SISTEM PENJADWALAN CERDAS DENGAN SIMULATED ANNEALING

M. Faris Farhan[1], Galih Hermawan, S.Kom, M.T.[2]

[1] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
E-mail : ffrhn07@gmail.com[1],galih.hermawan@gmail.com[2]

ABSTRAK

Penjadwalan mata kuliah pada sebuah universitas atau perguruan tinggi, merupakan hal yang sangat penting untuk keberlangsungan perkuliahan di setiap semesternya. Untuk menyelesaikan permasalahan penjadwalan maka diperlukan suatu algoritma optimasi untuk mengerjakan penjadwalan mata kuliah ini. Dalam penelitian ini digunakan algoritma Simulated Annealing (SA). Semua data-data yang dibutuhkan dimasukkan ke dalam aplikasi dan nantinya akan diproses.

Berdasarkan permasalahan yang telah diuraikan, maka solusi yang diusulkan adalah membuat sistem penjadwalan cerdas yang mampu membuat jadwal mata kuliah dengan algoritma Simulated Annealing dengan terhindar dari pelanggaran hard constraint dan soft constraint. Adapun data yang digunakan mengacu dataset dan mekanisme yang digunakan mengacu pada penjadwalan di prodi IF UNIKOM. Uji coba yang dilakukan adalah melihat seberapa cepat algoritma ini dalam membuat jadwal mata kuliah.

Hasil pengujian yang diperoleh adalah sebuah jadwal mata kuliah dalam satu semester tanpa adanya pelanggaran terhadap hard constraint dan soft constraint yang diterapkan. Berdasarkan pengujian dapat ditarik kesimpulan bahwa algoritma Simulated Annealing dapat dengan cepat memproses pembuatan jadwal dengan waktu rata-rata 10 detik, namun tingkat keakuratan yaitu sebesar 60% saja.

Kata kunci : penjadwalan mata kuliah, algoritma simulated annealing, sistem penjadwalan cerdas.

1. PENDAHULUAN

Penjadwalan mata kuliah pada sebuah universitas atau perguruan tinggi, merupakan hal yang sangat penting untuk keberlangsungan perkuliahan di setiap semesternya. Hal ini merupakan salah satu masalah yang dihadapi oleh setiap universitas dan perguruan tinggi saat ini. Penjadwalan mata kuliah merupakan hal yang perlu diperhitungkan dengan baik mengingat banyaknya kondisi yang harus dipenuhi seperti banyaknya mata kuliah, keterbatasan jumlah dosen, keterbatasan waktu/hari mengajar dan terbatasnya ruang kelas [1]. Dari banyaknya masalah dan batasan yang ada, bisa disimpulkan tidak boleh ada dosen, kelas atau ruangan yang muncul di waktu yang sama [2].

Untuk menyelesaikan permasalahan penjadwalan maka diperlukan suatu algoritma optimasi untuk mengerjakan penjadwalan mata kuliah ini. Dalam penelitian ini digunakan algoritma Simulated Annealing (SA) karena dari hasil penelitian yang membandingkan Algoritma Genetika dan Simulated Annealing dalam penjadwalan flow shop, ditinjau dari segi performa dan waktu proses dengan kriteria minimasi makespan dan total flowtime, menunjukan hasil simulasi Simulated Annealing lebih unggul dari dari Algoritma Genetika hingga 90%. Algoritma Genetika hanya unggul dalam waktu proses saja namun dengan tren waktu proses yang terbentuk diyakini pada problem dengan kombinasi job dan mesin yang banyak, algoritma Simulated Annealing dapat lebih cepat daripada Algoritma Genetika [5]. Simulated Annealing dikembangkan berdasarkan ide dari mekanisme perilaku pendinginan dan proses kristalisasi (annealing) material panas. Algoritma ini melakukan peningkatan iteratif untuk memperbaiki solusi yang dihasilkan teknik-teknik penjadwalan heuristik, dalam hal ini adalah sebuah solusi awal yang dibuat dengan teknik heuristic ataupun random, diiterasi secara berulang dengan metode annealing dengan menggunakan perturbasi lokal hingga tidak ada peningkatan lagi atau hingga jumlah iterasi yang diinginkan sudah dicapai [5].

Penelitian lain yang menggunakan Algoritma Simulated Annealing dalam menentukan urutan penjadwalan produksi mengatakan bahwa Algoritma Simulated Annealing memberikan hasil yang signifikan dibandingkan dengan metode Brute force dan metode FIFO (First In First Out), denga kesimpulan bahwa lama waktu proses dari SA lebih lama dibandingkan dengan FIFO, tetapi makespan yang dihasilkan lebih singkat, dan lama proses SA lebih singkat dari Brute Force tetapi makespan yang dihasilkan 100% sama [6].

Berdasarkan permasalahan dan keterbatasan tersebut maka harapan dari penelitian ini adalah terciptanya sebuah sistem penjadwalan cerdas yang mampu meng-generate jadwal kuliah yang terhindar

dari constraint-constraint sehingga kegiatan perkuliahan dapat berlangsung setiap semesternya.

2. ISI PENELITIAN

2.1 Definisi Penjadwalan

Penjadwalan berkaitan dengan segala aktivitas untuk membuat sebuah jadwal. Dalam kamus Collins Concise (4th Edition) menyatakan bahwa jadwal adalah susunan tabel kegiatan berdasarkan waktu dan tempat. Kegiatan ini biasanya pertemuan antar beberapa pihak pada waktu dan tempat yang sama. Jadwal harus memenuhi beberapa persyaratan dan memenuhi keinginan semua pihak yang terlibat sebaik mungkin. Waktu dari aktivitas harus disusun sedemikian rupa sehingga tidak ada salah satu komponen pun memliki lebih dari satu kegiatan pada waktu yang sama [7].

Penjadwalan kuliah merupakan pengaturan perencanaan belajar mengajar yang meliputi mata kuliah, dosen, waktu dan ketersediaan ruangan. Pada umumnya penjadwalan kegiatan belajar mengajar disajikan dalam sebuah tabel hari dalam seminggu yang terdiri dari slot waktu yang terdiri dari mata pelajaran, hari, jam, serta pengajar yang sesuai dengan mata pelajaran yang diajarkan [8].

2.2 Batasan dalam Penjadwalan

Dalam masalah penjadwalan memiliki beberapa macam batasan yang dapat menyebabkan output yang dihasilkan menjadi salah. Dalam menerapkan batasan dalam suatu masalah, biasanya tidak terlalu sama untuk setiap masalah [8]. Batasan tersebut terdiri dari :

1. Edge constraint

Edge constraint adalah batasan yang mengatur dua kejadian tidak boleh menempati satu slot waktu yang sama. Contohnya pada hari Senin jam 07.30 sampai 08.10 tidak mungkin guru A mengajar di ruang kelas 1A dan mengajar di ruang kelas 5A.

2. Ordering constraint

Ordering constraint adalah batasan yang menjaga urutan kejadian dalam timetable. Contoh mata pelajaran A harus dilakukan sebelum mata pelajaran B. Biasanya masalah batasan ini jarang digunakan karena dapat menyebabkan penjadwalan menjadi lebih rumit.

3. Event-spread constraint

Event-spread constraint adalah batasan yang mengatur penyebaran kejadian pada suatu timetable. Contoh dalam menjalani ujian, seorang siswa diperbolehkan max 3 mata pelajaran yang diujikan dalam 1 hari.

4. Present specification and exclusion

Present specification and exclusion adalah menentukan terlebih dahulu slot waktu yang akan digunakan oleh suatu kejadian sebelum proses pencarian solusi dilakukan. Contoh mata pelajaran agama digabung dengan kelas 1-A dan 1-B maka akan ditentukan waktu yang sama untuk kedua kelas tersebut agar tidak terganggu
untuk menyusun mata pelajaran yang lain.

5. Capacity constraint

Capacity constraint adalah batasan yang berhubungan dengan kapasitas ruangan. Untuk masing-masing kelas hanya boleh diisi sebanyak 40 siswa.

6. Hard and soft constraint

Hard constraint adalah batasan yang sama sekali tidak boleh dilanggar, sedangkan soft constraint adalah batasan yang diusahakan semaksimal mungkin tidak dilanggar namun jika dilanggar, hal tersebut masih dapat diterima.

2.3 Definisi Kecerdasan Buatan

Kecerdasan buatan merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Pada awal diciptakannya, computer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan zaman, maka peranan computer semakin mendominasi kehidupan umat manusia. Computer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu komputer diharapkan dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan manusia.

Pengertian kecerdasan buatan dapat dipandang dari berbagai sudut pandang antara lain :

a. Sudut pandang kecerdasan

Kecerdasan buatan akan membuat mesin menjadi cerdas (mampu berbuat seperti apa yang dilakukan oleh manusia)

b. Sudut pandang penelitian

Kecerdasan buatan adalah suatu studi bagaimana membuat agar computer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia

c. Sudut pandang bisnis

Kecerdasan buatan adalah kumpulan peralatan yang sangat powerful dan metodologis dalam menyelesaikan masalah-masalah bisnis.

d. Sudut pandang pemograman

Kecerdasan buatan meliputi studi tentang pemograman simbolik, penyelesaian masalah (problem solving) dan pencarian (searching).

2.4 *Simulated Annealing*

Simulated Annealing adalah salah satu algoritma untuk untuk optimisasi yang bersifat generic. Diperkenalkan oleh Gelett Kirkpatrick dan Vechi(1983) dan Cerny (1985). Algoritma ini berbasiskan probabilitas dan mekanika statistik untuk mencari pendekatan terhadap solusi optimum global dari suatu permasalahan.

Simulated Annealing dikembangkan berdasarkan ide dari mekanisme perilaku pendinginan dan proses kristalisasi (annealing) material panas. Algoritma ini melakukan

peningkatan iteratif untuk memperbaiki solusi yang dihasilkan teknik-teknik penjadwalan heuristik, dalam hal ini adalah sebuah solusi awal yang dibuat dengan teknik heuristic ataupun random, diiterasi secara berulang dengan metode annealing dengan menggunakan perturbasi lokal hingga tidak ada peningkatan lagi atau hingga jumlah iterasi yang diinginkan sudah dicapai [5]. Simulated Annealing adalah satu dari algoritma – algoritma terbaik saat ini dalam hampir semua pemecahan masalah optimasi. Dengan sedikit modifikasi, Simulated Annealing akan bekerja sangat baik dalam semua permasalahan optimasi [6].

2.5 Algoritma *Simulated Annealing*

Struktur algoritma Simulated Annealing secara umum adalah sebagai berikut [12]:

1. Pilih solusi awal *Sol* secara acak.
2. Pembobotan *E=f(Sol).*
3. Bangkitkan solusi baru *Solnew*
4. Hitung Bobot = *f(Solnew).*
5. Jika *f(solnew) > f(Sol)* maka *Sol* = Solnew dan *E = Enew.*
6. Berhenti jika telah mencapai kriteria pemberhentian, jika tidak kembali ke langkah 3.

Pada penelitian ini, langkah ke 2 diambil dari nilai pembobotan yang diperoleh dari pengecekan constraint-constraint. Modifikasi penelitian dilakukan pada tahap 3-7, yaitu untuk Solnew merupakan solusi yang memiliki nilai pembobotan yang buruk (langkah 3), sehingga akan dilakukan pembobotan lagi (langkah 4 & 5). Jika masih ada solusi buruk makan dilakukan pengacakan solusi antara solusi baik dan buruk, hal ini dilakukan untuk memperbanyak solusi yang diperoleh.

2.6 Analisis Masalah

Analisis masalah dapat didefinisikan suatu gambaran masalah yang diangkat dalam penulisan skripsi tentang optimasi penjadwalan menggunakan algoritma *Simulated Annealing*. Masalah penjadwalan kuliah merupakan permasalahan yang kompleks karena adanya berbagai komponen yang terdiri dari mata kuliah, dosen, ruang kelas, dan waktu kegiatan perkuliahan dengan dengan melihat juga dataset batasan jadwal mengajar dosen dimana dataset dan mekanisme yang digunakan mengacu pada penjadwalan di prodi IF UNIKOM. Dimana batasan-batasan tersebut tidak boleh ada yang dilanggar atau disebut *hard constraint.* Ada pula batasan-batasan yang boleh dilanggar atau disebut *soft constraint.*

2.7 Analisis Data Masukan

Data masukan yang akan di analisis di sini berdasarkan data dari prodi IF Unikom. Data yang berkaitan dengan jadwal adalah :

1. Data Mata Kuliah
   Data mata kuliah yang diambil adalah mata kuliah pada semester genap tahun ajaran 2015/2016
2. Data Dosen
   Data dosen yang diambil adalah dosen yang mengajar pada mata kuliah semester genap tahun ajaran 2015/2016.
3. Data Ruangan
4. Data Dosen Mata Kuliah
   Dosen mata kuliah adalah mata kuliah yang diajarkan oleh tiap dosen.

2.8 Aturan Penjadawalan

Untuk mendapatkan sebuah jadwal mata kuliah yang optimal dibutuhkan batasan-batasan atau aturan dalam membuat penjadwalan. Jika semua aturan yang ditentukan dapat terpenuhi maka penjadwalan mata kuliah dapat dikatakan optimal. Berikut ini adalah batasan-batasan (*hard constraint dan soft constraint*) yang akan diterapkan dalam membuat penjadwalan :

a) *Hard constraint* dalam penjadwalan prodi IF Unikom adalah sebagai berikut :

i. Bentrok ruangan dan waktu mengajar, ketika jam, hari atau ruangan terisi dalam waktu yang sama.
ii. Bentrok dosen, ketika dosen mengajar di jam yang sama, hari yang sama atau dosen yang sama.
iii. Bentrok dengan waktu shalat jumat.
iv. Bentrok waktu keinginan dosen.

b) Soft constraint (batasan lunak) dalam penjadwalan prodi IF Unikom adalah sebagai berikut:

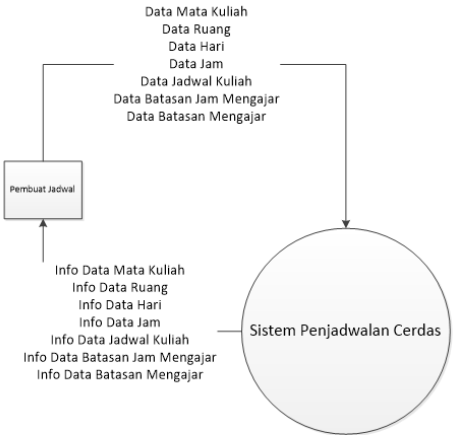
1. Dosen yang memiliki batasan waktu mengajar perkuliahan

Implementasi batasan-batasan tersebut dapat dilakukan dengan memberikan bobot nilai tertentu untuk setiap batasan yang tidak dapat terpenuhi.

2.9 Desain Sistem

Desain sistem diperlukan untuk keperluan iden-tifikasi dan analisa permasalahan. Desain sistem yang ada akan dijadikan sebagai rujukan dalam pembuatan sistem agar sesuai dengan kebutuhan. Model pe-rancangan sistem yang digunakan dalam penelitian ini adalah:

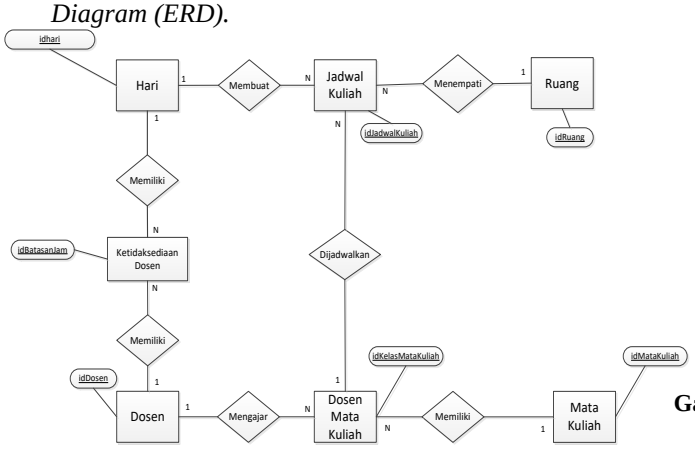
1. Diagram Konteks(DFD Level 0)
   Diagram konteks digunakan untuk menggambarkan sistem pertama kali secara garis besar. Diagram konteks juga merupakan diagram arus data (*data flow diagram (*DFD*)*) dengan level yang teratas (top level).



Gambar 1. Diagram Konteks

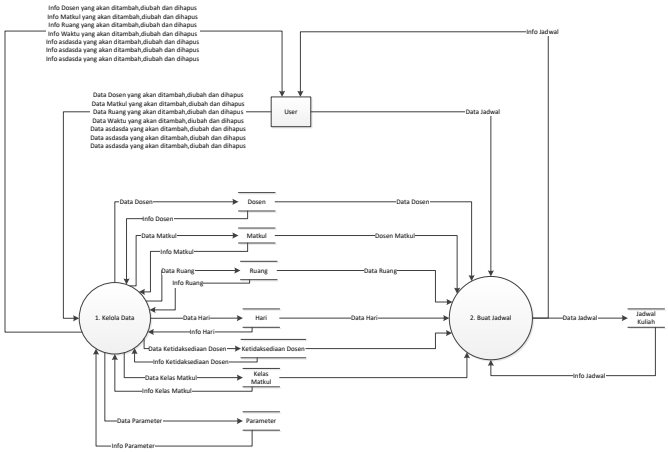
2. *Entity Relationship Diagram(ERD)*

Analisis basis data merupakan analisis kumpulan data yang saling berhubungan dan disimpan dalam suatu media penyimpanan tertentu tanpa pengulangan (*redundancy*). Terdapat data-data yang akan digunakan dalam proses pembuatan perangkat lunak sistem informasi dengan pendekatan *supply chain management*. Data-data yang diperoleh akan digunakan untuk mendesain basis data disertai dengan atribut-atribut dari masing-masing entitas serta relasi antar entitas-entitas tersebut. Dalam merancang basis data, model data yang akan digunakan adalah *Entity Relationship Diagram (ERD).*



Gambar 2. ERD

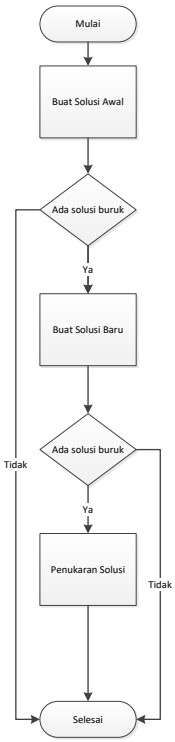
3. *DFD level 1*



Gambar 3. DFD Level 1

2.10 Analisis Algoritma *Simulated Annealing*

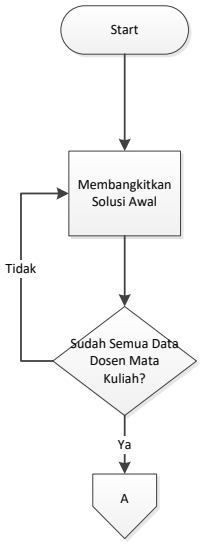
Tahapan Proses Simulated Annealing terdapat pada Gambar 4.



Gambar 4. Proses Algoritma Simulated Annealing

2.10.1 Membuat Solusi Awal

Dalam menetapkan jadwal awal akan dilakukan pembangkitan jadwal yang terdiri dari Dosen, Mata Kuliah, Jam, Hari dan Ruang. Adapun urutan prosesnya pada Gambar 5:



Gambar 5. Proses Membuat Solusi Awal

Untuk contoh solusi baru dapat dilihat pada Tabel 1.

Nomor	Kode_MK	Kode_dosen	Nama_kelas	Hari	Ruang	Jam Mulai	Jam Selesai
1	1	1	B.IND-1	1	R610	07:00:00	08:30:00
2	1	1	B.IND-2	1	R2505	07:00:00	08:30:00
3	1	1	B.IND-3	2	R610	08:30:00	10:00:00
4	1	1	B.IND-4	3	R5404	10:00:00	11:30:00
…	…	…	…	…	…	..	..
325	25	74	KSI-K	2	R5305	13:00:00	15:15:00

Tabel 1. Contoh Solusi Awal

2.10.2 Cek *Constraint* dan Pembobotan

Sebuah jadwal dikatakan belum optimal jika masih terdapat pelanggaran. Pelanggaran didapati dari bobot constraint yang diterapkan (Hard Constraint maupun Soft Constraint). Setiap terjadi pelanggaran makan akan diberi nilai dengan bobot +1. Rumus untuk menentukan bobot adalah sebagai berikut:

Total Bobot=Bobot 1 + Bobot 2 + Bobot 3 + Bobot 4

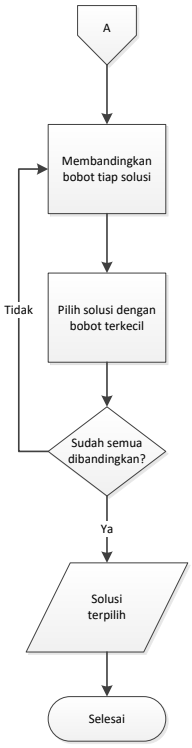
2.10.3 Menetapkan Temperatur Awal, Nilai Reduksi Suhu dan Jumlah Iterasi

Sebelum ketahap proses pencarian solusi, 3 komponen ini harus ditetapkan terlebih dahulu. Disini akan ditetapkan seperti berikut:

a. Temperatur awal disesuaikan dengan banyak dosen mata kuliah yang akan dijadwalkan, dalam kasus ini yaitu sebanyak 325 (sebanyak dosen mata kuliah).
b. Nilai reduksi Suhu adalah banyak ruang yang tersedia dalam kasus ini yaitu 19 ruangan, jadi setiap kali ruangan penuh maka banyak ruangan akan berkurang hingga ruangan atau mata kuliah sudah terpilih semua.
c. Jumlah iterasi = 1.

2.10.4 Membuat Solusi Baru

Dalam Simulated Annealing jika jadwal awal sudah merupakan jadwal yang optimal (tidak ada pelanggaran yang terjadi) maka yang digunakan adalah jadwal awal tersebut, sehingga tidak perlu melakukan pencarian jadwal baru. Jika tidak, maka akan dilakukan pencarian jadwal baru dengan pencarian struktur ketetanggaan. Jadi, jika jadwal yang melanggar hari senin, makan akan dicari tetangganya yaitu hari selasa, rabu, dst. Solusi dengan bobot paling kecil akan ditukar posisinya dengan bobot yang lebih besar. Untuk proses pembuatan solusi baru dapat dilihat pada Gambar 6.



Gambar 6. Proses Pembuatan Solusi Baru

Untuk contoh solusi baru dapat dilihat pada Tabel 2.

Nomor	Kode_MK	Kode_dosen	Nama_kelas	Hari	Ruang	Jam Mulai	Jam Selesai
1	1	1	B.IND-1	1	R610	07:00:00	08:30:00
2	1	1	B.IND-2	1	R2505	07:00:00	08:30:00
3	1	1	B.IND-3	2	R610	08:30:00	10:00:00
4	1	1	B.IND-4	3	R5404	10:00:00	11:30:00
…	…	…	…	…	…	..	..
325	25	74	KSI-K	2	R5305	13:00:00	15:15:00

Tabel 2. Contoh Solusi Baru

2.11 Implementasi Perangkat Pendukung

Spesifikasi perangkat keras yang digunakan untuk membangun aplikasi ini yaitu sebagai berikut:

1. *Processor 3.00 (4 CPUs)* GHz.
2. RAM 4 GB.
3. Layar dengan resolusi 1366 x 768 pixel.
4. *Keyboard* standar.
5. *Mouse* standar.

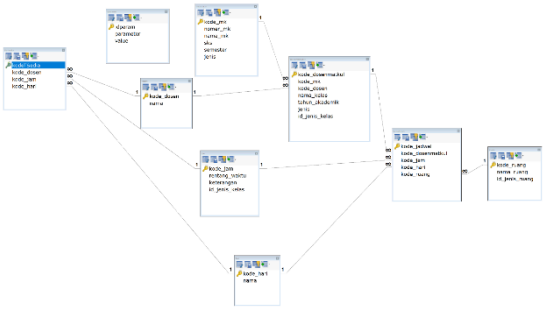
2.12 Implementasi Perangkat Pendukung

Spesifikasi perangkat lunak yang digunakan untuk membangun aplikasi ini yaitu sebagai berikut:

1. Sistem operasi Windows 7 Ultimate 32 Bit.
2. Bahasa pemrograman C#.
3. DBMS MySQL 5.5.

2.13 Skema Relasi

Skema relasi digunakan untuk menggambarkan relasi dari setiap tabel di dalam database yang akan dirancang.



Gambar 7. Skema Relasi

2.14 Perancangan Struktur Menu

Perancangan arsitektur merupakan hubungan di antara elemen-elemen struktural utama dari program. Perancangan arsitektur dapat memberikan gambaran mengenai struktur program.



Gambar 8 Struktur Menu

2.15 Hasil Penelitian

Algoritma yang digunakan dalam pembuatan program penjadwalan ini adalah Agoritma Simulated Annealing (SA).

Prinsip kerja algoritma simulated annealing adalah sebagai berikut:

1. Pertama adalah proses memasukan dataset yang nantinya akan di proses, seperti data dosen, mata kuliah, ruangan, lot waktu, waktu ketersediaan dosen. Setelah semua data dimasukan, maka sistem dapat membuat jadwal.
2. Kedua adalah proses algoritma simulated annealing.
   a. Langkah awal yang dilakukan adalah membangkitkan solusi awal dengan mengambil data dosen mata kuliah, ruang, hari dan waktunya, lalu dihitung dan dibandingkan bobotnya.
   b. Proses a dilakukan hingga seluruh data terjadwal.
   c. Setelah membangkitkan Solusi Awal, apabila terdapat constraint yang dilanggar maka dilanjut ke pembuatan Solusi Baru tapi jika seluruh constraint tidak ada yang dilanggar maka proses selesai
   d. Proses c diulangi hingga semua mata kuliah diberi bobot.
   e. Jika masih ada Solusi Buruk maka akan dilakukan pertukaran solusi dengan cara mengambil satu solusi baik dan satu solusi buruk secara random dan dilakukan pengecekan constraintnya, hal ini dilakukan hingga tidak ada lagi solusi buruk.
3. Ketiga adalah hasil dari pemrosesan algoritma *simulated annealing.*

2.15.1 Pengujian Algoritma Simulated Annealing

Pengujian algoritma dilakukan untuk mengamati seberapa cepat algoritma simulated annealing dalam membuat jadwal dengan constraint yang telah ditentukan. Hasil dari pengujian bisa dilihat di Tabel 3

Percobaan	Waktu yang dibutuhkan (detik)
1	10
2	8
3	11
4	9
5	14
Presentasi tingkat keberhasilan tiap pengujian	**60%**
Kecepatan Proses Rata-rata	00:00:10

Tabel 3 Hasil Pengujian

2.16 Kesimpulan Pengujian

Berdasarkan hasil pengujian yang ditampilkan pada tabel 3 dapat disimpulkan meskipun data yang diujikan sama, tetapi memiliki variasi waktu penyelesaian yang berbeda. Hal ini dikarenakan proses pengambilan data yang dilakukan secara random.

3. PENUTUP

Berdasarkan hasil yang didapat pada penelitian ini, diperoleh informasi penggunaan algoritma simulated annealing dapat menyelesaikan permasalahan penjadwalan meskipun hasil yang diperoleh belum optimal. Proses pencarian yang dilakukan secara random dapat mempengaruhi dalam lamanya algoritma dalam mencari solusi, dalam penelitian ini ditemukan rata-rata kecepatan proses dari 5 kali percobaan adalah 10,4 detik .

Adapun saran untuk pengembangan sistem selanjutnya, yaitu:

Perbaikan dalam penentuan temperature awal, nilai reduksi dan iterasi.

DAFTAR PUSTAKA

[1] A. Wasfy and F. Aloul, “Solving the university class scheduling problem using advanced ILP techniques,” IEEE GCC Conf., 2007.

[2] E. Aycan and T. Ayav, “Solving the Course Scheduling Problem Using Simulated Annealing,” 2009 IEEE Int. Adv. Comput. Conf., no. March, pp. 6–7, 2009.

[3] C. F. M. T. Muhammad Hamdani Azmi, Sugiono, “Penjadwalan Produksi Rokok Untuk

Meminimalkan Maximum Tardiness Menggunakan Algoritma Simulated Annealing (Studi Kasus di PR. Adi Bungsu Malang)" J. rekayasa dan Manaj. Sist. Ind., vol. 3, no. 2, pp. 353–362, 2015.

[4] H. A. S, C. Farela, and M. Tantrika, "Implementasi Algoritma Simulated Annealing Pada Penjadwalan Produksi Untuk Meminimasi Makespan (Studi Kasus di PT . Gatra Mapan , Karang Ploso , Malang)," J. Rekayasa Dan Manaj. Sist. Ind., vol. 3, no. 1, pp. 43–52, 2015.

[5] A. Pamungkas, "Perbandingan Kinerja Algoritma Genetika Dan Simulated Annealing Untuk Masalah Multiple" J.Teknik Industri.,vol. 4, no. 1, pp. 26–35, 2002.

[6] L. W. Santoso, J. Guntara, and I. N. Sandjaja, "Penjadwalan Produksi Dengan Menggunakan Algoritma Simulated Annealing," J. Ilm. Ilmu Komput., vol. 9, pp. 1–2, 2012.

[7] Willemen, R. J. School Timetable Construction : Algorithms and Complexity. Thesis. Netherlands: Universiteitsdrukkerij Technische Universiteit Eindhoven, 2002.

[8] Arviani, Yelly.Algoritma Ant Colony System Dalam Penjadwalan Kegiatan Belajar Mengajar Di Sekolah Dasar. Skripsi. Ilmu Komputer. Universitas Sumatera Utara, 2013.

[9] [9] L. D. B. K. C. D. Jeffery L. Whitten, Sistems Analysis and Design Methods, Tokyo: McGraw-Hill Education, 2004.

[10] S. Kusumadewi, "Artificial Intelligent (Teknik dan Aplikasinya)," Yogyakarta, Graha Ilmu, 2003.

[11] Arief, M.Rudianto. Pemrograman Web Dinamis Menggunakan PHP dan MySQL. Yogyakarta: Penerbit Andi, 2011.

[12] C. E. Nugraheni and L. Abednego, "Penerapan Algoritma Simulated Annealing Untuk Sidang Seminar," in SNASTI 2006, p. III-11, 2006.

[13] R. A.S, dan M. Shalahuddin, Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur & Berorientasi Objek), Bandung : Modula, 2011.