



SEMINAR NASIONAL INFORMATIKA

# Sertifikat

No. 2283/STMIK PU/SFK/KTA/X/2012

Diberikan Kepada :

**Ade Jamal**

Atas Partisipasi Aktif Sebagai :

**PEMBICARA**

**SEMINAR NASIONAL INFORMATIKA 2012**

Diselenggarakan Oleh :



Medan, 19 Oktober 2012

Ketua  
STMIK Potensi Utama



Rika Rosnelly, SH. M.Kom



Certified Management System  
DIN EN ISO 9001:2008  
Cert. No. 01 100 086092

Ketua Panitia



Mhd. Rusdi Tanjung, S.Kom, M.Ds



SEMINAR NASIONAL INFORMATIKA

# Sertifikat

No. 2192/STMIK PU/SFK/KTA/X/2012

Diberikan Kepada :

**Dr. Ade Jamal**

Atas Partisipasi Aktif Sebagai :

**PESERTA**

**SEMINAR NASIONAL INFORMATIKA 2012**

Diselenggarakan Oleh :



Medan, 19 Oktober 2012

Ketua

STMIK Potensi Utama



Rika Rosnelly, SH. M.Kom



Certified Management System  
DIN EN ISO 9001:2008  
Cert. No. 01 100 086092

Ketua Panitia



Mhd. Rusdi Tanjung, S.Kom, M.Ds

# PROSIDING



SEMINAR NASIONAL INFORMATIKA

# SNif2012

19 OKTOBER 2012



Diselenggarakan oleh :

**STMIK POTENSI UTAMA**



ISSN : 2088 - 9747



## **KOMITE PROGRAM**

Kridanto Surendro, Ph.D (Institut Teknologi Bandung)  
Dr. Rila Mandala (Institut Teknologi Bandung)  
Dr. Husni Setiawan Sastramihardja (Institut Teknologi Bandung)  
Prof. Dr. Tulus (Universitas Sumater Utara)  
Retantyo Wardoyo, Ph.D (Universitas Gajah Mada)  
Agus Harjoko, Ph.D (Universitas Gajah Mada)  
Sri Hartati, Ph.D (Universitas Gajah Mada)  
Dr. Djoko Soetarno (Universitas Bina Nusantara)  
Prof. Dr. Opim Salim Sitompul (Universitas Sumatera Utara)  
Prof. Ahmad Benny Mutiara (Universitas Gunadarma)  
Dr.Febriliyan Samopa (Institut Teknologi Sepuluh Nopember)  
Prof. Dr. Muhammad.Zarlis (Universitas Sumatera Utara)  
Prof. Dr. Joko Lianto Buliali (Institut Teknologi Sepuluh Nopember)  
Prof. Iping Supriana (Institut Teknologi Bandung)  
Dr. Zainal A Hasibuan (Universitas Indonesia)

## **TIM EDITORIAL**

### **PENANGGUNG JAWAB**

Rika Rosnelly, S.H, M.Kom (STMIK Potensi Utama)

### **KETUA PENYUTING**

Muhammad Rusdi Tanjung, S.Kom, M.Ds (STMIK Potensi Utama)

### **WAKIL KETUA PENYUNTING**

Edy Victor Haryanto S. M.Kom (STMIK Potensi Utama)

### **PENYUNTING PELAKSANA**

Ratih Puspasari, M.Kom (STMIK Potensi Utama)  
Lili Tanti, M.Kom (STMIK Potensi Utama)  
Linda Wahyuni, M.Kom (STMIK Potensi Utama)  
Yudhi Andrian, S.Si, M.Kom (STMIK Potensi Utama)  
Khairul Ummi. M.Kom (STMIK Potensi Utama)  
Mas Ayoe Elhias Nasution, S.Kom (STMIK Potensi Utama)  
Rahmadani Pane, S.Kom (STMIK Potensi Utama)  
Evri Ekadiansyah, S.Kom (STMIK Potensi Utama)  
Fithri Mayasari, S.Kom (STMIK Potensi Utama)  
Efani Desi, S.Kom (STMIK Potensi Utama)  
Fitriana Harahap, S.Kom (STMIK Potensi Utama)  
Dian Mayasari, S.Kom (STMIK Potensi Utama)  
Ria Ekasari, S.Kom (STMIK Potensi Utama)  
Budi Triandi, M.Kom (STMIK Potensi Utama)

## **ALAMAT REDAKSI**

Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Potensi Utama Medan  
Jl. K.L.Yos Sudarso Km.6,5 No.3-A Medan (20241)  
Telp (061) 6640525 Fax (061) 6636830  
Email : [snif.potensi-utama.ac.id](mailto:snif.potensi-utama.ac.id) dan [snif.potensi@gmail.com](mailto:snif.potensi@gmail.com)

## **PENERBIT**

Program Studi Teknik Informatika  
STMIK Potensi Utama

## KATA PENGANTAR

Seminar Nasional Informatika (SNIf) merupakan salah satu agenda kegiatan rutin tahunan STMIK Potensi Utama sebagai forum yang mempertemukan Akademisi, Peneliti, Praktisi dan Pengambil Kebijakan dibidang informatika guna penyebaran Ilmu pengetahuan dan Teknologi terkini. Kumpulan makalah dikemas dalam bentuk prosiding dan dikelompokkan sesuai dengan bidang kajian antara lain *Computer Science, Artificial Intelligence, Image Processing, Computer Networking* dan *Security, Multimedia, Wirelles Computing, Interfacing, Information System*, dan *Software Engineering*. Makalah yang diterima berasal dari seluruh Indonesia, makalah yang dimuat dalam Prosiding SNIf 2012 telah melalui tahap evaluasi oleh para reviewer yang berkompeten dibidangnya. Panitia mengucapkan selamat serta terima kasih atas keikutsertaan dalam Seminar Nasional Informatika (SNIf) 2012. Panitia juga mengucapkan terima kasih kepada Pemerintah Daerah Sumatera Utara dan semua pihak yang telah mendukung serta berpartisipasi aktif dalam mensukseskan acara Seminar Nasional ini. Saran dan Kritikan demi menuju kesempurnaan prosiding SNIf sangat diharapkan. Semoga prosiding ini dapat digunakan sebagai salah satu acuan dalam pengembangan teknologi dan peningkatan pembelajaran dibidang Informatika.

Medan, Oktober 2012  
Ketua Panitia

Mhd Rusdi Tanjung, S.Kom, M.Ds

## DAFTAR ISI

	Halaman
1. EXPERT SYSTEM FOR CHOOSING UNIVERSITY MAJOR <b>Rila Mandala, Tjong Wan Sen, Indri Maria</b>	1
2. IMMUNE INSPIRED ALGORITHMS BERBASIS CLONAL SELECTION SEBAGAI ALTERNATIF SOLUSI OPTIMASI (STUDI KASUS : TRAVELING SALESPERSON PROBLEM) <b>Ayi Purbasari</b>	7
3. SISTEM INFORMASI PENGONTROLAN MUTU PRODUK PADA PT SC JOHNSON MANUFACTURING MEDAN <b>Samudra Lubis, Muhammad Irwan Padli Nasution</b>	14
4. IMPLEMENTATION OF EFFECTIVE FOCUSED WEB CRAWLER <b>Ivan M. Siregar, Tjong Wan Sen, Ramero Forester Carlo</b>	25
5. SPASIAL DECISION SUPPORT SYSTEMS DALAM MENDUKUNG PEMANTAUAN STATUS GIZI BALITA DI KOTA PONTIANAK <b>Novi Safriadi, Santri Samanhudi</b>	29
6. MODEL KOMPUTASI PADA MANUSIA DENGAN PENDEKATAN AGENT KOLABORASI <i>BDI MODEL</i> DAN TUJUH PILAR KEHIDUPAN SEBAGAI INSPIRASI UNTUK MENGEMBANGKAN <i>ENACTED SERIOUS GAME</i> <b>Ririn Dwi Agusti, Iping Supriana Suwardi</b>	35
7. INFORMATION TECHNOLOGY (IT) INNOVATION IMPLEMENTATION : A SYNTHESIS OF PAST RESEARCH ON THEORETICAL PERSPECTIVES <b>M Qomarul Huda, Husnayati Hussin</b>	41
8. PENGUKUR VOLUME ZAT CAIR RESIDU BERBASIS GELOMBANG ULTRASONIK MENGGUNAKAN ARDUINO PADA PT.PLN PERSERO BELAWAN <b>Iwan Fitrianto Rahmad, M.Kom, Fahru Rozi</b>	48
9. VALIDASI DOKUMEN DIGITAL MENGGUNAKAN TEKNIK ENKRIPSI ASIMETRIK RSA <b>Nenny Anggraini, M.Kom, Feri Fahrianto</b>	55
10. PERANCANGAN <i>IT GOVERNANCE</i> LAYANAN AKADEMIK MENGGUNAKAN <i>FRAMEWORK INFORMATION TECHNOLOGY INFRASTRUCTURE LIBRARY (ITIL)</i> VERSI 3 <b>Ichsan Taufik, Kridanto Surendro</b>	58
11. PENYELIDIKAN SISTEM INFORMASI AKADEMIK UNIVERSITAS DI ACEH BERBASIS SERVICE ORIENTED ARCHITECTURE (SOA) <b>Taufiq</b>	66

73. SEGMENTASI VOLUME 3D CITRA USG BERDASARKAN WARNA RGB DENGAN METODE GAUSSIAN 476  
**Pandapotan Siagian Eriya**
74. RANCANGAN SISTEM INFORMASI SOSIALISASI BEASISWA UNGGULAN BPKLN KEMDIKBUD RI DALAM RANGKA PENINGKATAN JUMLAH PENERIMA BEASISWA UNGGULAN BAGI MAHASISWA BERPRESTASI. 486  
**Tuti Hartati**
75. PORTAL LAYANAN ORANG TUA MAHASISWA BERBASIS FRAMEWORK CODE IGNITER 495  
**Mohamad Irfan, Hilman Ali**
76. IMPLEMENTASI PENJADWALAN KULIAH SECARA PARALEL MENGGUNAKAN METODE HILL CLIMBING PADA PLATFORM GPU 502  
**Faisal Hardi, Ade Jamal**
77. PENERAPAN COMPONENT BASED SOFTWARE ENGINEERING DALAM PENGEMBANGAN WEBSITE SISTEM INFORMASI SEKOLAH (STUDI KASUS : SMA SANTO PAULUS PONTIANAK) 507  
**Sandy Kosasi, David**
78. PEMODELAN INFORMASI SISTEM PERPUSTAKAAN PADA PT. STIKOM CKI - JAKARTA 513  
**Sugiyono, Mesra Betty Yel , Dadang Iskandar Mulyana**
79. SISTEM INFORMASI NILAI SISWA PADA SD PUTRA 1 JAKARTA 519  
**Muryan Awaludin, Dwi Budi Srisulistiwati, Sugiyono**
80. ANALISIS PENGGUNAAN TEKNOLOGI INFORMASI PADA RUMAH SAKIT STELLA MARIS MAKASSAR 528  
**N. Tri Suswanto Saptadi**
81. IMPLEMENTASI PEMODELAN MULTI KRITERIA (PMK) PADA SISTEM PENDUKUNG KEPUTUSAN PENGUJIAN MUTU BAN SEPEDA MOTOR 536  
**Muliadi**

# IMPLEMENTASI PENJADWALAN KULIAH SECARA PARALEL MENGUNAKAN METODE HILL CLIMBING PADA PALTFORM GPU

Faisal Hardi Ade Jamal

Jurusan Teknik Informatika, Fakultas Sains dan Teknologi  
Universitas Al Azhar Indonesia, Jl. Sisingamangaraja, Jakarta 12110  
Tel. 7244456, fax. 7244767, website: uai.ac.id

## Abstrak

Penyusunan penjadwalan kuliah dengan batasan masalah yang dimilikinya dapat dicari solusinya dengan menggunakan algoritma pencarian dan optimalisasi. Pada sebuah artikel[8] disebutkan bahwa, metode "Hill Climbing" berhasil mencari solusi penjadwalan kuliah. Hill Climbing membutuhkan banyak iterasi (perulangan) untuk menemukan jadwal kuliah sesuai yang diinginkan. Terdapat banyak cara untuk menggantikan proses iterasi, salah satunya adalah menjalankan proses secara bersamaan pada waktu yang sama (*Parallel Processing*). Graphics processing unit (GPU) yang dulu latar belakang kehadirannya digunakan untuk merender suatu grafik, kini dapat digunakan untuk hal yang lain seperti, menjalankan sebuah proses secara paralel dengan tanpa beban dari aplikasi lainnya.

**Kata kunci :** Penjadwalan perkuliahan, Hill Climbing, Parallel Processing, GPU

## I. PENDAHULUAN

Kegiatan penyusunan jadwal kuliah yang baik dan benar merupakan kunci sukses bagi terselenggaranya kegiatan belajar dan mengajar di Universitas. Dalam setahun, sekurangnya dua kali penjadwalan kuliah dilakukan oleh setiap Biro Akademik Universitas. Kegiatan ini menjadi masalah umum sebagai kegiatan yang rutin dilakukan. Pencarian solusi penjadwalan kuliah yang baik dan benar haruslah tidak mengabaikan salah satu atau lebih batasan masalah yang ada. Kita dapat mengatakan bahwa pencarian solusi masalah penjadwalan kuliah adalah optimalisasi kombinatorial. Optimalisasi kombinatorial maksudnya adalah proses mengoptimalkan fungsi objektif dari permasalahan yang sifatnya kombinasi dengan menggunakan sumberdaya yang terbatas. Dimana jumlah ruangan dan slot waktu yang terbatas menjadi sumberdayanya.

Permasalahan penjadwalan kuliah seperti ruang kelas yang memiliki tempat duduk yang terbatas. Serta dosen dan mahasiswa yang tidak boleh bentrok dalam suatu penjadwalan kuliah, merupakan kombinasi permasalahan yang akan kita cari urutan penempatan kelasnya (solusinya) dengan cara menjabarkan permasalahan umum tersebut kedalam multidimensional ruang dan waktu dalam satu sumberdaya yang terbatas. Tidak lupa, bahwa setiap permasalahan yang ada tidak boleh diabaikan. jika diabaikan solusi penjadwalan

kuliah yang dihasilkan akan tidak seperti yang diinginkan. Sehingga, penjadwalan kuliah yang baik dan benar akan kita dapat.

Pencarian solusi penjadwalan kuliah dapat menggunakan metode pencarian Hill Climbing. Hill climbing merupakan bagian dari keluarga pencarian heuristik (local search). Secara garis besar pencarian heuristik adalah metode pencarian yang tidak selalu menemukan solusi yang terbaik akan tetapi menjamin ditemukan solusi yang memuaskan dalam waktu singkat. Referensi [6] menjelaskan bahwa solusi penjadwalan kuliah yang didapat tidak pernah mendapatkan solusi yang spesifik atau tidak dapat selalu dapat ditetapkan (Non-deterministik).

Parallel processing adalah metode menjalankan proses yang banyak secara bersamaan dalam satu waktu yang sama. . Idealnya, parallel prosesing membuat program berjalan lebih cepat karena menjalankan proses pada lebih dari satu sumber daya komputasi.

## II. DASAR TEORI

Algoritma Optimalisasi dapat didefinisikan sebagai algoritma atau metode numerik untuk menemukan nilai  $x$  sedemikian hingga menghasilkan  $f(x)$  yang bernilai terkecil atau terbesar untuk suatu fungsi  $f$  yang diberikan, yang mungkin disertai dengan beberapa batasan pada  $x$ . Di sini,  $x$  bisa berupa skalar atau vektor dari nilai-nilai kontinu maupun diskrit.

Berdasarkan metode operasinya, algoritma optimasi (AO) dapat dibagi ke dalam dua kelas besar, yaitu algoritma deterministik dan probalistik. Pada setiap langkah eksekusi dalam algoritma deterministik, terdapat maksimum satu jalan untuk diproses. Jika tidak ada jalan, berarti algoritma sudah selesai. Algoritma deterministik sering digunakan untuk masalah yang memiliki relasi yang jelas antara karakteristik calon solusi dengan utilitasnya. Dengan demikian, ruang pencarian dapat dieksplorasi menggunakan, misalnya, metode *branch and bound* atau *divide and conquer scheme*. Berdasarkan akurasi dan kecepatannya, algoritma optimasi dibedakan menjadi dua, yaitu : Optimasi *Online* dan Optimasi *Offline*.

Optimasi *Online*, sesuai dengan namanya, optimasi ini ditujukan untuk permasalahan yang membutuhkan solusi dalam waktu cepat dan biasanya permasalahan tersebut terjadi secara berulang-ulang. Untuk mendapatkan solusi yang baik, biasanya para praktisi melakukan kompromi antara optimalitas dan kecepatan. Pada algoritma optimasi ini, akurasi tidak harus yang paling baik (optimum global).

Optimasi *Offline*, optimasi jenis ini ditujukan untuk permasalahan yang membutuhkan solusi tidak dalam waktu cepat dan biasanya masalah ini terjadi dalam periode waktu yang lama. Untuk ruang masalah yang tidak terlalu besar, algoritma optimasi yang bersifat deterministik ini biasanya menghasilkan solusi yang lebih baik dibandingkan optimasi probalistik.

Berdasarkan analogi dan nama yang digunakan, algoritma optimasi bisa dibedakan menjadi dua: algoritma minimasi dan algoritma maksimasi. Algoritma minimasi yaitu Algoritma yang menggunakan analogi meminimalkan sesuatu pada dunia nyata. Algoritma maksimasi. Algoritma yang menggunakan analogi memaksimalkan sesuatu pada dunia nyata.

*Hill climbing* adalah sebuah algoritma pencarian yang digunakan untuk mendapatkan urutan langkah-langkah penyelesaian masalah optimalisasi. *Hill Climbing* secara berulang-ulang memilih tetangga yang terbaik guna mencapai suatu titik dimana tidak ada lagi tetangga yang lebih baik. Metode *Hill Climbing* sering digunakan untuk menemukan *Travelling Salesman Problem (TPS)*. Dengan menggunakan kondisi awal (nilai awal) yang bebas membuat *Hill Climbing* secara *incremental* (bertahap-tahap) berusaha mencari dan berpindah ke nilai solusi yang terbaik

sehingga mencapai suatu keadaan yang dimana solusi itu tidak dapat lagi tergantikan.

*Modified Hill Climbing*[8] merupakan algoritma pencarian local yang mencari kemungkinan solusi yang terdekat pada suatu keadaan dimana solusi tersebut ditentukan atau dibatasi sehingga solusi yang ada tidak diambil seluruhnya.

*Parallel processing* adalah penggunaan lebih dari satu pusat pengolahan (*central processing unit / CPU*) untuk menjalankan sebuah program secara simultan. Pemrograman paralel adalah teknik pemrograman komputer yang memungkinkan eksekusi perintah/operasi secara bersamaan baik dalam komputer dengan satu (prosesor tunggal) ataupun banyak (prosesor ganda dengan mesin paralel) CPU. Tujuan utama dari pemrograman paralel adalah untuk meningkatkan performa komputasi. Semakin banyak hal yang bisa dilakukan secara bersamaan (dalam waktu yang sama), semakin banyak pekerjaan yang bisa diselesaikan.

CUDA™ adalah suatu model platform komputasi secara paralel dan diperkenalkan pertama kali oleh NVIDIA untuk dapat diprogram oleh pengguna (pengembang) software aplikasi. Dengan CUDA, memungkinkan suatu aplikasi berjalan dengan memanfaatkan komputasi paralel *graphics processing unit (GPU)*.

*Synchronization Function* Merupakan salah satu fasilitas yang diberikan oleh CUDA untuk para developer yang memiliki kebutuhan untuk mengatur semua *thread* agar menunggu selesai pekerjaan *thread* tetangganya sampai pada line tertentu..

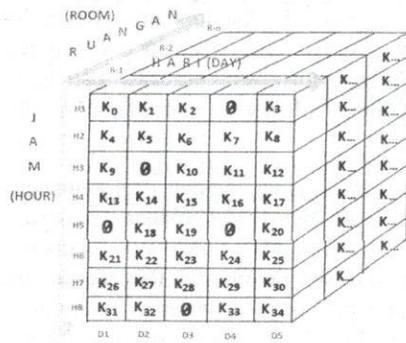
*Atomic function* merupakan fungsi yang beroperasi untuk melakukan *read-modify-write* pada 32-bit atau 64-bit word pada *global-memory* atau *shared-memory*. *Atomic Function* memberikan jaminan bahwa tidak akan melibatkan *thread* pada block berbeda, jadi hanya *thread* pada block yang sama saja yang terikat dalam proses *atomic function* ini.

### III. ANALISIS DAN PERANCANGAN

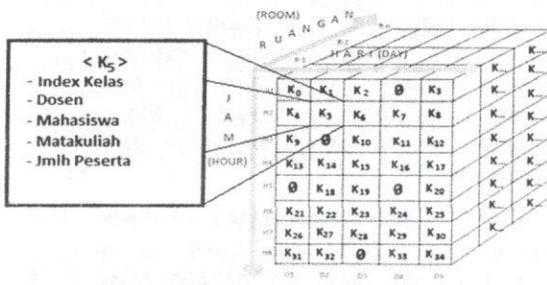
#### Model Data

Pemodelan data penjadwalan kuliah penulis petakan menjadi matrik 3-dimensi (x, y, z), dimana setiap arah dimensi memiliki arti. Pada gambar dibawah ini dijelaskan model data penjadwalan kuliah berbentuk matrik 3-dimensi, untuk dimensi-x menandakan *hari (day)*, dimensi-y menandakan *jam (hour)* dan

yang terakhir dimensi-z menandakan ruangan (room).



Gambar Model Data Penjadwalan Kuliah



Gambar Elemen Kelas pada Matrik Jadwal Kuliah

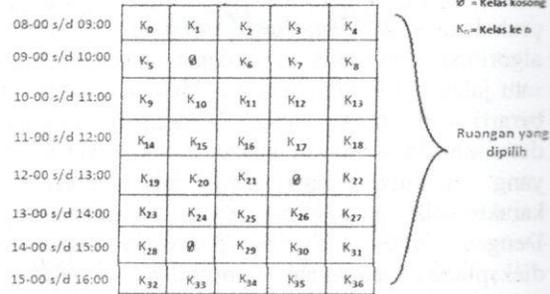
Apabila kita mengambil satu elemen kelas pada matrik jadwal kuliah, sebagaimana gambar diatas. Maka sebuah element kelas akan memiliki arti :

Tabel Properti Jadwal Kuliah Pada Elemen Kelas

<b>Index kelas :</b>	Menjelaskan index kelas yang merujuk pada data kuliah
<b>Dosen :</b>	Menjelaskan siapa dosen yang mengajar
<b>Mahasiswa :</b>	Menjelaskan siapa prodi mahasiswa yang mengikuti kelas ini
<b>Matakuliah :</b>	Menjelaskan matakuliah yang diajarkan di kelas tersebut
<b>Jmlh Peserta :</b>	Menjelaskan jumlah perserta pengikut kelas kuliah

Apabila kita mengambil salah satu blok ruangan pada model data penjadwalan kuliah 3-dimensi pada gambar diatas, maka model data akan terlihat 2-dimensi (x, y) ~ dimensi-x untuk hari (Day) dan dimensi-y untuk jam (Hour). Ruangan tersebut akan terdiri dari 5 kolom hari dan 8 baris jam kuliah, sehingga terdapat 40 kelas (5 \* 8) dalam satu ruangan.

S E M I N A R N A S I O N A L I N F O R M A T I K A 2 0 1 2



Gambar Model Data Penadwalan Kuliah 2-Dimensi

Rumus Loncatan (Stride) Pada Matrik Jadwal

Secara teknis pemograman untuk pemodelan tipe data pada serial maupun parallel (platform CUDA), penulis mendefinisikan model data jadwal kuliah 3-dimensi (sebagaimana dijelaskan diatas) menjadi 1-dimensi array(larik) pada teknis pemograman. seperti :

```
intmatrik_jadwal_kuliah[ND * NH * NR * NM]
```

Dimana :

- ND : Number of Day ~ jumlah hari
- NH : Number of Hour ~ jumlah jam
- NR : Number of Room ~ jumlah ruangan
- NM : Number of Matrik ~ jumlah matrik (batas duplikasi)

Sebagai contoh :

```
intmatrik_jadwal_kuliah[5 * 8 * 10 * 40]
intmatrik_jadwal_kuliah[16000]
```

Jika terdapat 40 matrik jadwal yang digunakan untuk menduplikasi state (matrik jadwal kuliah), maka secara linier terdapat 16000 pointer (array) untuk menampung jadwal kelas kuliah.

```
[K0, K1, K2, K3, ..... , K(16000-1)]
```

Dari rentetan jadwal kelas kuliah tersebut, munculah pertanyaan bagaimana cara menentukan bahwa apabila kita mengambil suatu element (jadwal kelas) pada array linear tersebut, maka element tersebut milik jadwal matrik ke berapa?. Untuk menjawab pertanyaan itu, penulis menggunakan rumus stride (lompatan). Rumus ini penulis definisikan sendiri yang tujuannya untuk menemukan sebuah elemen kelas kuliah dan dapat mengetahui sampai mana batas deretan elemen kelas kuliah untuk suatu matrik jadwal.

Adapun rumus *stride* (lompatan) sebagai berikut :

$$\text{matrik\_jadwal\_kuliah}[\text{index\_kelas} + \text{stride}]$$

$$= \text{matrik\_jadwal\_kuliah}[\text{index\_kelas} + (\text{ND} * \text{NH} * \text{NR} * \text{NM})]$$

Dimana :

**index\_kelas** : index kelas yang diinginkan ~ harus kurang dari  $(\text{ND} * \text{NH} * \text{NR})$

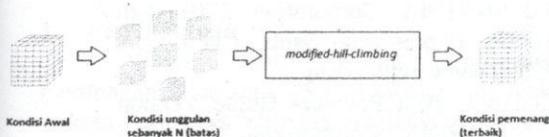
**stride** : bilangan untuk mengelompokkan matrik jadwal keberapa- $(\text{ND} * \text{NH} * \text{NR} * \text{NM})$

#### Implementasi (Perancangan) Penjadwalan Kuliah

Karena tugas akhir ini diambil dari sebuah artikel penelitian (reverensi), serta tujuan penulis mengerjakan tugas akhir ini adalah karena ingin mengetahui perbedaan *efisien waktu* antara penjadwalan kuliah yang dijalankan secara serial (reverensi) dengan penjadwalan kuliah yang dijalankan secara parallel (tugas akhir penulis).

Maka untuk mendapatkan hasil perbandingan waktu diantara keduanya, penulis mengembangkan 2 (dua) aplikasi penjadwalan kuliah yang berjalan secara serial dan berjalan secara parallel. Masing-masing aplikasi menggunakan metode Hill Climbing dan model data yang sama.

#### Penggunaan Modified Hill Climbing



Gambar Proses Modified Hill Climbing

Berdasarkan gambar diatas, dari kondisi awal matrik jadwal atau yang kita kenal pada istilah Hill Climbing sebagai *inisial state*, akan dihasilkan tetangga (*neighbour state*) yang berpotensi menjadi matrik jadwal yang diinginkan.

```

function MODIFIED-HILL-CLIMBING(problem)
inputs:      problem: a problem
localvariables: current, neighbor: a state
              successors[], a state
returns: a state that is local maximum
current ← INITIAL-STATE(problem)
for iteration ← 1 to ∞ do
successors[] ← SCATTERED-SUCCESSORS(current)
neighbor ← a highest valued of successors[]
if VALUE(current) ≤ VALUE(neighbor) then
current ← neighbor
if VALUE(current) = OBJECTIVE-VALUE then
return current

```

Pada Modified Hill Climbing tidak semua tetangga matrik jadwal akan dipilih, akan tetapi proses pemilihannya dilakukan secara random dengan batas tertentu. Dari matrik jadwal tetangga yang tersedia, maka akan dipilih matrik jadwal yang memiliki total konflik paling minimum, total konflik minimum itu adalah konflik dosen, ruangan dan mahasiswa berjumlah sedikit (minimum). Sehingga matrik jadwal dengan total konflik minimum itulah yang menjadi matrik pemenang atau *state* akhir.

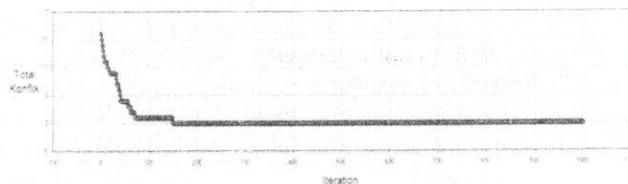
#### IV.EVALUASI DAN PENGUJIAN

Pengujian (*atau Percobaan*) penyusunan penjadwalan kuliah yang penulis lakukan terbagi menjadi dua bagian, diantaranya adalah percobaan pada bagian serial dan parallel. Diharapkan dari percobaan ini akan didapatkan perbedaan efisiensi waktu yang berbeda antara menjalankan aplikasi penjadwalan kuliah secara serial dengan secara parallel.

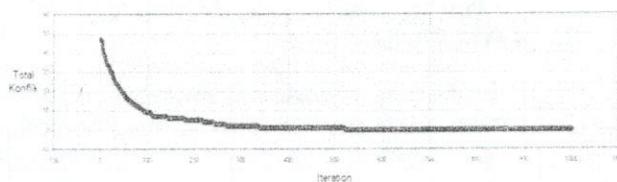
##### Percobaan Serial

	Jumlah Ruang	Jumlah Kelas (Kuliah)
Percobaan Pertama	10	350
Percobaan Kedua	15	550
Percobaan Ketiga	20	662
Percobaan Keempat	25	662

##### Pada percobaan pertama



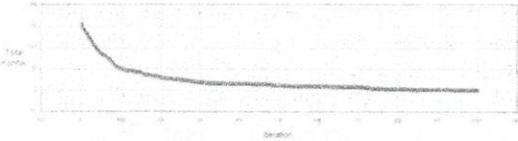
##### Percobaan Kedua



##### Percobaan Ketiga



**Percobaan Keempat**



Dari keempat percobaan diatas kita dapatkan hasil sebagai berikut :

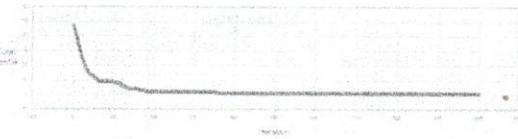
	Jumlah Ruang	Jumlah Kelas	Jumlah Konflik Awal	Jumlah Konflik Akhir	Waktu (detik)
Percobaan Pertama	10	350	17	0	73.40
Percobaan Kedua	15	550	47	1	249.53
Percobaan Ketiga	20	662	84	2	424.75
Percobaan Keempat	25	662	154	4	418.13

**Percobaan Paralel**

**Percobaan Pertama**



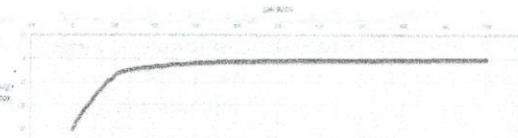
**Percobaan Kedua**



**Percobaan Ketiga**



**Percobaan Keempat**



Dari keempat percobaan diatas kita dapatkan hasil sebagai berikut :

	Jumlah Ruang	Jumlah Kelas	Jumlah Konflik Awal	Jumlah Konflik Akhir	Waktu (detik)
Percobaan Pertama	10	350	16	0	0.47
Percobaan Kedua	15	550	47	0	0.59
Percobaan Ketiga	20	662	83	0	0.81
Percobaan Keempat	25	662	154	1	0.91

**V. KESIMPULAN**

Dari rangkaian percobaan yang telah dilakukan, maka terdapat hasil perbedaan waktu yang sangat jauh antara menjalankan aplikasi penjadwalan kuliah proses secara serial dengan parallel. Penjadwalan kuliah

dengan menggunakan metode Hill Climbing yang dijalankan secara paralel efisien waktu sangatlah berbanding jauh dengan proses serial.

Kemudian tingkat perbedaan waktu sangatlah dekat dari tiap-tiap percobaan yang telah dilakukan dalam proses paralel. Sehingga jikakita melakukan percobaan yang lain dengan jumlah data yang lebih besar dari keempat percobaan yang kita lakukan maka waktu yang digunakan tidaklah berbeda jauh dari serangkaian percobaan yang telah kita lakukan. Sebagaimana yang terlihat pada gambar 4.6 grafik waktu tempuh percobaan paralel, antara satu percobaan dengan percobaan lainnya waktu yang digunakan tidak terlalu berbeda jauh.

	Jumlah Ruang	Jumlah Kelas (Kuliah)
Percobaan Pertama	10	350
Percobaan Kedua	15	550
Percobaan Ketiga	20	662
Percobaan Keempat	25	662

**DAFTAR PUSTAKA**

- [1] Suyanto. 2010. Algoritma Optimasi (deterministik atau probabilistik). Yogyakarta : Graha Ilmu
- [2] Fundamental of Genetic Algorithm: AI Course Slide [WWW.myreaders.info](http://WWW.myreaders.info) , Juni 01 2010
- [3] CUDA C Programming Guide Version 3.2 (NVIDIA Corporation 2701 San Tomas Expressway Santa Clara, CA 95050 [www.nvidia.com](http://www.nvidia.com))
- [4] CUDA C Best Practices Guide Version 3.2 (NVIDIA Corporation 2701 San Tomas Expressway Santa Clara, CA 95050 [www.nvidia.com](http://www.nvidia.com))
- [5] Introduction to CUDA C San Jose Convention Center | September 20, 2010
- [6] Thomas Weise. 2007. Global Optimization - Theory and Application. E-book.
- [7] <http://www.nvidia.com/object/gpu.html>
- [8] A. Jamal, "Solving University Course Scheduling Problem using Improved Hill Climbing Approach", *Proceeding of the International Joint Seminar in Engineering*, Jakarta, Indonesia, August 2008, pp. 210-241.

Pen  
yang  
dan  
dilal  
Sant  
men  
sem  
men  
tekn  
Mic  
men  
men  
kom  
hasil  
baha  
  
Kat  
Mic  
  
1.  
  
mert  
di ko  
ini l  
calor  
maju  
di S  
jumlah  
42  
infor  
deng  
peny  
yang  
data  
  
juga  
seper  
siswa  
peng  
pada  
to  
infor  
guru.