

LAPORAN PENELITIAN PNBP



Exhaustive Search dengan Distributed Processing untuk permasalahan penjadwalan pada Jurusan Teknik Informatika UNG

Oleh:

Moh. Syafri Tuloli, ST, MT

UNIVERSITAS NEGERI GORONTALO

September 2012

HALAMAN PENGESAHAN LAPORAN AKHIR

1. Judul Usulan : Exhaustive Search dengan Distributed Processing untuk permasalahan penjadwalan pada Jurusan Teknik Informatika UNG
2. Ketua Peneliti
 - a) Nama lengkap : Moh. Syafri Tuloli, MT
 - b) Jenis Kelamin : Laki-laki
 - c) NIP : 198207252008121004
 - d) Jabatan Struktural : Kepala Laboratorium Jurusan Teknik Informatika
 - e) Jabatan Fungsional : Tenaga Pengajar
 - f) Fakultas / Jurusan : Teknik / Jurusan Teknik Informatika FT UNG
 - g) Pusat Penelitian : Program Studi Sistem Informasi
 - h) Alamat : Fakultas Teknik Universitas Negeri Gorontalo, Jl. Jenderal Sudirman No. 6 Kota Gorontalo.
 - i) Telpon/Faks : (0435) 821183
 - j) Alamat Rumah : Jln. Sawah Besar No.10, RT.01, RW. 05, Kota Timur, Kota Gorontalo
 - k) Telpon/Faks/E-mail : 081314270499/ - /ekohabit@gmail.com
3. Jangka Waktu Penelitian : 6 bulan
4. Pembiayaan

Jumlah biaya yang diajukan: Rp. **5.000.000,-**

Gorontalo, 15 September 2012

Mengetahui,

Dekan Fakultas Teknik UNG

Ketua Peneliti,

Ir. Rawiyah Husnan, MT
NIP. 19640427 1994032001

Moh. Syafri Tuloli, ST, MT
NIP. 198207252008121004

Menyetujui,

Ketua Lembaga Penelitian UNG,

Dr. Fitryane Lihawa, M.Si
NIP. 19691209 199303 2001

ABSTRAK

Permasalahan pengaturan jadwal perkuliahan (Universal Time Tabling Problem) adalah permasalahan yang selalu terjadi pada Jurusan Teknik Informatika UNG. Permasalahan ini menjadi kompleks karena dalam penentuan jadwal harus memperhatikan konstrain yang berkaitan dengan dosen, mahasiswa, dan dengan ruangan.

Pendekatan pencarian menggunakan exhaustive search merupakan pendekatan yang memiliki kemungkinan paling tinggi dalam menghasilkan solusi terbaik. Tetapi metode ini dihindari, karena isu kecepatan dan memori. Tetapi dengan semakin berkembangnya teknologi perangkat keras dan perangkat lunak, dan untuk permasalahan dalam domain yang terbatas (Jurusan Teknik Informatika UNG) yang memerlukan solusi yang seoptimal mungkin, bisa jadi pemecahan menggunakan metode ini adalah yang terbaik.

Distributed processing adalah salah satu perkembangan teknologi perangkat lunak yang memungkinkan penggunaan sumber daya pemrosesan yang terhubung melalui jaringan untuk melakukan suatu tugas.

Penelitian ini membuktikan bahwa penerapan distributed processing memang dapat meningkatkan proses pencarian solusi menggunakan metode exhaustive search, tetapi peningkatan proses tersebut tidak sebanding dengan total durasi waktu yang dibutuhkan untuk mencari solusi. Sehingga penggunaan exhaustive search tidak tepat untuk memecahkan masalah penjadwalan di teknik Informatika UNG.

Kata Kunci : Universal TimeTabling Problem, exhaustive search, *distributed processing*

ABSTRACT

Universal Time Tabling Problem is a recurring problem in Teknik Informatika UNG. This problem so complex due to strict constraints related to lecturer, student, and with room.

Exhaustive search is a searching method that have a highest probability to get the optimal solution. But this method is rarely used for its high memory usage and very slow in searching speed. But with the technology advance of lately hardware and software, and the need of optimal solution in the Teknik Informatika UNG, this method become a strong candidate to use to solve this problem.

Distributed processing is one of technology that used the computing resources in the network to do a computing task.

This research proves that the usage of distributed processing improves the searching speed of exhaustive search method, but the improved speed is not significant due to the overly high overall search time needed to explore all of the solution, hence the usage of exhaustive search is not good to use to the timetabling problem in Teknik Informatika UNG.

Keyword: Universal TimeTabling Problem, exhaustive search, distributed processing.

RINGKASAN

Permasalahan pengaturan jadwal adalah permasalahan yang selalu terjadi pada setiap semester pada setiap fakultas/jurusan/prodi pada lingkungan UNG, termasuk pada lingkungan Jurusan Teknik Informatika UNG. Beberapa aplikasi telah digunakan dan dikembangkan untuk memecahkan permasalahan ini, tapi terdapat beberapa kekurangan pada aplikasi-aplikasi ini yang menyulitkan dalam membuat jadwal. Diantaranya adalah: tidak memiliki fasilitas untuk memperbaiki tabrakan pada jadwal dan tidak menghasilkan jadwal yang paling optimal (hanya menghasilkan jadwal yang memiliki tabrakan paling minimal yang bias ditemukan).

Pendekatan exhaustive search adalah pendekatan yang bisa mengatasi kedua masalah tersebut. Metode ini mampu untuk mencarikan solusi apabila solusi yang ada masih memiliki tabrakan, metode ini juga diyakini mampu untuk mencari solusi yang paling optimal jika dibandingkan dengan metode pencarian yang lain. Tetapi metode ini memiliki satu kelemahan yang fatal, yaitu memiliki ruang pencarian yang paling besar diantara semua metode pencarian, hal ini dikarenakan metode ini harus mengimplementasikan proses pencarian yang memastikan bahwa semua kemungkinan solusi harus bias dibangkitkan. Kelemahan ini membuat metode pencarian ini adalah metode yang tidak favorit untuk digunakan pada permasalahan penjadwalan.

Distributed processing adalah suatu teknologi untuk mempercepat pelaksanaan tugas komputasi dengan menggunakan sumber daya berupa komputer-komputer yang saling terhubung dalam suatu jaringan. Distributed processing bekerja dengan cara membagi-bagi suatu tugas pemrosesan menjadi sub-sub tugas yang lebih kecil, kemudian menugaskan komputer-komputer dalam jaringan untuk melaksanakan tugas tersebut secara paralel. Pelaksanaan tugas yang dilakukan secara paralel dan secara bersama-sama oleh beberapa computer inilah yang membuat pelaksanaan tugas komputasi menjadi lebih cepat. Distributed processing atau distributed computing sangat bias digunakan pada lingkungan Teknik Informatika UNG dikarenakan tersedianya banyak sumber daya komputasi (komputer, server, smartphone, dll) yang bias dimanfaatkan.

Penelitian ini bertujuan untuk menggabungkan manfaat yang didapatkan dari penggunaan metode Exhaustive Search dan percepatan pemrosesan dari teknologi Distributed Processing

untuk memecahkan permasalahan Penjadwalan pada lingkungan Jurusan Teknik Informatika UNG. Diharapkan dengan menggabungkan kedua metode tersebut akan menghasilkan suatu disain perangkat lunak yang mampu untuk menghasilkan solusi penjadwalan yang paling optimal dari yang mungkin didapatkan, tetapi dalam durasi waktu komputasi yang masuk akal.

KATA PENGANTAR

Segala puji syukur kehadirat Allah S.W.T atas keberkahan dan rahmat-Nya yang membuat laporan penelitian ini dapat diselesaikan.

Diucapkan terimakasih yang sebanyak-banyaknya atas partisipasi dan dukungan dari banyak pihak yang telah membantu baik berupa saran, kritikan, bantuan moril, dan lain-lain.

Semoga penelitian ini dapat bermanfaat baik bagi penulis, lingkungan Jurusan Teknik Informatika, mahasiswa, peneliti, dan lain-lain. Penulis juga berharap agar supaya penelitian ini dapat dilanjutkan baik untuk pengembangan keilmuan juga untuk dilanjutkan ke dalam implementasi perangkat lunak yang bisa digunakan.

DAFTAR ISI

	Halaman
LEMBAR JUDUL	i
LEMBAR PENGESAHAN	ii
ABSTRAK	iii
RINGKASAN	iv
PRAKATA	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
BAB I PENDAHULUAN	1
I.1. Latar Belakang	1
I.2. Rumusan Masalah	1
I.3. Tujuan Penelitian	2
I.4. Urgensi Penelitian	2
BAB II TINJAUAN PUSTAKA	3
II.1. Universal Timetabling Problem	3
II.2. Exhaustive Search	4
II.3. Distributed Processing	5
BAB III METODOLOGI	7
BAB IV HASIL DAN PEMBAHASAN	10
IV.1. Analisa Kelemahan, Kebutuhan, dan Kelayakan	10
IV.2. Perancangan	11
IV.2.1. Arsitektur Sistem & Perancangan Komunikasi	11
IV.2.2. Antarmuka	12
IV.2.3. Model Exhaustive Search	17
IV.2.3.1. Generate	18
IV.2.3.2. Test	22
IV.3. Implementasi	24
IV.3.1. Basis Data	24
IV.3.2. Aplikasi	25
IV.4. Testing	27
IV.4.1. Rencana Pengujian	27
IV.4.2. Kasus Uji	27
BAB V SIMPULAN, IMPLIKASI DAN SARAN	37
DAFTAR PUSTAKA	39
LAMPIRAN	

DAFTAR GAMBAR

Gambar 1. Solusi jadwal pada ruangan	3
Gambar 2. Depth First Search	4
Gambar 3. Pemrosesan Tanpa Distributed Processing	5
Gambar 4. Distributed Processing	5
Gambar 5. Distributed Processing menggunakan Local Area Network	6
Gambar 6. Bagan Alir Tahapan Penelitian	9
Gambar 7. Pengiriman Solusi Awal dan Rentang dari Server ke Semua Client ..	11
Gambar 8. Pembangkitan Solusi Lain Sejumlah Rentang oleh Client Kemudian Pengiriman Solusi Terbaik kepada Server	12
Gambar 9. Disain Antarmuka halaman Home	13
Gambar 10. Disain Antarmuka halaman Input Dosen	13
Gambar 11. Disain Antarmuka halaman Input Matakuliah	14
Gambar 12. Disain Antarmuka halaman Input hubungan Mengajar	15
Gambar 13. Disain Antarmuka halaman Input Ruang	16
Gambar 14. Disain Antarmuka halaman Input Jenis Ruang	16
Gambar 15. Disain Antarmuka halaman Input Prodi	17
Gambar 16. Disain Antarmuka halaman Input Parameter	17
Gambar 17. Pemetaan urutan daftar kelas ke slot ruang-hari-sks	18
Gambar 18. Disain Struktur Data urutan	20
Gambar 19. Struktur Database	24
Gambar 20. Struktur Paket Aplikasi	25
Gambar 21. Urutan yang akan dipetakan menjadi jadwal	30
Gambar 22. Jadwal yang merupakan pemetaan dari suatu urutan	31

Gambar 23. Nilai Total_tabrakan dari suatu urutan.....	32
Gambar 24. Konfirmasi total tabrakan mahasiswa pada jadwal.....	33
Gambar 25. Konfirmasi total tabrakan dosen pada jadwal	34
Gambar 26. Pengaruh jumlah komputer pada Durasi Pelaksanaan Tugas	37
Gambar 27. Pengaruh parameter sistem pada Durasi Pelaksanaan Tugas.....	38

DAFTAR TABEL

Tabel 1. Pembagian Team Teaching.....	3
Tabel 2. Daftar tabel dan fungsinya.....	24
Tabel 3. Daftar kelas pada paket org.jppf.application.template	25
Tabel 4. Daftar kelas pada paket Utama	26
Tabel 5. Daftar kelas pada paket Utilitas	26
Tabel 6. Rencana Pengujian.....	27
Tabel 7. Kasus uji kelas Permutator	28
Tabel 8. Kasus uji kelas MultiPermutator.....	29
Tabel 9. Kasus uji pemetaan urutan pada jadwal.....	30
Tabel 10. Kasus uji penghitungan nilai Total_tabrakan	32
Tabel 11. Kasus uji performansi dengan jumlah komputer tertentu.....	35
Tabel 12. Kasus uji pengaruh parameter sistem pada performansi	36

BAB I

PENDAHULUAN

1. Latar Belakang

Permasalahan penjadwalan adalah permasalahan yang setiap awal semester terjadi pada berbagai lingkungan perkuliahan termasuk pada lingkungan jurusan teknik informatika UNG. Permasalahan ini yang disebut *university timetabling problems* adalah permasalahan yang memerlukan waktu yang sangat lama untuk dipecahkan (Nugraha, 2008: 1). Hal ini dikarenakan dalam pemecahannya diperlukan untuk mempertimbangkan faktor-faktor seperti tabrakan jadwal dosen, tabrakan jadwal kelas dan juga tabrakan ruangan, belum lagi kalau memasukkan faktor seperti beban dosen perhari, beban kelas perhari, waktu interest dosen, dan lain-lain, dan semakin bertambah lagi dengan semakin banyaknya jumlah kelas, dan semakin sedikitnya dosen/ruangan.

Salah satu metode pencarian solusi yang sering digunakan untuk permasalahan ini adalah dengan menggunakan metode pencarian heuristic seperti simulated annealing (Solis, 2007), algoritma genetika (Solis, 2006) bahkan metode yang lebih baru seperti hyper-heuristic (Pillay). Exhaustive Search atau Generate and Test adalah metode pencarian heuristic paling sederhana dan menjamin ditemukannya solusi, tapi memerlukan waktu yang lama untuk ruang masalah yang luas (Suyanto, 2007: 23). Untuk mengatasi kekurangan dari metode ini maka metode ini perlu dikombinasikan dengan metode lain yang dapat mempercepat proses pencarian solusi.

Distributed processing adalah teknologi yang memungkinkan untuk memanfaatkan sumber daya pemroses untuk mempercepat proses penyelesaian suatu tugas dengan cara membagi-bagi tugas tersebut dan melaksanakan bagian-bagian tugas secara paralel. Salah satu penerapan dari metode ini adalah dengan arsitektur multikomputer, yaitu sekumpulan komputer yang saling terhubung untuk menyelesaikan suatu tugas. Distributed processing dengan arsitektur multikomputer sebenarnya dapat dibangun dengan memanfaatkan fasilitas komputer dan interkoneksi pada laboratorium komputer teknik informatika UNG.

Metode exhaustive search dengan distributed processing dipilih adalah karena beberapa alasan: (1) Pemecahan permasalahan penjadwalan memerlukan solusi yang terbaik, (2) distributed processing dapat diterapkan pada fasilitas laboratorium teknik informatika UNG.

2. Rumusan Masalah

Rumusan Permasalahan pada penelitian ini adalah :

1. Bagaimana menerapkan metode exhaustive search pada permasalahan UTP
2. Bagaimana menggunakan Distributed Processing untuk mempercepat proses pencarian permasalahan menggunakan exhaustive search.

3. Tujuan Penelitian

Tujuan penelitian ini adalah untuk meneliti penerapan metode exhaustive search dan distributed processing untuk menyelesaikan permasalahan penjadwalan pada jurusan teknik informatika UNG.

4. Urgensi Penelitian

Penelitian ini sangat diperlukan sebagai dasar dalam pengembangan perangkat lunak pemecahan permasalahan penjadwalan yang terjadi pada jurusan teknik informatika UNG.

BAB II

TINJAUAN PUSTAKA

1. Universal Timetabling Problem

Permasalahan penjadwalan adalah permasalahan yang selalu terjadi pada setiap semester pada jurusan teknik informatika UNG. Permasalahan ini menyangkut pembuatan suatu jadwal, misalkan terdapat sejumlah m matakuliah Matakuliah $M = \{M_1, M_2 \dots M_m\}$, sejumlah n kelas $K = \{K_1, K_2 \dots K_n\}$, sejumlah o Ruang $R = \{R_1, R_2 \dots R_o\}$. Dari matakuliah dan kelas terbentuk MatakuliahKelas $MK = \{M_1K_1, M_1K_2, M_2K_1, M_2K_2, M_3K_1, M_3K_2, \dots M_mK_n\}$. Apabila jumlah perkuliahan adalah 6 (enam) $H = \{H_a, H_b, H_c, H_d, H_e, H_f\}$, dan setiap hari terdapat 4 (empat) waktu perkuliahan $W = \{W_1, W_2, W_3, W_4\}$, setiap minggu maka terbentuk TimeSlot = $\{R_1H_aW_1, R_1H_aW_2, R_1H_aW_3, \dots R_2H_fW_4\}$.

Senin	Selasa	Rabu	Kamis	Jumat	Sabtu
M1K1			M3K1		
M2K2					

Ruangan R1

Senin	Selasa	Rabu	Kamis	Jumat	Sabtu
M2K1			M3K2		

Ruangan R2

Gambar 1. Solusi Jadwal Pada Ruangan

Setiap matakuliah diampu oleh team teaching yang dapat terdiri atas dua atau lebih dosen, misalkan seperti pada tabel dibawah :

Matakuliah	Team Teaching
M1	DosenV, DosenW
M2	DosenW, DosenX
M3	DosenY, DosenZ
...	
Mm	

Tabel 1. Pembagian Team Teaching

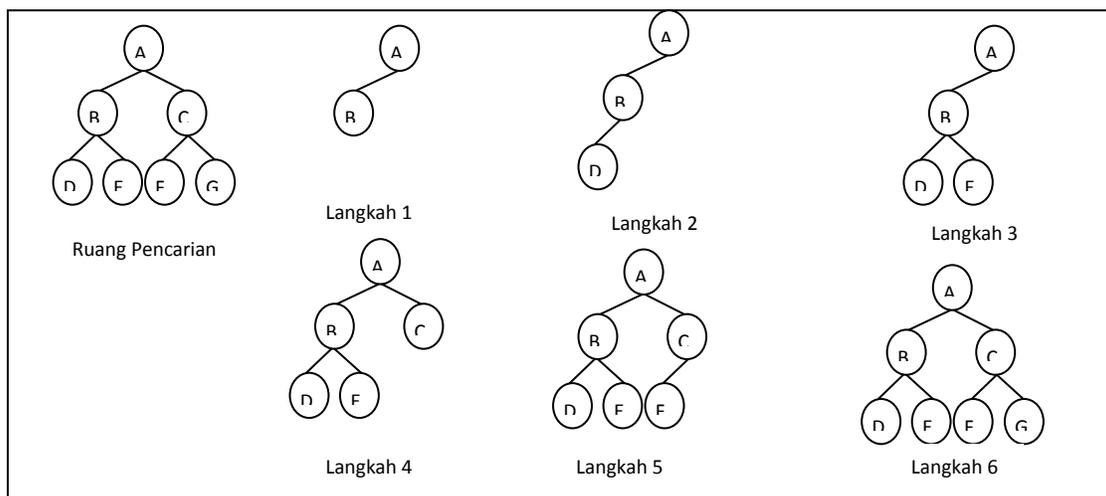
Permasalahan timbul karena terdapatnya beberapa konstrain yang harus dipenuhi untuk mendapatkan jadwal yang terbaik:

- Tidak boleh terdapat timeslot (ruangan dan waktu) yang dipakai oleh MatakuliahKelas yang berbeda, seperti pada kasus M1K1 dengan M2K2 yang keduanya menggunakan TimeSlot R1HaW1 (Ruang R1, hari senin, dan waktu perkuliahan W1).
- Meminimalkan Matakuliah sama diajarkan pada waktu yang sama, walaupun pada ruangan yang berbeda, seperti pada kasus M3K1 dengan M3K2 pada tabel 1.
- Tidak boleh terdapat Kelas yang sama mendapatkan matakuliah yang berbeda diajarkan pada waktu yang sama, walaupun pada ruangan yang berbeda, seperti pada kasus M1K1 dan M2K1 pada tabel 1.

2. Exhaustive Search

Exhaust search atau metode generate and test adalah metode pencarian yang akan membangkitkan semua kemungkinan solusi yang mungkin dari suatu permasalahan, kemudian akan menguji nilai dari masing-masing solusi, untuk menentukan solusi terbaik.

Dalam membangkitkan semua solusi, metode ini menggunakan prosedur Depth First Search (Suyanto, 2007: 23). Prosedur pencarian ini akan memprioritaskan pada kedalaman eksplorasi daripada pada keluasan eksplorasi.



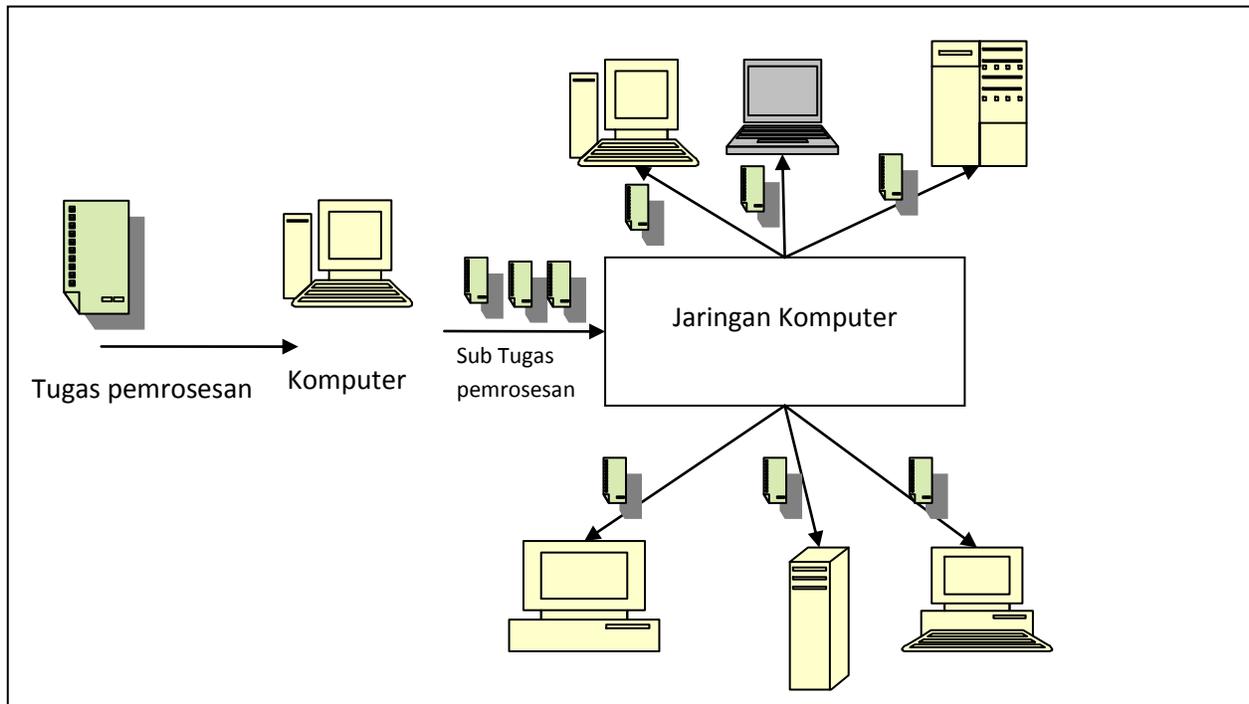
Gambar 2. Depth First Search (DFS)

3. Distributed Processing

Sebelum adanya teknologi ini, tugas pemrosesan hanya dikerjakan secara mandiri oleh processor yang menjalankan aplikasi yang bersangkutan (gambar 3). Hal ini membuat kecepatan pemrosesan hanya tergantung pada kecepatan pemrosesan processor dari komputer yang digunakan. Dengan adanya teknologi jaringan komputer maka muncul pemikiran untuk memaksimalkan sumber daya pemrosesan berupa komputer-komputer yang saling terhubung tersebut untuk secara bersama-sama melaksanakan suatu tugas (gambar 3).

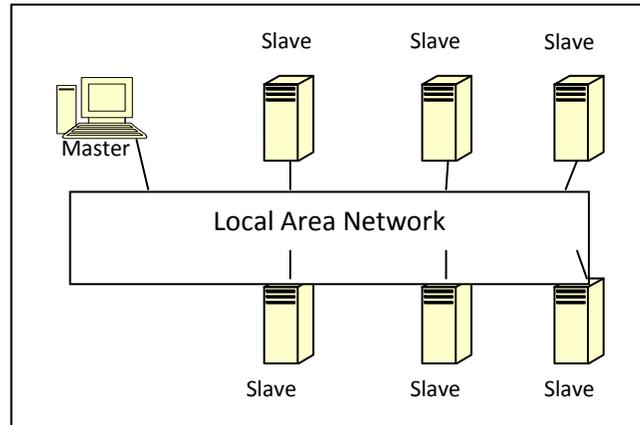


Gambar 3. Pemrosesan Tanpa Distributed Processing



Gambar 4. Distributed Processing

Distributed processing adalah metode untuk mempercepat proses pengekseskuan suatu tugas pemrosesan dengan cara membagi-bagi tugas menjadi sub-sub tugas dan melaksanakan sub-sub tugas tersebut secara paralel. Arsitektur sistem DP (Distributed Processing) dapat berupa satu komputer yang memiliki lebih dari satu prosesor, atau dapat pula berupa beberapa komputer yang saling terhubung misalkan melalui suatu jaringan Local Area Network (LAN).



Gambar 5. Distributed Processing menggunakan Local Area Network

BAB III

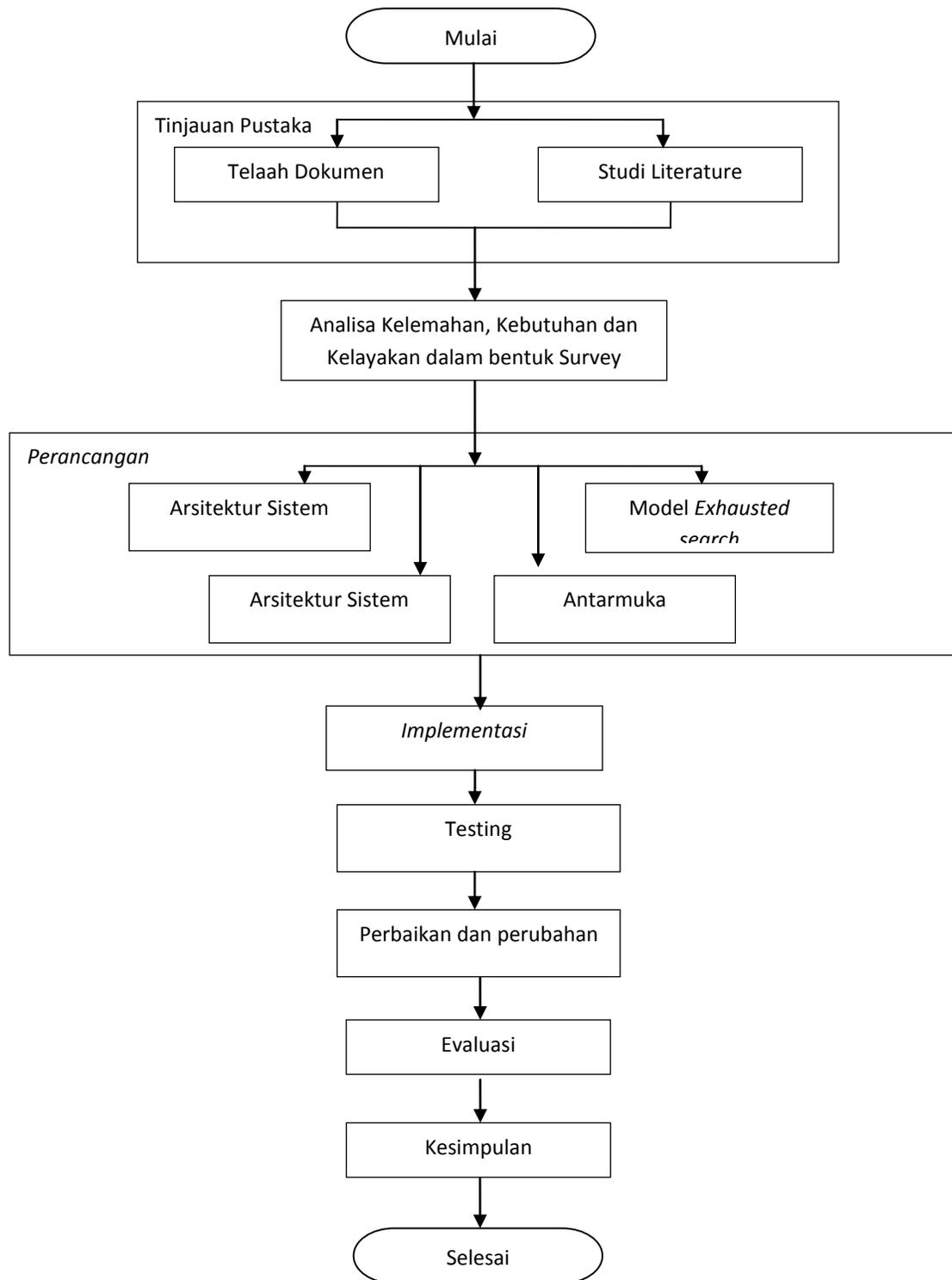
METODOLOGI

Penelitian ini dilakukan dalam beberapa langkah, yaitu tinjauan pustaka, analisa kelemahan, kebutuhan dan kelayakan sistem, perancangan sistem, pembuatan prototipe sistem, pengujian prototipe sistem, implementasi dan sosialisasi serta evaluasi. Penjabaran dari langkah-langkah penelitian tersebut yaitu :

1. **Tinjauan pustaka**, untuk menemukan teori-teori dan penelitian sebelumnya mengenai universal timetabling problem (UTP), exhaustive search dan distributed processing. Keluaran dari tahapan ini adalah teridentifikasi permasalahan dalam pemecahan permasalahan UTP dan keefektifan metode-metode yang telah diujicobakan untuk memecahkan permasalahan tersebut.
2. **Analisa**, untuk melakukan analisis kelemahan pada sistem yang telah ada, analisis kebutuhan dan kelayakan teknologi untuk pengembangan sistem dalam bentuk survey. Pada tahap ini, akan dibahas mengenai perlunya pengembangan suatu perangkat lunak untuk mempermudah dalam pemecahan permasalahan penjadwalan. Keluaran dari tahapan ini adalah daftar kebutuhan dan kelayakan dalam pengembangan perangkat lunak penjadwalan. Target atau indikator keberhasilan pada tahap ini adalah teridentifikasinya kebutuhan dan kelayakan terhadap pengembangan perangkat lunak penjadwalan.
3. **Perancangan sistem**, untuk merancang model pemecahan permasalahan penjadwalan menggunakan metode exhaustive search. Keluaran dari tahapan ini adalah model pemecahan menggunakan exhaustive search, rancangan arsitektur perangkat lunak, rancangan antarmuka perangkat lunak, rancangan prosedural perangkat lunak.
4. **Implementasi prototype**, untuk pengimplementasian rancangan sistem yang telah dibuat kedalam bentuk suatu program prototype yang bisa dijalankan pada lingkungan pengembangan. Keluaran dari tahapan ini adalah terdapatnya suatu perangkat lunak yang bisa dijalankan pada lingkungan pengembangan dan dapat mensimulasikan fungsi-fungsi dari sistem.
5. **Pengujian sistem**, untuk menguji program prototype yang telah dibuat pada tahap implementasi. Keluaran dari tahapan ini adalah teridentifikasinya kesalahan-kesalahan yang menjadi dasar untuk melakukan perbaikan dan perubahan.

6. **Perbaikan dan perubahan sistem**, pada tahapan ini dilakukan perbaikan dan atau perubahan, baik pada program prototype juga pada rancangan-rancangan sistem yang telah dibuat sebelumnya. Keluaran dari tahapan ini adalah program prototype yang siap diinstall pada lingkungan pengguna.

Penjabaran langkah-langkah penelitian dapat disederhanakan dalam bentuk bagan alir tahapan penelitian berikut :



Gambar 6. Bagan Alir Tahapan Penelitian

BAB IV

HASIL DAN PEMBAHASAN

A. Analisa Kelemahan, Kebutuhan, dan Kelayakan

1. Analisa Kelemahan Sistem

Dari hasil pengamatan dalam proses pembuatan jadwal menggunakan aplikasi dalam lingkungan teknik informatika UNG, didapatkan beberapa masalah yang selalu muncul:

- a. Sistem yang digunakan hanya menyediakan fasilitas pendeteksian tabrakan, tetapi belum mengimplementasikan pencarian solusi secara otomatis, sehingga pembuatan dan perbaikan jadwal harus dilakukan secara manual.
- b. Sistem yang ada belum dapat menemukan solusi yang optimal, dan menghasilkan solusi yang memiliki beberapa tabrakan antar kelas dan dosen.
- c. Sistem tidak memanfaatkan sumber daya pemrosesan berupa komputer yang banyak tersedia pada lingkungan teknik informatika UNG untuk melakukan komputasi pencarian solusi.

2. Analisa Kebutuhan sistem

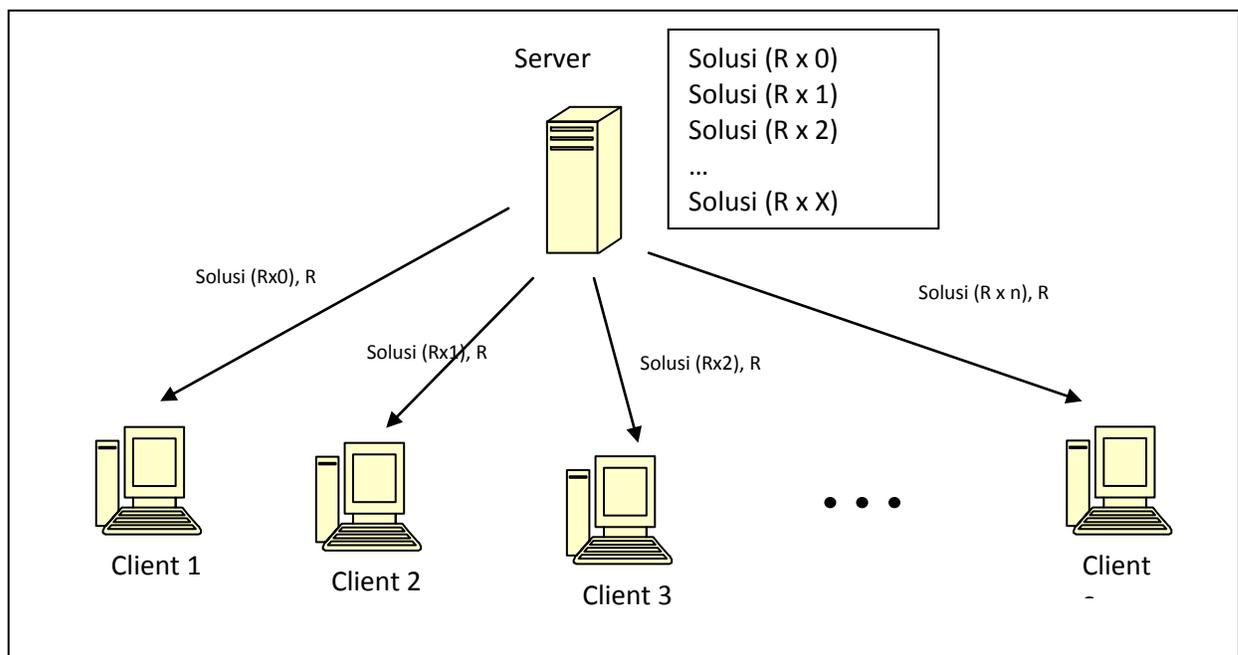
- a. Dibutuhkan suatu sistem yang dapat mencari solusi secara otomatis dan juga dapat mendeteksi tabrakan baik antar kelas dan dosen.
- b. Dibutuhkan suatu sistem yang dapat menggunakan metode pencarian yang dapat menjamin ditemukannya solusi yang optimal.
- c. Dibutuhkan sistem yang dapat menggunakan sumber daya pemrosesan dalam melakukan proses pencarian solusi.

B. Perancangan

1. Arsitektur Sistem & Perancangan Komunikasi

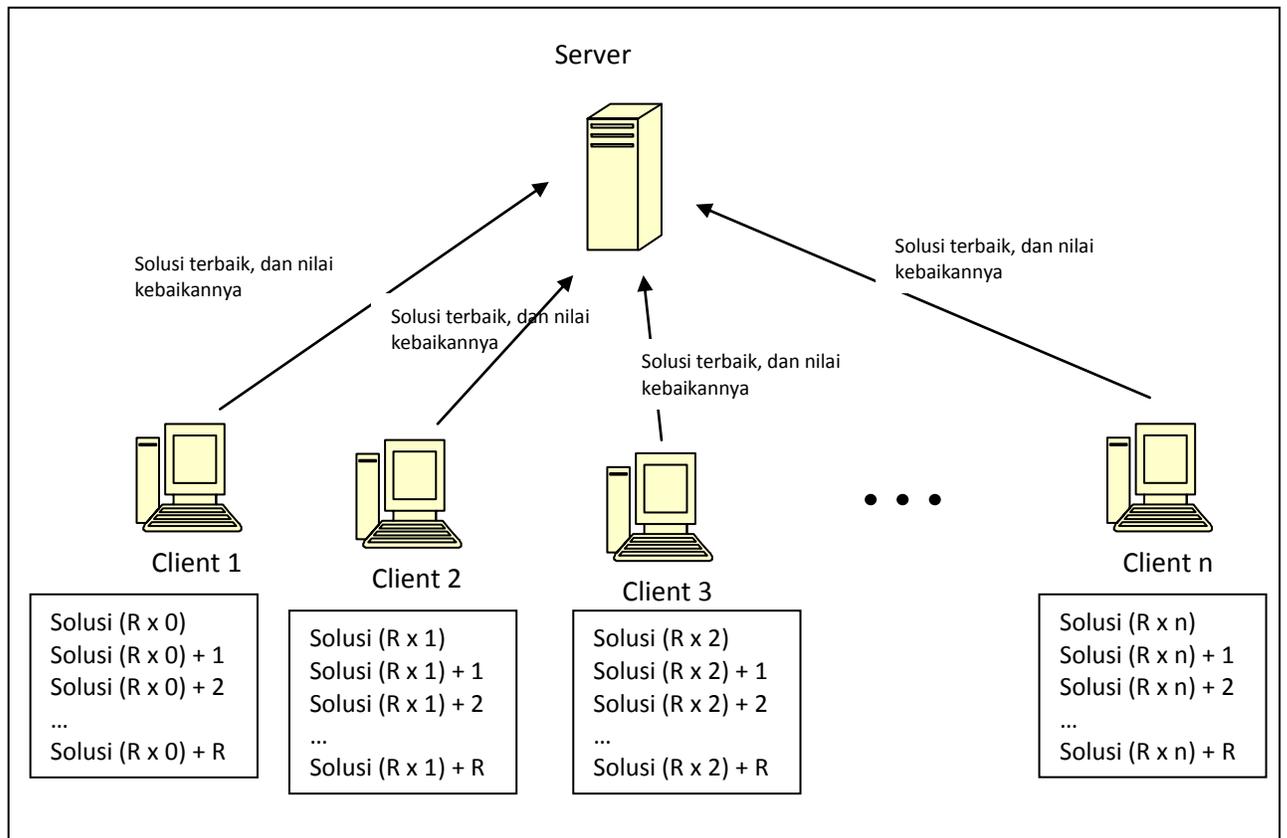
Arsitektur sistem yang digunakan adalah berbentuk client-server. Hal yang perlu diperhatikan dalam perancangan komunikasi antara client dan server adalah diperlukannya untuk meminimalisir jumlah pengiriman data antara client dengan server, dan memaksimalkan pemrosesan mandiri pada setiap client. Hal ini membuat model komunikasi yang dipilih adalah seperti dibawah:

1. Server membangkitkan sejumlah solusi awal dalam suatu rentang (R) tertentu. Kemudian server mengirimkan solusi awal beserta nilai rentang kepada client.



Gambar. 7. Pengiriman solusi awal dan rentang dari Server ke semua client

2. Setelah menerima solusi awal, client akan membangkitkan solusi-solusi lain sejumlah nilai rentang, melakukan penilaian terhadap semua solusi tersebut, kemudian mengirimkan solusi terbaik kepada server.

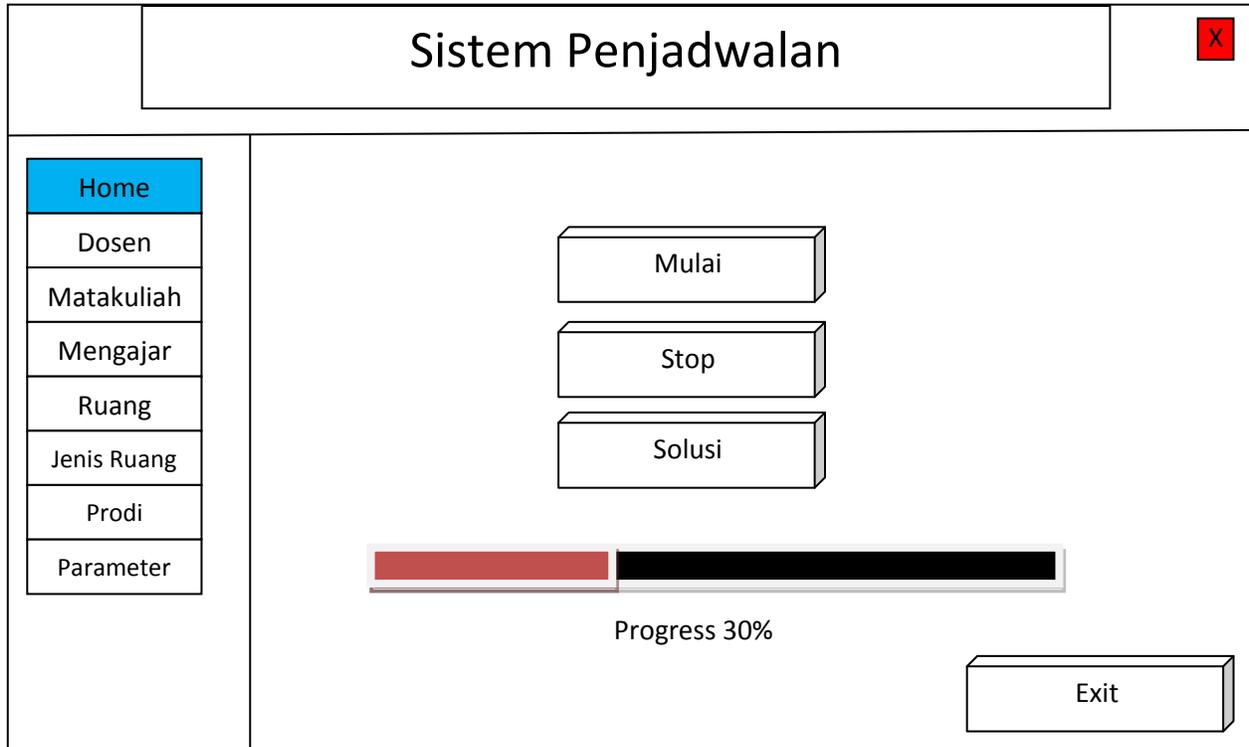


Gambar. 8. Pembangkitan Solusi Lain Sejumlah Rentang oleh Client Kemudian Pengiriman Solusi Terbaik kepada Server

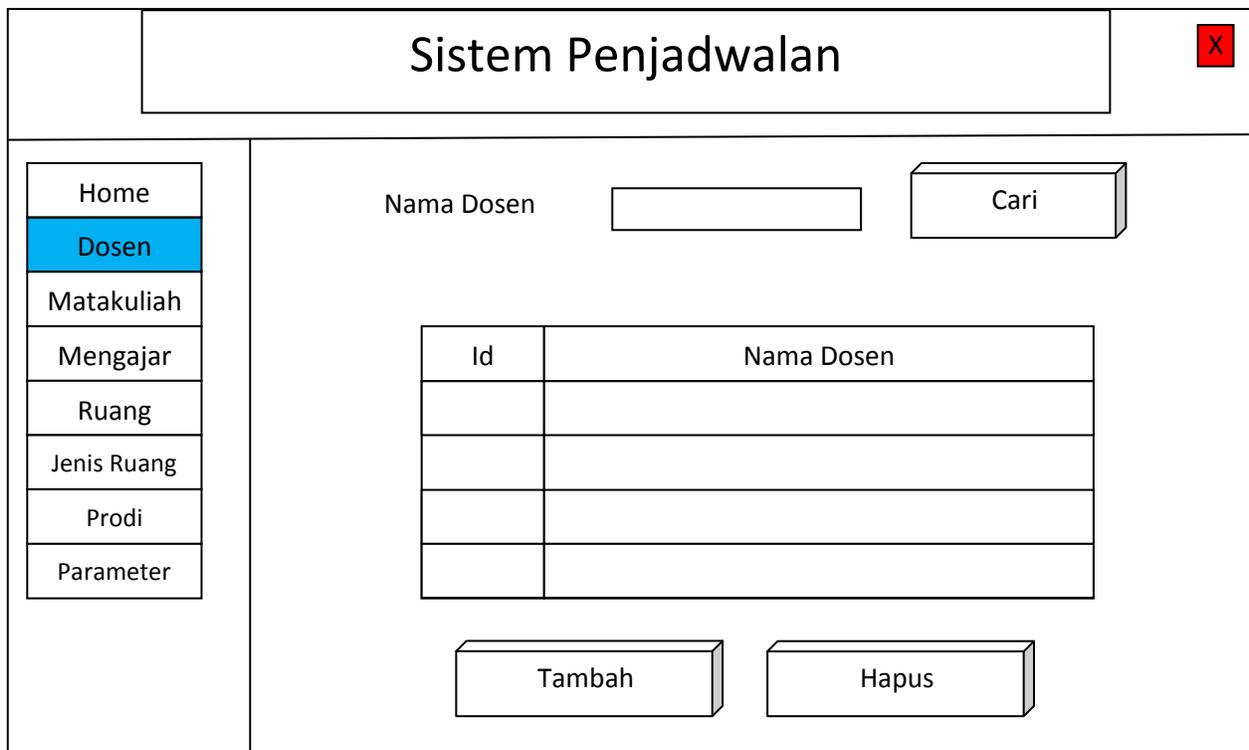
Pendekatan model komunikasi diatas mengurangi jumlah data yang harus dikirim antara client dan server, sehingga mengurangi kemungkinan terjadinya bottleneck karena cost untuk pengiriman data.

2. Antarmuka

Disain antarmuka dari sistem perangkat lunak ini digambarkan pada gambar berikut.



Gambar 9. Disain Antarmuka Halaman Home



Gambar 10. Disain Antarmuka halaman Input Dosen

Sistem Penjadwalan

X

Home

Dosen

Matakuliah

Mengajar

Ruang

Jenis Ruang

Prodi

Parameter

Nama Matakuliah

Prodi ▾

Jenis Ruangan ▾

SKS

Semester

Jumlah Kelas

Id	Nama Matakuliah	Jenis Ruangan	SKS	Semester	Jml Kelas

Gambar 11. Disain Antarmuka halaman Input Matakuliah

Sistem Penjadwalan

X

Home

Dosen

Matakuliah

Mengajar

Ruang

Jenis Ruang

Prodi

Parameter

Nama Matakuliah

Nama Dosen

Matakuliah

Id	Nama Matakuliah

Daftar Pengajar

Id	Nama Dosen

Tambah
Sebagai
Pengajar
←

Keluar
n dari
Pengajar
X

Id	Nama Dosen

Gambar 12. Disain Antarmuka halaman Input hubungan Mengajar

Sistem Penjadwalan X

Home
 Dosen
 Matakuliah
 Mengajar
Ruang
 Jenis Ruang
 Prodi
 Parameter

Nama Ruang

Jenis Ruang

Kapasitas

Id	Nama Ruang	Jenis Ruang	Kapasitas

Gambar 13. Disain Antarmuka halaman Input Ruang

Sistem Penjadwalan X

Home
 Dosen
 Matakuliah
 Mengajar
 Ruang
Jenis Ruang
 Prodi
 Parameter

Nama Jenis Ruang

Id	Nama Jenis Ruang

Gambar 14. Disain Antarmuka halaman Input Jenis Ruang

Sistem Penjadwalan		X						
<ul style="list-style-type: none"> Home Dosen Matakuliah Mengajar Ruang Jenis Ruang <li style="background-color: #00aaff; color: white;">Prodi Parameter 	<p>Nama Prodi <input type="text"/> <input type="button" value="Cari"/></p> <p><input type="button" value="Tambah"/> <input type="button" value="Hapus"/></p> <table border="1"> <thead> <tr> <th>Id</th> <th>Nama Prodi</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> </tr> </tbody> </table>		Id	Nama Prodi				
Id	Nama Prodi							

Gambar 15. Disain Antarmuka halaman Input Prodi

Sistem Penjadwalan		X									
<ul style="list-style-type: none"> Home Dosen Matakuliah Mengajar Ruang Jenis Ruang Prodi <li style="background-color: #00aaff; color: white;">Parameter 	<p>Nama Parameter <input type="text"/> <input type="button" value="Cari"/></p> <p>Nilai Parameter <input type="text"/></p> <p><input type="button" value="Tambah"/> <input type="button" value="Ubah"/> <input type="button" value="Hapus"/></p> <table border="1"> <thead> <tr> <th>Id</th> <th>Nama Parameter</th> <th>Nilai Parameter</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>		Id	Nama Parameter	Nilai Parameter						
Id	Nama Parameter	Nilai Parameter									

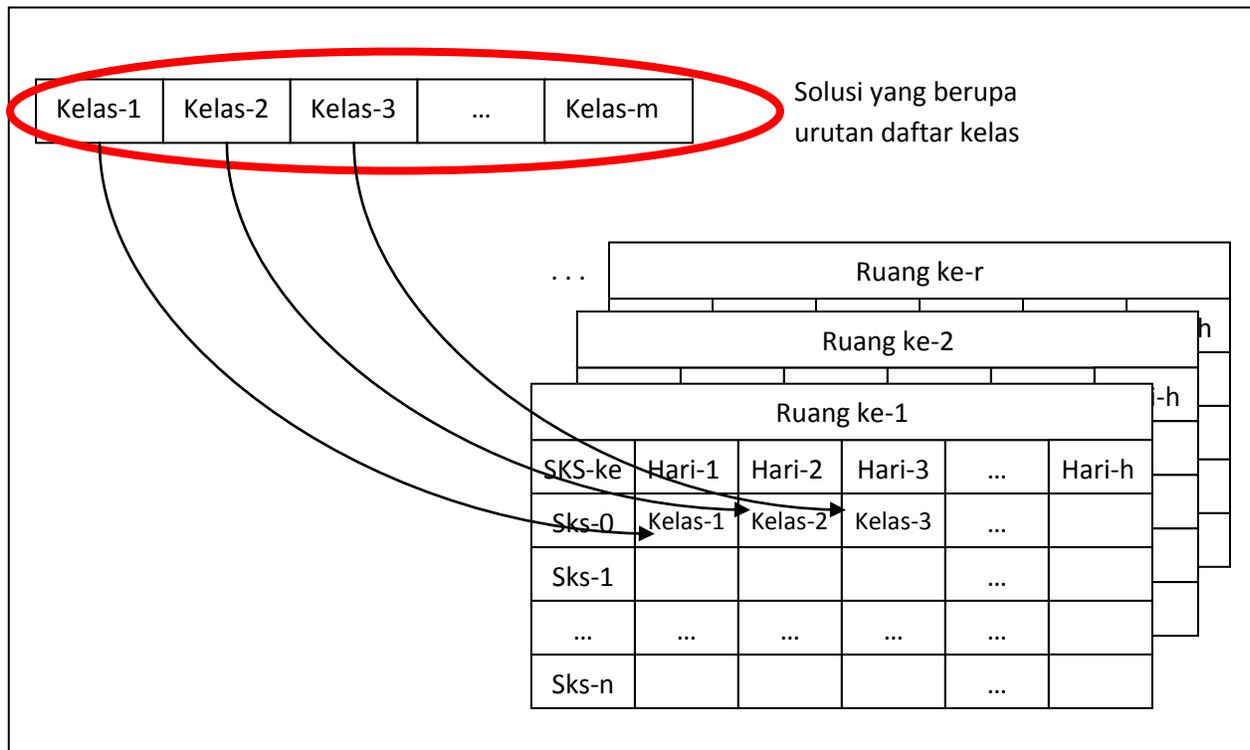
Gambar 16. Disain Antarmuka halaman Input Parameter

3. Model Exhaustive Search

Metode pencarian Exhaustive Search terdiri dari dua langkah yaitu:

1. Generate

Langkah generate adalah proses pembangkitan semua kemungkinan solusi dari permasalahan. Untuk permasalahan penjadwalan, maka solusi merupakan suatu **urutan daftar kelas** yang bisa dipetakan pada slot ruang-hari-sks sehingga menjadi suatu jadwal.



Gambar 17. Pemetaan urutan daftar kelas ke slot ruang-hari-sks

Model solusi yang akan digunakan adalah menggunakan suatu urutan dari kelas-kelas perkuliahan.

Untuk sejumlah K kelas dibuat urutan kelas seperti dibawah:

Kelas 1	Kelas 2	Kelas 3	Kelas 4	...	Kelas K
0	1	2	3	...	K-1

Atau direpresentasikan hanya menggunakan indexnya.

0	1	2	...	K-3	K-2	K-1
---	---	---	-----	-----	-----	-----

Dari solusi awal ini akan dibangkitkan solusi-solusi lainnya. Metode pembangkitan adalah mengikuti model pembangkitan sistem bilangan dengan aturan permutasi, yaitu suatu bilangan tidak boleh muncul lebih dari satu kali. Misalkan sistem bilangan octal 0-7 yang dikenai dengan aturan permutasi maka nilai awalnya adalah

0 1 2 3 4 5 6 7

Nilai berikutnya adalah

0 1 2 3 4 5 7 6

Dan seterusnya:

0 1 2 3 4 5 7 6

0 1 2 3 4 6 5 7

0 1 2 3 4 6 7 5

0 1 2 3 4 7 5 6

...

Sampai pada nilai tertinggi yaitu:

7 6 5 4 3 2 1

Sehingga algoritma umumnya adalah seperti berikut:

1. Buat urutan bilangan yang mewakili posisi kelas perkuliahan pada slot ruang-hari-sks pada jadwal
2. Bangkitkan urutan bilangan selanjutnya menggunakan sifat sistem bilangan dengan rentang nilai 0 – (kelas-1) dengan aturan permutasi

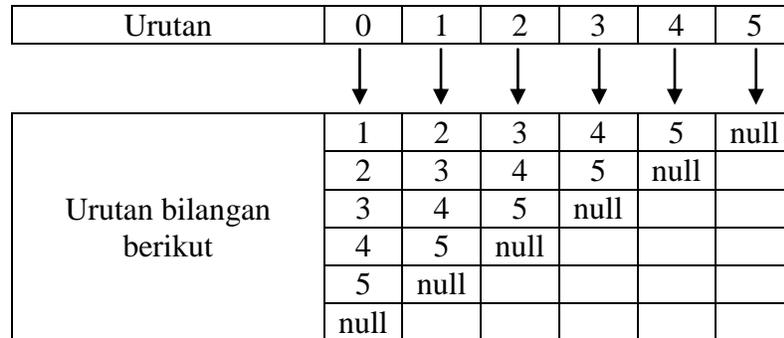
0	1	2	3	4	...	K-1
---	---	---	---	---	-----	-----

0	1	2	<u>4</u>	<u>3</u>	...	K-1
---	---	---	----------	----------	-----	-----

0	1	<u>3</u>	<u>2</u>	<u>4</u>	...	K-1
---	---	----------	----------	----------	-----	-----

0	1	3	<u>4</u>	<u>2</u>	...	K-1
---	---	---	----------	----------	-----	-----

Untuk mengimplementasikan pembangkitan solusi-solusi tersebut, akan digunakan struktur data urutan seperti dibawah:



Gambar 18. Disain Struktur Data urutan

Kemudian pada urutan kelas tersebut dilakukan permutasi untuk menentukan/menggenerate urutan-urutan selanjutnya dengan algoritma dibawah:

a. Algoritma inisialisasi :

```

urutan[]: adalah daftar nilai dari urutan sekarang
K = jumlah bilangan dalam urutan

//Buat urutan bilangan selanjutnya dari seluruh bilangan yang ada
1. Dari i = 0 sampai i = (K-1) lakukan
  a. urutan[i] = i
  b. bilanganBerikut = null;
  c. dari j = (K-1) sampai j = (i+1)
    i. jika j == (K - 1)
      1) bilanganBerikut = j
      2) bilanganBerikut.berikut = null;
    ii. jika j > i
      1) bilanganSebelum = j
      2) bilanganSebelum.berikut = bilanganBerikut
      3) bilanganBerikut = bilanganSebelum
  d. urutan[i].berikut = bilanganBerikut

```

b. Algoritma untuk pembangkitan urutan kelas selanjutnya :

```

urutan[]: adalah daftar nilai dari urutan sekarang
K = jumlah bilangan dalam urutan

//Maju satu langkah untuk menghasilkan urutan baru

```

1. Dari $i = (K-1)$ sampai $i = 0$ lakukan
 - a. Jika $urutan[i] == null$
 - i. lanjutkan ke i berikutnya
 - b. jika $urutan[i] \neq null$
 - i. buat urutan bilangan yang terdapat pada $urutan[i]$ sampai ke $urutan[K-1]$ kecuali bilangan yang sama dengan $urutan[i].berikut$ (simpan pada $urutanBantu$)
 - ii. $urutan[i] = urutan[i].berikut$
 - iii. urutkan $urutanBantu$ secara ascending
 - iv. $counter = 0$;
 - v. dari $j = i + 1$ sampai $j = (K-1)$
 - 1) $urutan[j] = urutanBantu[counter]$
 - 2) $counter = counter + 1$
 - vi. dari $k = i + 1$ sampai $k = (K - 2)$
 - 1) $bilanganBerikut = urutan[K - 1]$
 - 2) dari $m = (K - 1)$ sampai $m = (K+1)$
 - a) jika $m = (K - 1)$
 - i). $bilanganBerikut.berikut = null$
 - b) jika $m < (K - 1)$
 - i). $bilanganSebelum = urutan[m]$
 - ii). $bilanganSebelum.berikut = berikut$
 - iii). $bilanganBerikut = bilanganSebelum$
 - 3) $urutan[k].berikut = bilanganBerikut$
 - vii. keluar dari perulangan b.

c. Algoritma untuk memetakan urutan pada jadwal

JHK : Jumlah hari perkuliahan

JRuang : Jumlah ruang

SKSPERHARI : Jumlah sks per-hari

JKelas : Jumlah total kelas

daftarKelas[] : daftar kelas yang akan dimasukkan pada jadwal

urutan[]: adalah daftar nilai dari urutan sekarang

1. $jadwal[JHK][JRuang][SKSPERHARI]$ //buat jadwal yang merupakan array 3 dimensi
2. $indexKelas = 0$
3. dari $i = 0$ sampai $i = (JHK - 1)$ selama $indexKelas < JKelas$
 - a. dari $j = 0$ sampai $j = (JRuang - 1)$ selama $indexKelas < JKelas$
 - i. dari $k = 0$ sampai $k = (SKSPERHARI - 1)$ selama $indexKelas < JKelas$
 - 1). $sks = daftarKelas[urutan[indexKelas]].sks$
 - 2). Jika $k + (sks - 1) < SKSPERHARI$
 - a). Dari $l = k$ sampai $l = k + (sks - 1)$
 - i). $jadwal[i][j][l] = daftarKelas[urutan[indexKelas]].id$
 - b). $k = k + (sks - 1)$
 - c). $indexKelas = indexKelas + 1$

- 3). Jika $k + (sks - 1) \geq SKSPERHARI$
 a). Keluar dari perulangan i.

2. Test

Langkah test adalah proses penghitungan total_tabrakan dari setiap solusi yang dibangkitkan. Total_tabrakan dihitung :

$$\text{Total_Tabrakan} = (\text{jumlah_tabrakan_kelas_mahasiswa} + \text{jumlah_tabrakan_dosen} + \text{jumlah_kelas_tidak_masuk_jadwal})$$

Keoptimalan suatu solusi dinilai berdasarkan nilai total_tabrakan yang dimiliki solusi tersebut, semakin kecil total_tabrakan maka semakin optimal solusi tersebut.

- a. Algoritma menghitung jumlah tabrakan_kelas_mahasiswa, jumlah_tabrakan_dosen dan jumlah kelas_tidak_masuk_jadwal

JHK : Jumlah hari perkuliahan
 JRuang : Jumlah ruang
 SKSPERHARI : Jumlah sks per-hari
 JKelas : Jumlah total kelas
 daftarKelas[] : daftar kelas yang akan dimasukkan pada jadwal
 urutan[]: adalah daftar nilai dari urutan sekarang
 DKJN[]: adalah daftar kode jenis ruang
 DR[]: adalah daftar ruang
 JADWAL[]: adalah daftar jadwal
 daftarMengajar[]: daftar dosen yang mengajar pada matakuliah

tabrakanMhs = 0;
 tabrakanDsn = 0;

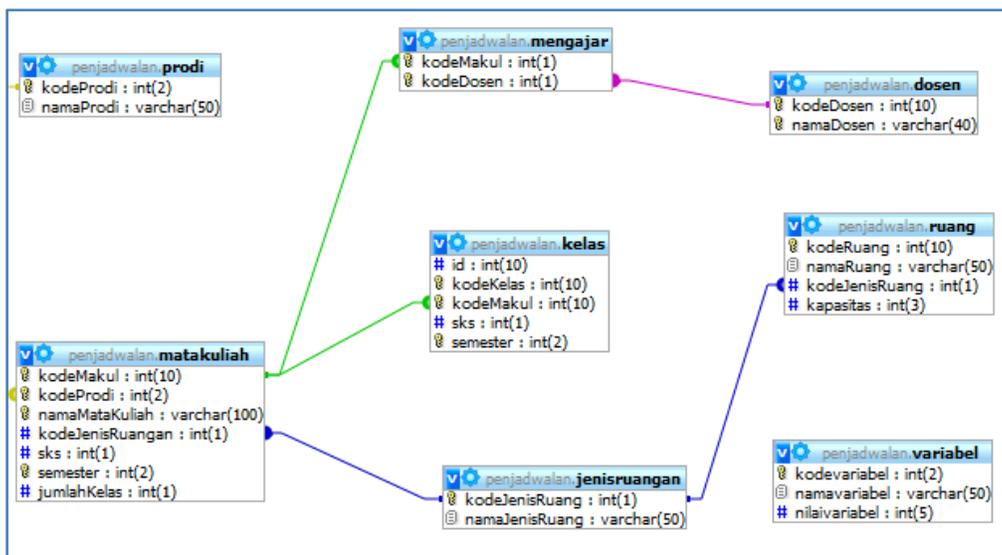
1. Dari $i = 0$ sampai $i = JHK - 1$
 - a. cekKelas = new hashtable<int, hashtable<int,byte>>();
 - b. cekDosen = new hashtable<int, byte>();
 - c. dari $j = 0$ sampai $j = SKSPERHARI - 1$
 - i. dari $k = 0$ sampai $k = DKJN.length - 1$
 - 1). kodeJenisRuang = DKJN[k]
 - 2). ruangRuang = DR[kodeJenisRuang];
 - 3). Dari $L = 0$ sampai $L = ruangRuang.length - 1$
 - a) idKelas = jadwal[ruangRuang[L].getKodeRuang()][i][j]
 - b) idKelas = idKelas - 1
 - c) jika idKelas < -1 //kalau jadwal kosong atau shalat Jumat
 - i). lanjut ke perulangan 3).
 - d) kodeKelas = daftarKelas[kodeJenisRuang][idKelas].getKodeKelas()
 - e) jika kodeKelas < 0 //kalau kelas kosong

- i). lanjut ke perulangan 3).
- f) Semester = daftarKelas[kodeJenisRuang][idKelas].getSemester()
- g) Jika cekKelas.get(semester) == null
 - i). cekDaftarKodeKelas = new hashtable<int,byte>()
 - ii). cekDaftarKodeKelas.put(kodeKelas,1)
 - iii). cekKelas.put(semester, cekDaftarKodeKelas)
- h) jika cekKelas.get(semester).get(kodeKelas) == null
 - i). cekDaftarKodeKelas = cekKelas.get(semester)
 - ii). cekDaftarKodeKelas.put(kodeKelas, 1)
 - iii). cekKelas.put(semester,cekDaftarKodeKelas)
- i) jika cekKelas.get(semester).get(kodeKelas) == 1
 - i). tabrakanMhs = tabrakanMhs + 1
- j) kodeMakul = daftarKelas[kodeJenisRuang][idKelas].getKodeMakul()
- k) dari m = 0 sampai m = daftarMengajar.length
 - i). kodeDosen = daftarMengajar[m].getKodeDosen()
 - ii). Jika daftarMengajar[m].getKodeMakul() == kodeMakul
 - (1).Jika cekDosen.get(kodeDosen) <> null
 - (a). tabrakanDsn = tabrakanDsn + 1
 - (2).Jika tidak
 - (a). cekDosen.put(daftarMengajar[m].getKodeDosen(), 1)
- ii. cekKelas.clear()
- iii. cekDosen.clear()

C. IMPLEMENTASI

1. Basis Data

Sistem ini diimplementasikan dalam dengan menggunakan aplikasi basis data MySQL versi 5.0.8 , struktur database yang digunakan adalah seperti dibawah:



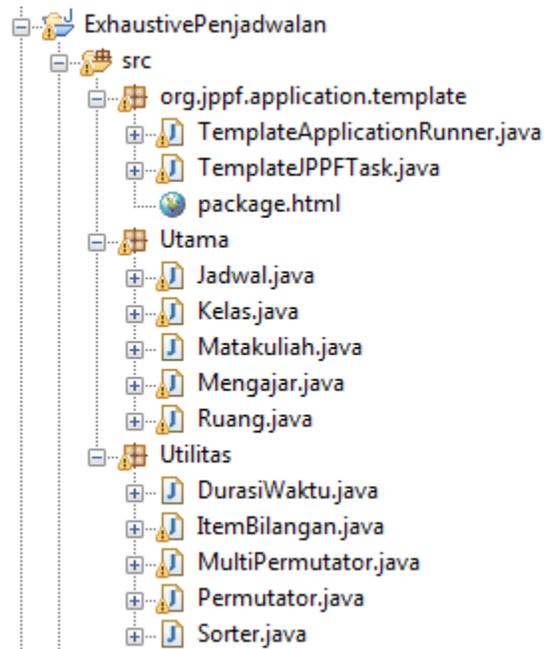
Gambar 19. Struktur Database

No	Nama Tabel	Fungsi
1	Jenisruangan	Menyimpan data mengenai kategori ruangan, misalnya ruang lab, ruang kuliah, dll.
2	Ruang	Menyimpan daftar ruangan yang bisa digunakan
3	Matakuliah	Menyimpan daftar matakuliah
4	Dosen	Menyimpan daftar dosen yang bisa mengajar
5	Prodi	Meyimpan daftar program studi
6	Mengajar	Menyimpan pemetaan dari dosen ke matakuliah yang diajarnya
7	Variabel	Menyimpan parameter-parameter dalam permasalahan, misalnya: parameter JUMLAHHARIPERKULIAHAN, SKSPERHARI, dll.
8	Kelas	Menyimpan kelas yang dibangkitkan berdasarkan data daftar matakuliah.
9	testJadwal	Menyimpan data jadwal yang digunakan untuk pengujian sistem.

Tabel 2. Daftar tabel dan fungsinya

2. Aplikasi

Sistem menggunakan framework JPPF yang merupakan framework pengembangan aplikasi untuk komputasi terdistribusi. Framework JPPF ini berbasis bahasa pemrograman Java, tools pengembangan yang digunakan adalah Eclipse. Struktur package aplikasi adalah seperti dibawah.



Gambar 20. Struktur Paket Aplikasi

Daftar kelas yang digunakan adalah:

a. Paket org.jppf.application.template

Nama kelas	Fungsi
TemplateApplicationRunner.java	Merupakan kelas yang mengatur keseluruhan pemrosesan : <ul style="list-style-type: none"> • Pengambilan data dari database • Pengiriman data ke client • Penanggulangan response dari client
TemplateJPPFTask.java	Merupakan kelas yang merupakan pendefinisian pekerjaan yang akan dilakukan oleh client <ul style="list-style-type: none"> • Memetakan urutan ke jadwal • Menghitung tabrakan antar mahasiswa • Menghitung tabrakan antar dosen • Menghitung kelas yang tidak masuk jadwal

Tabel 3. Daftar kelas pada paket org.jppf.application.template

b. Paket Utama

Nama kelas	Fungsi
Jadwal.java	Kelas yang menyimpan data jadwal berupa pemetaan kelas ke hari-ruang-sks
Kelas.java	Representasi data dari tabel kelas
Matakuliah.java	Representasi data dari tabel matakuliah
Mengajar.java	Representasi data dari tabel mengajar
Ruang.java	Representasi data dari tabel ruang

Tabel 4. Daftar kelas pada paket Utama

c. Paket Utilitas

Nama kelas	Fungsi
Durasiwaktu.java	Kelas untuk memformat waktu milidetik ke dalam bentuk jam:menit:detik
ItemBilangan.java	Representasi dari setiap bilangan dalam suatu urutan bilangan
Permutator.java	Kelas yang menyimpan urutan bilangan: <ul style="list-style-type: none"> • Membuat urutan baru • Membangkitkan (satu atau n) urutan selanjutnya
MultiPermutator.java	Kelas yang menggabungkan beberapa permutator dan menyediakan fasilitas: <ul style="list-style-type: none"> • Membuat beberapa urutan baru • Membangkitkan (satu atau n) urutan selanjutnya
Sorter.java	Kelas untuk mengurutkan sekumpulan ItemBilangan.

Tabel 5. Daftar kelas pada paket Utilitas

D. Testing

Pengujian akan menggunakan data sampel yang berupa data jadwal teknik informatika pada semester Genap 2011-2012 yang terdiri dari 68 kelas teori dan 46 kelas praktikum.

1. Rencana Pengujian

No	Item Pengujian	Jenis Pengujian
1	Pengujian Kelas Permutator	BlackBox
2	Pengujian Kelas MultiPermutator	BlackBox
3	Pengujian pemetaan urutan pada jadwal	BlackBox
4	Pengujian penghitungan nilai Total_tabrakan	BlackBox
5	Pengujian performansi sistem pada jumlah client yang berbeda	BlackBox
6	Pengujian performansi sistem dengan parameter sistem yang berbeda	BlackBox

Tabel 6. Rencana Pengujian

2. Kasus Uji

a. Pengujian kelas Permutator

Identifikasi	Deskripsi	Prosedur Pengujian	Keluaran yang diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
U-1-01	Pengujian Kelas permutator membuat urutan dengan panjang 1	1. Buat sebuah objek permutator yang memiliki panjang 1	Terbuat suatu urutan dengan panjang 1.	Urutan yang terbentuk: 0,	0 ,	Diterima
U-1-02	Pengujian Kelas permutator membuat	1. Buat sebuah objek permutator yang memiliki	Terbuat suatu urutan dengan panjang 10.	Urutan yang terbentuk 0,1,2,3,4,5	0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 ,	Diterima

	urutan dengan panjang 10	panjang 10		,6,7,8,9		
U-1-03	Pengujian Kelas permutator membangkitkan satu urutan selanjutnya dari suatu urutan	1. Buat sebuah objek permutator dengan panjang 5 2. Bangkitkan urutan selanjutnya dari urutan pertama	Terbuat urutan dengan panjang 5 kemudian terbuat urutan selanjutnya dari urutan pertama tersebut.	Urutan pertama: 0,1,2,3,4 Urutan selanjutnya: 0,1,2,4,3	0,1,2,3,4 0,1,2,4,3	Diterima
U-1-04	Pengujian Kelas permutator membangkitkan urutan ke-5 dari urutan pertama	1. Buat sebuah objek permutator dengan panjang 5 2. Bangkitkan urutan ke-5 dari urutan pertama	Terbuat urutan dengan panjang 5 kemudian terbuat urutan ke-5 dari urutan pertama	Urutan pertama: 0,1,2,3,4 Urutan ke-5: 0,1,4,3,2	0,1,2,3,4 0,1,4,3,2	Diterima
U-1-05	Kelas permutator berhenti membangkitkan urutan selanjutnya saat sudah sampai pada urutan paling akhir	1. Buat sebuah objek permutator dengan panjang 5 2. Bangkitkan terus urutan selanjutnya sampai mencapai urutan terakhir	Terbuat urutan dengan panjang 5 kemudian bangkitkan urutan selanjutnya sampai pada urutan terakhir	Urutan pertama: 0,1,2,3,4 Urutan terakhir: 4,3,2,1,0	0,1,2,3,4 ... 4,3,2,1,0	Diterima

Tabel 7. Kasus uji kelas Permutator

b. Pengujian Kelas MultiPermutator

Identifikasi	Deskripsi	Prosedur Pengujian	Keluaran yang diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
U-2-01	Pengujian Kelas MultiPermutator yang terdiri dari dua sub-urutan dengan panjang masing-masing	1. Buat sebuah objek MultiPermutator yang terdiri dari dua sub-urutan dengan panjang masing-masing	Terbuat suatu urutan yang terdiri dari dua sub-urutan dengan panjang 2 dan 3.	Urutan yang terbentuk: 0,1:0,1,2	0,1:0,1,2	Diterima

	2 dan 3	2 dan 3				
U-2-02	Pengujian Kelas MultiPermutator or membangkitkan satu urutan selanjutnya dari suatu urutan	1. Buat sebuah objek MultiPermutator yang terdiri dari dua sub-urutan dengan panjang masing-masing 2 dan 3 2. Bangkitkan urutan selanjutnya dari urutan pertama	Terbuat suatu urutan yang terdiri dari dua sub urutan dengan panjang 2 dan 3, kemudian terbuat urutan selanjutnya dari urutan pertama tersebut.	Urutan pertama: 0,1:0,1,2 Urutan selanjutnya: 0,1:0,2,1	0,1:0,1,2 0,1:0,2,1	Diterima
U-2-03	Pengujian Kelas MultiPermutator or membangkitkan beberapa urutan selanjutnya dari suatu sub-urutan sampai mengubah sub-urutan yang lebih kiri	1. Buat sebuah objek MultiPermutator yang terdiri dari dua sub-urutan dengan panjang masing-masing 2 dan 3 2. Bangkitkan 6 urutan selanjutnya	Terbuat suatu urutan yang terdiri dari dua sub urutan dengan panjang 2 dan 3, kemudian terbuat 6 urutan selanjutnya dari urutan pertama tersebut	Urutan pertama: 0,1:0,1,2 Urutan selanjutnya: 1,0:0,1,2	0,1:0,1,2 1,0:0,1,2	Diterima
U-2-04	Kelas MultiPermutator berhenti membangkitkan urutan selanjutnya saat sudah sampai pada urutan paling akhir	1. Buat sebuah objek MultiPermutator yang terdiri dari dua sub-urutan dengan panjang masing-masing 2 dan 3 2. Bangkitkan terus urutan selanjutnya sampai mencapai urutan terakhir	Terbuat suatu urutan yang terdiri dari dua sub urutan dengan panjang 2 dan 3, kemudian bangkitkan urutan selanjutnya sampai pada urutan terakhir	Urutan pertama: 0,1:0,1,2 Urutan selanjutnya: 1,0:2,1,0	0,1:0,1,2 1,0:2,1,0	Diterima

Tabel 8. Kasus uji kelas MultiPermutator

c. Pengujian pemetaan urutan pada jadwal

Identifikasi	Deskripsi	Prosedur Pengujian	Keluaran yang diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
U-3-01	Pengujian Pemetaan suatu urutan menjadi suatu jadwal	1. Bangkitkan sebuah urutan menggunakan data sample. 2. Petakan urutan tersebut menjadi sebuah jadwal. 3. Lihat urutan pada alamat http://localhost/penjadwalan/pengujian/jadwal.php . 4. Lihat jadwal dari urutan tersebut dengan menekan link pada halaman tersebut.	Terbuat suatu jadwal yang sesuai dengan urutan yang digunakan.	Jadwal yang terbuat sesuai dengan urutan.	Urutan adalah pada gambar 18, dan jadwal dari urutan tersebut pada gambar 19.	Diterima, jadwal hasil pemetaan sesuai dengan urutan.

Tabel 9. Kasus uji pemetaan urutan pada jadwal

Fitness	Urutan
299	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, :

Gambar 21. Urutan yang akan dipetakan menjadi jadwal

Jadwal Nama Matakuliah									
Hari	PP 35								
Hari	Hari ke-0	Hari ke-1	Hari ke-2	Hari ke-3	Hari ke-4	Hari ke-5	Hari ke-0	Hari ke-1	Hari ke-2
sks ke-0	mk:1 Etika Profesi	mk:2 Interaksi Manusia dan Komputer	mk:3 Pengantar Kecerdasan Buatan	mk:5 Keamanan Komputer	mk:6 Analisis dan Desain Sistem Informasi	mk:8 Sistem Informasi Geografis	mk:9 Testing dan Implementasi Sistem	mk:11 Manajemen SDM	mk:13 D Manajen Bisnis
sks ke-1	mk:1 Etika Profesi	mk:2 Interaksi Manusia dan Komputer	mk:3 Pengantar Kecerdasan Buatan	mk:5 Keamanan Komputer	mk:6 Analisis dan Desain Sistem Informasi	mk:8 Sistem Informasi Geografis	mk:9 Testing dan Implementasi Sistem	mk:11 Manajemen SDM	mk:13 D Manajen Bisnis
sks ke-2	mk:1 Etika Profesi	mk:3 Pengantar Kecerdasan Buatan	mk:3 Pengantar Kecerdasan Buatan	mk:5 Keamanan Komputer	mk:6 Analisis dan Desain Sistem Informasi	mk:8 Sistem Informasi Geografis	mk:9 Testing dan Implementasi Sistem	mk:11 Manajemen SDM	mk:14 Pengetah Lingkun
sks ke-3	mk:1 Etika Profesi	mk:3 Pengantar Kecerdasan Buatan	mk:4 Pancasila dan Kewarganegaraan	mk:5 Keamanan Komputer	mk:6 Analisis dan Desain Sistem Informasi	mk:8 Sistem Informasi Geografis	mk:9 Testing dan Implementasi Sistem	mk:12 Aplikasi E-Bisnis	mk:14 Pengetah Lingkun

Gambar 22. Jadwal yang merupakan pemetaan dari suatu urutan

d. Pengujian penghitungan nilai Total_tabrakan

Identifikasi	Deskripsi	Prosedur Pengujian	Keluaran yang diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
U-4-01	Pengujian penghitungan nilai total_tabrakan	1. Bangkitkan sebuah urutan menggunakan data sample. 2. Hitung nilai total_tabrakan 3. Lihat urutan pada alamat http://localhost/penjadwalan/pengujian/jadwal.php .	Ditampilkan konfirmasi visual informasi total_tabrakan	Nilai total_tabrakan dibuktikan hasil penghitungannya	Urutan adalah pada gambar 20, dan pembuktian penghitungan total_tabrakan adalah	Diterima, jadwal hasil pemetaan sesuai dengan urutan.

		4. Lihat hasil penghitungan total dari urutan tersebut dengan menekan link pada halaman tersebut.			pada gambar 21 dan 22.	
--	--	---	--	--	------------------------	--

Tabel 10. Kasus uji penghitungan nilai Total_tabrakan

Fitness	Urutan
<u>299</u>	0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 11 , 12 , 13 , 14 , 15 , 16 , 17 , 18 , 19 , 20 , 21 , 22 , 23 , 24 , 25 , 26 , 27 , 28 , 29 , 30 , 31 , 32 , 33 , 34 , 35 , 36 , 37 , 38 , 39 , 40 , 41 , 42 , 43 , 44 , 45 , 46 , 47 , 48 , 49 , 50 , 51 , 52 , 53 , 54 , 55 , 56 , 57 , 58 , 59 , 60 , 61 , 62 , 63 , 64 , 65 , 66 , 67 , : 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 11 , 12 , 13 , 14 , 15 , 16 , 17 , 18 , 19 , 20 , 21 , 22 , 23 , 24 , 25 , 26 , 27 , 28 , 29 , 30 , 31 , 32 , 33 , 34 , 35 , 36 , 37 , 38 , 39 , 40 , 41 , 42 , 43 , 44 , 45 , :

Gambar 23. Nilai Total_tabrakan dari suatu urutan

sks ke-3	2 : kd=2 sm=6 masuk	21 : kd=3 sm=6 masuk	42 : kd=3 sm=4 masuk	61 : kd=2 sm=4 masuk	70 : kd=1 sm=2 masuk	89 : kd=2 sm=2 masuk	kd=1 sm=2 sudah ada	1	7 : kd=1 sm=6 masuk	24 : kd=1 sm=8 masuk	45 : kd=2 sm=2 masuk	kd=2 sm=2 sudah ada	74 : kd=5 sm=2 masuk	kd=5 sm=2 sudah ada	111 : kd=3 sm=8 masuk	2	10 : kd=1 sm=4 masuk	kd=sm sud ada
sks ke-4	3 : kd=3 sm=6 masuk	21 : kd=3 sm=6 sudah ada	42 : kd=3 sm=4 masuk	62 : kd=3 sm=4 sudah ada	70 : kd=1 sm=2 masuk	89 : kd=2 sm=2 masuk	108 : kd=1 sm=2 sudah ada	3	7 : kd=1 sm=6 masuk	24 : kd=1 sm=8 masuk	45 : kd=2 sm=2 masuk	65 : kd=2 sm=2 sudah ada	75 : kd=1 sm=4 masuk	92 : kd=5 sm=2 masuk	111 : kd=3 sm=8 masuk	1	10 : kd=1 sm=4 masuk	28 kd=sm ma
sks ke-5	3 : kd=3 sm=6 masuk	21 : kd=3 sm=6 sudah ada	42 : kd=3 sm=4 masuk	62 : kd=3 sm=4 sudah ada	71 : kd=2 sm=2 masuk	89 : kd=2 sm=2 sudah ada	108 : kd=1 sm=2 masuk	3	8 : kd=2 sm=6 masuk	24 : kd=1 sm=8 masuk	45 : kd=2 sm=2 masuk	65 : kd=2 sm=2 sudah ada	75 : kd=1 sm=4 masuk	92 : kd=5 sm=2 masuk	111 : kd=3 sm=8 masuk	1	10 : kd=1 sm=4 masuk	28 kd=sm ma
sks ke-6	4 : kd=1 sm=6 masuk	22 : kd=1 sm=2 masuk	43 : kd=4 sm=4 masuk	63 : kd=4 sm=4 sudah ada	71 : kd=2 sm=2 masuk	90 : kd=3 sm=2 masuk	108 : kd=1 sm=2 sudah ada	2	8 : kd=2 sm=6 masuk	25 : kd=2 sm=8 masuk	46 : kd=3 sm=2 masuk	66 : kd=3 sm=2 sudah ada	75 : kd=1 sm=4 masuk	93 : kd=1 sm=4 sudah ada	112 : kd=1 sm=6 masuk	2	11 : kd=2 sm=4 masuk	29 kd=sm ma
sks ke-7	4 : kd=1 sm=6 masuk	22 : kd=1 sm=2 masuk	43 : kd=4 sm=4 masuk	63 : kd=4 sm=4 sudah ada	72 : kd=3 sm=2 masuk	90 : kd=3 sm=2 sudah ada	109 : kd=1 sm=8 masuk	2	8 : kd=2 sm=6 masuk	25 : kd=2 sm=8 masuk	46 : kd=3 sm=2 masuk	66 : kd=3 sm=2 sudah ada	76 : kd=2 sm=4 masuk	93 : kd=1 sm=4 masuk	112 : kd=1 sm=6 masuk	1	11 : kd=2 sm=4 masuk	29 kd=sm ma
sks ke-8	5 : kd=2 sm=6 masuk	Kosong	43 : kd=4 sm=4 masuk	Kosong	72 : kd=3 sm=2 masuk	90 : kd=3 sm=2 sudah ada	109 : kd=1 sm=8 masuk	1	Kosong	25 : kd=2 sm=8 masuk	46 : kd=3 sm=2 masuk	66 : kd=3 sm=2 sudah ada	76 : kd=2 sm=4 masuk	93 : kd=1 sm=4 masuk	112 : kd=1 sm=6 masuk	1	11 : kd=2 sm=4 masuk	30 kd=sm ma
sks ke-9	5 : kd=2 sm=6 masuk	Kosong	Kosong	Kosong	Kosong	Kosong	109 : kd=1 sm=8 masuk	0	Kosong	Kosong	Kosong	Kosong	76 : kd=2 sm=4 masuk	Kosong	Kosong	0	Kosong	30 kd=sm ma
Total tabrakan mahasiswa								16								13		

Total tabrakan pada jadwal = 52

Gambar 24. Konfirmasi total tabrakan mahasiswa pada jadwal

sks ke-3	mk:1 masuk:1 masuk:2	masuk:4 masuk:12 masuk:17 masuk:18	tabrakan:2 masuk:13 masuk:14 masuk:20	tabrakan:1 masuk:6 tabrakan:12	tabrakan:1 tabrakan:2 masuk:5 tabrakan:17	tabrakan:4 tabrakan:5 tabrakan:17 tabrakan:18	mk:41 tabrakan:13 tabrakan:14	12	masuk:4 masuk:5 masuk:6 masuk:7	mk:12 masuk:11 masuk:18	masuk:21 masuk:22 masuk:25	masuk:29 masuk:30 masuk:31	masuk:1 masuk:2 tabrakan:5 masuk:17	tabrakan:4 tabrakan:5 tabrakan:17 tabrakan:18	mk:42 tabrakan: tabrakan:	
sks ke-4	mk:1 masuk:1 masuk:2	mk:9 masuk:4 masuk:12 masuk:17 masuk:18	mk:18 tabrakan:2 masuk:13 masuk:14 masuk:20	mk:23 tabrakan:1 masuk:6 tabrakan:12	mk:27 tabrakan:1 tabrakan:2 masuk:5 tabrakan:17	mk:33 tabrakan:4 tabrakan:5 tabrakan:17 tabrakan:18	mk:41 tabrakan:13 tabrakan:14	12	mk:3 masuk:4 masuk:5 masuk:6 masuk:7	mk:12 masuk:11 masuk:18	mk:19 masuk:21 masuk:22 masuk:25	mk:24 masuk:29 masuk:30 masuk:31	mk:28 tabrakan:7 masuk:10 masuk:17 masuk:33	mk:33 tabrakan:4 tabrakan:5 tabrakan:17 tabrakan:18	mk:42 tabrakan: tabrakan:	
sks ke-5	mk:1 masuk:1 masuk:2	mk:9 masuk:4 masuk:12 masuk:17 masuk:18	mk:18 tabrakan:2 masuk:13 masuk:14 masuk:20	mk:23 tabrakan:1 masuk:6 tabrakan:12	mk:27 tabrakan:1 tabrakan:2 masuk:5 tabrakan:17	mk:33 tabrakan:4 tabrakan:5 tabrakan:17 tabrakan:18	mk:41 tabrakan:13 tabrakan:14	12	mk:3 masuk:4 masuk:5 masuk:6 masuk:7	mk:12 masuk:11 masuk:18	mk:19 masuk:21 masuk:22 masuk:25	mk:24 masuk:29 masuk:30 masuk:31	mk:28 tabrakan:7 masuk:10 masuk:17 masuk:33	mk:33 tabrakan:4 tabrakan:5 tabrakan:17 tabrakan:18	mk:42 tabrakan: tabrakan:	
sks ke-6	mk:2 masuk:1 masuk:2 masuk:3	mk:10 masuk:19	mk:18 tabrakan:2 masuk:13 masuk:14 masuk:20	mk:23 tabrakan:1 masuk:6 masuk:12	mk:27 tabrakan:1 tabrakan:2 masuk:5 masuk:17	mk:33 masuk:4 tabrakan:17 masuk:18	mk:41 tabrakan:13 tabrakan:14	8	mk:3 masuk:4 masuk:5 masuk:6 masuk:7	mk:12 masuk:11 masuk:18	mk:19 masuk:21 masuk:22 masuk:25	mk:24 masuk:29 masuk:30 masuk:31	mk:28 tabrakan:7 masuk:10 masuk:17 masuk:33	mk:34 masuk:1 masuk:34	mk:43 tabrakan: tabrakan:	
sks ke-7	mk:2 masuk:1 masuk:2 masuk:3	mk:10 masuk:19	mk:18 tabrakan:2 masuk:13 masuk:14 masuk:20	mk:23 tabrakan:1 masuk:6 masuk:12	mk:27 tabrakan:1 tabrakan:2 masuk:5 masuk:17	mk:33 masuk:4 tabrakan:17 masuk:18	mk:42 masuk:11 tabrakan:18	7	mk:3 masuk:4 masuk:5 masuk:6 masuk:7	mk:12 masuk:11 masuk:18	mk:19 masuk:21 masuk:22 masuk:25	mk:24 masuk:29 masuk:30 masuk:31	mk:28 tabrakan:7 masuk:10 masuk:17 masuk:33	mk:34 masuk:1 masuk:34	mk:43 tabrakan: tabrakan:	
sks ke-8	mk:2 masuk:1 masuk:2 masuk:3	Kosong	mk:18 tabrakan:2 masuk:13 masuk:14 masuk:20	Kosong	mk:27 tabrakan:1 tabrakan:2 masuk:5 masuk:17	mk:33 masuk:4 tabrakan:17 masuk:18	mk:42 masuk:11 tabrakan:18	6	Kosong	mk:12 masuk:11 masuk:18	mk:19 masuk:21 masuk:22 masuk:25	mk:24 masuk:29 masuk:30 masuk:31	mk:28 masuk:7 masuk:10 masuk:17 masuk:33	mk:34 masuk:1 masuk:34	mk:43 masuk:6 tabrakan:	
sks ke-9	mk:2 masuk:1 masuk:2 masuk:3	Kosong	Kosong	Kosong	Kosong	Kosong	mk:42 masuk:11 masuk:18	0	Kosong	Kosong	Kosong	Kosong	mk:28 masuk:7 masuk:10 masuk:17 masuk:33	Kosong	Kosong	
Total tabrakan dosen									93							

Total tabrakan pada jadwal = 247

Gambar 25. Konfirmasi total tabrakan dosen pada jadwal

e. Pengujian performansi sistem pada jumlah client yang berbeda

Identifikasi	Deskripsi	Prosedur Pengujian	Keluaran yang diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
U-5-01	Pengujian performansi pada 1 client komputer.	1. Menentukan parameter JUMLAHURUTANPERTASK=10000, JUMLAHTASKPERJOB=100, JUMLAHJOB=1 2. Menginstalasi program client pada 1 buah komputer 3. Menjalankan	Menampilkan durasi waktu yang digunakan untuk mengerjakan tugas.	Aplikasi menjalankan tugas sampai selesai.	Setelah dijalankan 3 kali, waktu penyelesaian adalah 4:30 4:29 4:27	Diterima

		aplikasi dan mencatat durasi waktu penyelesaian tugas.				
U-5-02	Pengujian performansi pada 5 buah client komputer	1. Menentukan parameter JUMLAHURUTANPERTASK=10000, JUMLAHTASKPERJOB=100, JUMLAHJOB=1 2. Menginstalasi program client pada 5 buah komputer 3. Menjalankan aplikasi dan mencatat durasi waktu penyelesaian tugas.	Menampilkan durasi waktu yang digunakan untuk mengerjakan tugas.	Durasi waktu yang ditampilkan lebih kecil/cepat dari durasi waktu pada pengujian U-5-01	Setelah dijalankan 3 kali, waktu penyelesaian adalah 0:1:48 0:1:54 0:1:52	Diterima, karena durasi waktu lebih kecil dari pengujian U-5-01.
U-5-03	Pengujian performansi pada 10 buah client komputer	1. Menentukan parameter JUMLAHURUTANPERTASK=10000, JUMLAHTASKPERJOB=100, JUMLAHJOB=1 2. Menginstalasi program client pada 10 buah komputer 3. Menjalankan aplikasi dan mencatat durasi waktu penyelesaian tugas.	Menampilkan durasi waktu yang digunakan untuk mengerjakan tugas.	Durasi waktu yang ditampilkan lebih kecil/cepat dari durasi waktu pada pengujian U-5-01, dan U-5-02	Setelah dijalankan 3 kali, waktu penyelesaian adalah 0:1:26 0:1:13 0:1:24	Diterima, karena durasi waktu lebih kecil dari pengujian U-5-01, dan U-5-02

Tabel 11. Kasus uji performansi dengan jumlah komputer tertentu

f. Pengujian performansi sistem dengan parameter sistem yang berbeda

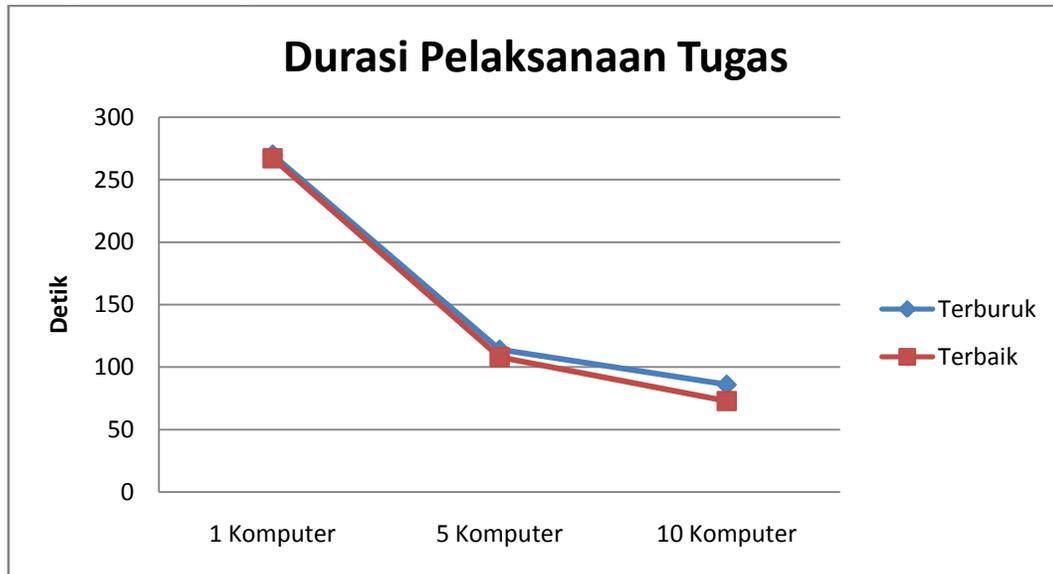
Identifikasi	Deskripsi	Prosedur Pengujian	Keluaran yang diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
U-6-01	Pengujian performansi dengan nilai	1. Menentukan parameter JUMLAHURUTANPERTASK=100,	Menampilkan durasi waktu	Aplikasi menjalankan tugas	Aplikasi Out of memory	Diterima, membuk

	parameter JUMLAHURUTANPER TASK=100, JUMLAHTASKPERJO B=10000, JUMLAHJOB=1	JUMLAHTASKPERJOB= 10000, JUMLAHJOB=1 2. Menginstalasi program client pada 10 buah komputer 3. Menjalankan aplikasi dan mencatat durasi waktu penyelesaian tugas.	yang digunakan untuk mengerjak an tugas.	sampai selesai.	error	tikan bahwa pilihan setting pada pengujia n U-5-x adalah yang tepat.
U-6-02	Pengujian performansi dengan nilai parameter JUMLAHURUTANPER TASK=1000, JUMLAHTASKPERJO B=1000, JUMLAHJOB=1	1. Menentukan parameter JUMLAHURUTANPER TASK=1000, JUMLAHTASKPERJOB= 1000, JUMLAHJOB=1 2. Menginstalasi program client pada 10 buah komputer 3. Menjalankan aplikasi dan mencatat durasi waktu penyelesaian tugas.	Menampil kan durasi waktu yang digunakan untuk mengerjak an tugas.	Aplikasi menjalank an tugas sampai selesai.	Setelah dijalanka n 3 kali, waktu penyeles aian adalah 0:1:29 0:1:29 0:1:25	Diterima , membuk tikan bahwa pilihan setting pada pengujia n U-5-x adalah yang tepat.

Tabel 12. Kasus uji pengaruh parameter sistem pada performansi

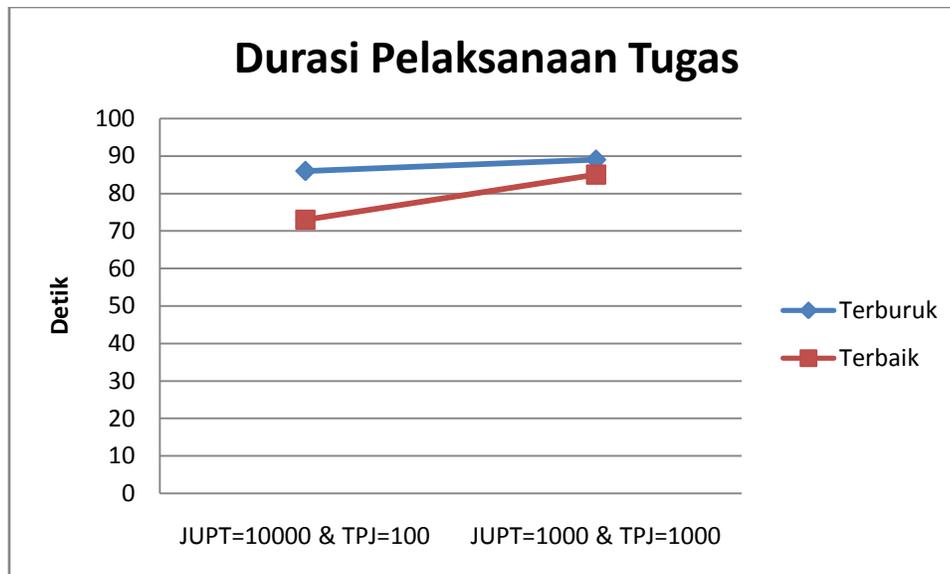
BAB V SIMPULAN, IMPLIKASI DAN SARAN

1. Disain Struktur data dan algoritma urutan (Bab IV) bisa diimplementasikan, hal ini dibuktikan pada pengujian U-1-01 s.d U-1-05, dan U-2-01 s.d U-1-04.
2. Sistem berhasil memanfaatkan distributed processing untuk mempercepat pelaksanaan tugas, hal ini dibuktikan dengan pengujian U-5-01, U-5-02 dan U-5-03. Terlihat bahwa dengan penambahan jumlah komputer yang membantu melaksanakan tugas, durasi pelaksanaan tugas menjadi berkurang.



Gambar 26. Pengaruh jumlah komputer pada Durasi Pelaksanaan Tugas

3. Disain komunikasi dengan memperbanyak urutan yang diproses oleh setiap task (JUPT: Jumlah Urutan Per Task) adalah sangat baik untuk meningkatkan performansi sistem (Bab III) , hal ini dibuktikan pada pengujian U-5-03, U-6-01, dan U-6-02.



Gambar 27. Pengaruh parameter sistem pada Durasi Pelaksanaan Tugas

4. Kecepatan pengolahan solusi pada konfigurasi terbaik yaitu JUPT=10000 dan TPJ=100 adalah :

$$\begin{aligned} (\text{JUPT} * \text{TPJ}) / \text{durasi_rata_rata} &= (10000 * 100) / 81 \text{ detik} \\ &= 12345 \text{ urutan per detik.} \end{aligned}$$

Sehingga untuk menyelesaikan data sample yang berupa:

$$\begin{aligned} \text{Total_kelas_teori!} * \text{Total_kelas_praktikum!} &= 68! * 46! \\ &= 2,48004\text{E}+96 * 5,50262\text{E}+57 \\ &= 1,3647\text{E}+154 (\text{solusi}) \end{aligned}$$

Dibutuhkan waktu pengerjaan sebanyak = $1,3647\text{E}+154 / 12345$

$$\text{dalam Detik} = 1,1054\text{E}+150 (\text{detik})$$

$$\text{Atau} = 3,554\text{E}+142 \text{ (tahun)}$$

Sehingga dapat disimpulkan bahwa implementasi distributed processing belum cukup untuk mempercepat proses pencarian menggunakan metode exhaustive search untuk kasus penjadwalan. Beberapa solusi yang bisa dikembangkan adalah:

- Tetap menggunakan metode exhaustive search yang menerapkan beberapa konstrain yang ditetapkan secara manual, misalkan menetapkan beberapa kelas terikat dengan beberapa ruang tertentu, hal ini akan sangat memperkecil ruang pencarian tetapi juga memiliki efek negatif yaitu memperkecil kemungkinan mendapatkan solusi yang paling optimal.
- Menggunakan metode heuristic seperti Algoritma Genetika, yang dipercepat dengan menggunakan metode distributed processing.

DAFTAR PUSTAKA

- Nugraha, Ivan, 2008, Aplikasi Algoritma Genetik untuk Optimasi Penjadwalan Kegiatan Belajar Mengajar, <http://sutanto.staff.uns.ac.id/files/2008/09/makalahif2251-2008-023.pdf> , diakses tanggal 20 februari 2012.
- Pillay, Nelishiac, A Study into the Use of Hyper-Heuristics to Solve the School Timetabling Problem, <http://delivery.acm.org/>, diakses tanggal 20 februari 2012.
- Smith, Justin, R., The Design and Analysis of Parallel Algorithms.
- Solis, Juan Frausto., Pecina, Federico Alonso, 2007, Analytically Tuned Parameters of Simulated Annealing for the Timetabling Problem, <http://www.wseas.us/e-library/conferences/2007tenerife/papers/572-653.pdf?CFID=68460493&CFTOKEN=63835213>, diakses tanggal 20 februari 2012.
- Solis, Juan Frausto., Vargas, Jaime Mora., Larre, Monica., Ramos, Jose Luis Gomez., 2006, Genetic Algorithm with Forced Diversity for the University TimeTabling Problem, <http://www.wseas.us/e-library/conferences/2006cscf/papers/594-163.pdf?CFID=68460493&CFTOKEN=63835213>, diakses tanggal 20 februari 2012.
- Suyanto, 2007, Artificial Intelligence: Searching, Reasoning, Planning dan Learning, Informatika Bandung.

LAMPIRAN

RIWAYAT HIDUP PENELITI

A. Data Pribadi

1. Nama : Moh. Syafri Tuloli, MT
2. Tempat/tanggal lahir : Gorontalo, 25 Juli 1982
3. Jenis Kelamin : Laki-laki
4. Golongan/Pangkat/NIP : III b/Penata Muda Tk.1/198207252008121004
5. Jabatan Fungsional : Tenaga Pengajar
6. Alamat:
 - a. Rumah : Jln. Sawah Besar No.10, RT.01, RW. 05, Kota Timur, Kota Gorontalo.
 - b. Kantor : Fakultas Teknik
Jl. Jend. Sudirman 6, Kota Gorontalo.
7. Pekerjaan : Dosen Fakultas Teknik, UNG.

B. Riwayat Pendidikan

1. SDN 66 Kota Gorontalo tahun, 1986
2. SMP Negeri 2 Gorontalo, tahun 1994
3. SMA Negeri 3 Gorontalo, tahun 1997
4. S1 Teknik Informatika Universitas Islam Indonesia, tahun 2000
5. S2 Teknik Informatika STEI ITB, Tahun 2005

C. Pengalaman Penelitian

1. Pengaturan tata letak barang dengan menggunakan algoritma genetika, 2007
2. Pengembangan Konten Lokal Interaktif untuk Pembelajaran, 2011

Gorontalo, 15 September 2012
Ketua Peneliti,

Moh. Syafri Tuloli, ST, MT
NIP. 198207252008121004