudah koordinator penjadwalan praktikum dalam membuat jadwal matakuliah praktikum yang akan dilaksanakan pada semester yang sedang berlangsung.
3. Pendekatan dengan menggunakan algoritma genetika lebih sesuai digunakan untuk penjadwalan yang lebih kompleks karena kemampuan menangani berbagai macam variabel

DAFTAR PUSTAKA

- [1] Firmansyah, Kiki. (2009). *STUDI ALGORITMA GENETIKA UNTUK OPTIMASI PENJADWALAN MATA KULIAH*. Universitas Pendidikan Indonesia
- [2] Anonim. 2011. Udinus - Universitas Dian Nuswantoro Semarang._Diakses pada : <http://portalsemarang.com/kampus/universitas-dian-nuswantoro-udinus/> tanggal 1-11-2011 pukul 10:00
- [3] Anonim. 2011. Genetic_algorithm. Diakses pada : http://en.wikipedia.org/wiki/Genetic_algorithm tanggal 1-11-2011 pukul 09:00

- [4] Gen, Mitsuo, Cheng, Runwei. (2000). *Genetic Algorithms and Engineering Optimization*. New York : Wiley-Interscience publication.
- [5] Herjanto, Eddy. (2007). *Manajemen Operasi* (Edisi Ketiga). Jakarta : Grasindo
- [6] Haupt, R.L, Haupt, S. (2004). *Practical Genetic Algorithm*. Canada : John Willey & Sons.Inc
- [7] Kusumadewi, Sri. (2003). *Artificial Intelligence (Teknik & Aplikasinya)*. Yogyakarta: Graha Ilmu.
- [8] Juniawati. (2003). *Implementasi Algoritma Genetika untuk Mencari Volume Terbesar Bangun Kotak Tanpa Tutup dari Suatu Bidang Datar Segi Empat*. Universitas Surabaya
- [9] Pressman, Roger S. (2002). *Rekayasa Perangkat Lunak (Pendekatan Praktisi – Buku Satu)*.Edisi keempat.Yogyakarta: Andi