

**LAPORAN AKHIR
PENELITIAN FUNDAMENTAL**
(Teknik Elektro dan Informatika)



**APLIKASI ALGORITMA GENETIKA
PADA SKEMA PENJADWALAN KOMPUTASI TERDISTRIBUSI
(STUDI KASUS PADA *SOFTWARE-DEFINED RADIO*)**
Tahun ke-2 dari rencana 2 tahun

Dr. Eko Marpanaji/008066709 (Ketua)
Adi Dewanto, M. Kom./0028127203 (Anggota 1)
Didik Hariyanto, M.T./0002057705 (Anggota 2)

Dibiayai oleh:
DIPA Universitas Negeri Yogyakarta dengan Surat Pernjanjian
Penugasan dalam rangka Pelaksanaan Program Penelitian Desentralisasi BOPTN
Skim: Fundamental Tahun Anggaran 2014
Nomor: 231/Fund.-BOPTN/UN34.21/2014 Tanggal 17 Maret 2014

UNIVERSITAS NEGERI YOGYAKARTA
LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA
MASYARAKAT
OKTOBER 2014

HALAMAN PENGESAHAN

Judul Kegiatan : APLIKASI ALGORITMA GENETIKA PADA SKEMA
PENJADWALAN KOMPUTASI TERDISTRIBUSI (STUDI KASUS
PADA SOFTWARE-DEFINED RADIO)

Peneliti / Pelaksana

Nama Lengkap : Dr. EKO MARPANAJI MT
NIDN : 0008066709
Jabatan Fungsional :
Program Studi : Pendidikan Teknologi Dan Kejuruan
Nomor HP : 08164264136
Surel (e-mail) : eko@uny.ac.id

Anggota Peneliti (1)

Nama Lengkap : ADI DEWANTO M.Kom
NIDN : 0028127203
Perguruan Tinggi : Universitas Negeri Yogyakarta

Anggota Peneliti (2)

Nama Lengkap : DIDIK HARIYANTO M.T.
NIDN : 0002057705
Perguruan Tinggi : Universitas Negeri Yogyakarta

Institusi Mitra (jika ada)

Nama Institusi Mitra :
Alamat :
Penanggung Jawab :

Tahun Pelaksanaan : Tahun ke 2 dari rencana 2 tahun

Biaya Tahun Berjalan : Rp. 40.000.000,00

Biaya Keseluruhan : Rp. 115.000.000,00

Mengetahui
Dekan Fakultas Teknik



(Dr. Moch. Bruri Triyono)

NIP/NIK 195602161986031003

Yogyakarta, 30 - 10 - 2014,
Ketua Peneliti,

(Dr. EKO MARPANAJI MT)

NIP/NIK 196706081993031002

Menyetujui,
Ketua LPPM



(Prof. Dr. Anik Ghufro)

NIP/NIK 196211111988031001

RINGKASAN

Kendala utama dalam mewujudkan sistem pengolah sinyal digital frekuensi tinggi adalah dalam hal kebutuhan komputasi untuk memenuhi pemrosesan laju bit yang sangat tinggi. Komputasi terdistribusi dapat digunakan untuk meningkatkan kecepatan proses komputasi, namun sampai saat ini belum banyak yang meneliti untuk proses komputasi pengolah sinyal digital frekuensi tinggi khususnya dalam pengembangan sofradio, softtv, dan sofradar. Permasalahan utama dalam hal komputasi terdistribusi adalah penjadwalan. Skema penjadwalan *weighted-selective* dan metoda paralelisme data (*data-parallelism*) yang telah dilakukan oleh peneliti sebelumnya akan selalu memprioritaskan node yang memiliki kemampuan komputasi paling tinggi. Selain itu, metoda paralelisme data menyebabkan beberapa node tidak melakukan komputasi sama sekali karena nilai prioritasnya paling rendah, sehingga beban kerja komputasi paralel menjadi tidak merata.

Tujuan penelitian ini adalah memperbaiki kelemahan skema penjadwalan *weighted-selective* dengan metoda paralelisme data. Penelitian ini mengembangkan komputasi terdistribusi dengan mengambil studi kasus tugas-tugas *Software-Defined Radio* (SDR) dengan menggunakan metoda paralelisme tugas (*task-parallelism*) sehingga tiap-tiap tugas dapat dibagikan kepada seluruh node komputasi. Optimasi skema penjadwalan menggunakan algoritma genetika sehingga diperoleh sistem penjadwalan yang optimal dan pembagian beban kerja lebih merata. Penelitian ini merupakan lanjutan dari penelitian-penelitian sebelumnya tentang aspek-aspek komputasi pengolah sinyal digital frekuensi tinggi, arsitektur perangkat keras dan perangkat lunak, serta dekomposisi tugas-tugas komputasi dengan mengambil studi kasus dalam pengembangan SDR.

Metoda yang digunakan dalam penelitian ini diawali dengan proses identifikasi tugas-tugas komputasi pengolahan sinyal digital frekuensi tinggi dengan studi kasus. tugas-tugas komputasi untuk menjalankan fungsi radio dalam sistem SDR. Setelah itu, dilakukan dekomposisi tugas sehingga diperoleh sebuah grafik tugas komputasi SDR lengkap dengan parameter tiap node dalam grafik tersebut. Parameter tiap node dapat diperoleh dengan menghitung nilai ketinggian (*height*) masing-masing tugas dan menguji waktu eksekusi masing-masing tugas. Berdasarkan grafik tugas tersebut, kemudian dilakukan optimasi penjadwalan menggunakan algoritma genetika. Tahap awal yang harus dilakukan adalah menentukan fungsi *fitness* berdasarkan parameter yang digunakan untuk mengukur kinerja sebuah jadwal, yaitu waktu komputasi. Sebuah jadwal dikatakan optimal jika waktu komputasinya paling rendah dengan waktu idle paling kecil. Operator genetika yang digunakan disesuaikan dengan konstrain proses komputasi terdistribusi. Kinerja penjadwalan untuk komputasi terdistribusi dengan metoda paralelisme tugas diamati berdasarkan waktu eksekusi total penyelesaian setiap jadwal serta waktu idle sebagai indikator pemerataan beban kerja.

Pertanyaan penelitian yang akan dijawab dalam penelitian ini adalah bagaimana formula fungsi *fitness* untuk skema penjadwalan tugas-tugas komputasi SDR? Apakah dengan skema penjadwalan menggunakan algoritma dapat menghasilkan penjadwalan yang optimal dilihat dari waktu eksekusi dan apakah penyeimbangan beban kerja menjadi lebih merata antar tiap node komputasi?

Hasil penelitian ini diharapkan dapat menjadikan konsep dasar dalam mengembangkan komputasi terdistribusi pengolahan sinyal digital frekuensi tinggi yang memiliki laju bit sangat tinggi sehingga dapat digunakan untuk pengembangan sofradio, softtv ataupun sofradar. Luaran penelitian ini adalah metode penjadwalan komputasi dan sebuah prototipe komputasi terdistribusi khususnya dalam pengembangan SDR. Laporan ini merupakan rangkuman hasil kegiatan penelitian Tahap I (2013) dan Tahap II (2014).

PRAKATA

Puji syukur ke hadirat Allah SWT karena dengan kemurahanNya telah memberi jalan untuk selesainya penelitian ini.

Kami berterima kasih kepada Almamater, Rektor Universitas Negeri Yogyakarta, Ketua Lembaga Penelitian UNY, Dekan Fakultas Teknik, Ketua Jurusan Pendidikan Teknik Elektronika, atas kesempatan dan pendanaan yang diberikan sehingga penelitian ini dapat terlaksana.

Terima kasih juga disampaikan kepada civitas akademika Pendidikan Teknik Elektronika FT UNY dan Puskom UNY, beserta rekan-rekan semua yang tidak dapat disebutkan satu persatu atas dukungan yang telah diberikan dalam bentuk apapun sehingga penulis dapat menyelesaikan penelitian ini.

Kami menyadari bahwa laporan penelitian ini masih banyak kekurangan dan kelemahannya. Untuk itu, kritik serta saran akan kami terima dengan senang hati, demi perbaikan laporan penelitian ini.

Hormat kami,

Peneliti

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN	ii
RINGKASAN.....	iii
PRAKATA	iv
DAFTAR ISI	v
DAFTAR TABEL	vii
DAFTAR GAMBAR.....	viii
BAB 1 PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Batasan dan Rumusan Masalah	2
1. Batasan Masalah	2
2. Rumusan Masalah.....	3
BAB 2 TINJAUAN PUSTAKA.....	4
A. Algoritma Genetika	4
B. Software-Defined Radio (SDR).....	7
C. Penjadwalan Komputasi Terdistribusi	9
D. Grafik Tugas Komputasi SDR dengan Modulasi GMSK	10
BAB 3 TUJUAN DAN MANFAAT PENELITIAN.....	13
A. Tujuan Penelitian.....	13
B. Manfaat Penelitian	13
BAB 4 METODE PENELITIAN	15
A. Rencana/Disain Pelaksanaan Penelitian	15
B. Tahapan Penelitian.....	15
C. Hasil/Sasaran Penelitian	19
BAB 5 HASIL DAN PEMBAHASAN	20

A. Persiapan.....	20
B. Pelaksanaan Kegiatan	20
1. Simulasi dan Pengujian Algoritma Genetika.....	20
2. Analisis Perbandingan Metoda Paralelisme Data dan Paralelisme Tugas.....	24
3. Grafik Tugas untuk Penjadwalan Komputasi Parallel/Terdistribusi Menggunakan Algoritma Genetika.....	33
4. Operator Genetika untuk Skema Penjadwalan Komputasi Parallel	34
5. Fungsi Fitness Algoritma Genetik untuk Penjadwalan Komputasi Terdistribusi	42
C. Faktor-Faktor Pendukung	42
D. Faktor-Faktor Penghambat	43
E. Jalan Keluar/Solusi	44
F. Ketercapaian.....	44
BAB 6 KESIMPULAN DAN SARAN	45
A. Kesimpulan.....	45
B. Saran	45
DAFTAR PUSTAKA.....	46
LAMPIRAN	48
A. Surat Kontrak Penelitian.....	48
B. Artike Jurnal	52
C. Buku Catatan Harian Penelitian (Log Book).....	58
D. Seminar Instrumen.....	66
E. Seminar Hasil.....	70

DAFTAR TABEL

Tabel 1. Waktu Eksekusi Total (satuan waktu) Penyelesaian Penjadwalan Komputasi Terdistribusi DP-WS dan TP-GA untuk 2 <i>client</i>	29
Tabel 2. Waktu <i>Idle</i> untuk Menyatakan Pemerataan Beban Kerja Komputasi Terdistribusi DP-WS dan TP-GA	30

DAFTAR GAMBAR

Gambar 2-1. Proses algoritma genetika.....	6
Gambar 2-2. Grafik Tugas Modulator GMSK	11
Gambar 2-3. Grafik Tugas Demodulator GMSK	12
Gambar 4-1. <i>Fishbone</i> Metode Penelitian.....	18
Gambar 5-1. Banyaknya generasi dan total waktu eksekusi GA Standar untuk mencapai solusi (2000 kali pengujian)	23
Gambar 5-2. Banyaknya generasi dan total waktu eksekusi GA <i>Terminated-N</i> untuk mencapai solusi (2000 kali pengujian untuk $N = 500$)	24
Gambar 5-3. Waktu eksekusi dan waktu idle metoda DP-WS (2 paket, 3 client).....	27
Gambar 5-4. Waktu eksekusi dan waktu <i>idle</i> metoda TP-GA (2 paket, 3 <i>client</i>)	28
Gambar 5-5. Analisis <i>Finishing Time</i> (FT) antara Paralelisme Data (<i>Data-Parallelism</i>) dengan Paralelisme Tugas (<i>Task-Parallelism</i>)	31
Gambar 5-6. Analisis beban kerja antara Paralelisme Data (<i>Data-Parallelism</i>) dengan Paralelisme Tugas (<i>Task-Parallelism</i>).....	32
Gambar 5-7. Grafik tugas DAG dengan nilai bobot dan ketinggian tiap node	34
Gambar 5-8. Dua buah string induk yang akan mengalami <i>crossover</i>	36
Gambar 5-9. Letak titik pemotongan tiap-tiap kromosom	36
Gambar 5-10. Hasil <i>crossover</i> dengan nilai <i>fitness</i> yang baru	37
Gambar 5-11. Letak titik <i>crossover</i> dan hasil proses <i>crossover</i>	38

BAB 1

PENDAHULUAN

A. Latar Belakang

Kendala utama dalam mewujudkan sebuah sistem pengolah sinyal digital untuk sinyal frekuensi tinggi (radio, tv dan radar) adalah dalam hal kebutuhan komputasi untuk memenuhi pemrosesan laju bit yang sangat tinggi. Beban komputasi yang sangat besar dalam pengolahan sinyal digital frekuensi tinggi memerlukan waktu komputasi yang cukup lama sehingga menyebabkan waktu tunda (*delay*) dan akan menurunkan laju bit yang dapat dikirimkan. Dengan demikian, permasalahan komputasi akan menurunkan kualitas pengiriman informasi secara real-time. Untuk itu, perlu adanya sebuah metoda dalam hal komputasi pengolahan sinyal digital untuk frekuensi tinggi sehingga meskipun komputasi cukup besar tetap dapat menghasilkan laju bit yang diinginkan untuk komunikasi digital secara *real-time*.

Komputasi pengolahan sinyal digital frekuensi tinggi dapat digunakan dalam mengembangkan *software radio* (*softradio*), *software televisi* (*softtv*), dan *software radar* (*softradar*), atau sistem digital frekuensi tinggi lainnya yang membutuhkan proses real-time. Pengertian *softradio*, *softtv*, dan *softradar* disini adalah sebuah radio/tv/radar yang fungsi kerja sistem tersebut dijalankan oleh perangkat lunak (*software*) untuk meningkatkan fleksibilitas sistem. Perangkat lunak yang dimaksud menggantikan fungsi perangkat keras untuk menjalankan sistem radio/tv/radar konvensional yang tidak fleksibel karena rangkaian dirancang untuk menjalankan sebuah fungsi yang telah didefinisikan sebelumnya. Jika fungsi radio/tv/radar dijalankan menggunakan perangkat lunak, maka ada kemungkinan mengubah algoritma perangkat lunak sehingga fungsi radio/tv/radar dapat berbeda tergantung algoritma perangkat lunak yang dijalankan.

Metoda komputasi terdistribusi dapat digunakan untuk mengatasi permasalahan komputasi yang cukup besar sehingga waktu eksekusi menjadi lebih singkat. Namun, penerapan komputasi terdistribusi untuk komputasi pengolahan sinyal digital frekuensi tinggi belum banyak digunakan. Permasalahan utama dalam proses komputasi terdistribusi adalah penjadwalan tugas-tugas komputasi sehingga menghasilkan penjadwalan yang optimal yaitu waktu total eksekusi lebih singkat serta beban kerja sumber daya komputasi lebih merata.

Pengolahan sinyal digital dengan menerapkan komputasi paralel telah dilakukan oleh Zheng (2007) dengan mengimplementasikan “*weighted-selective scheduling scheme*” untuk penjadwalan komputasi paralel *Software-Defined Radio* (SDR) dalam sebuah jaringan cluster PC, sebagai perbaikan dari skema penjadwalan *round robin*.. Metoda komputasi paralel menggunakan paralelisme data yaitu semua node komputasi melakukan tugas yang sama untuk paket data yang berbeda. Skema penjadwalan *weighted-selective* yang digunakan oleh Zheng akan selalu memprioritaskan node yang memiliki kemampuan komputasi paling tinggi untuk setiap paket data yang akan diproses. Dengan adanya metoda paralelisme data maka ada kemungkinan beberapa node tidak melakukan komputasi sama sekali karena nilai prioritasnya paling rendah. Dengan demikian, beban kerja komputasi terdistribusi menjadi tidak merata.

Penelitian ini memperbaiki kelemahan sistem yang dikembangkan oleh Zheng tersebut di atas, baik dalam hal optimasi penjadwalan maupun optimasi pembagian beban kerja proses komputasi terdistribusi sehingga diperoleh kinerja komputasi tugas-tugas pengolahan sinyal digital frekuensi tinggi akan lebih baik. Metoda yang digunakan dalam skema penjadwalan adalah menggunakan Algoritma Genetika (*genetic algorithm*) dalam melakukan optimasi penjadwalan komputasi terdistribusi pada sistem SDR. Selain itu, metoda dalam pembagian beban kerja pemrosesan paralel menggunakan metoda paralelisme tugas sehingga pendistribusian tugas menjadi lebih fleksibel dan beban tugas menjadi lebih merata untuk semua node komputasi yang terlibat.

Algoritma Genetika sangat ditentukan oleh fungsi *fitness* dalam melakukan optimasi permasalahan penjadwalan, sehingga fungsi *fitness* yang digunakan perlu ditentukan terlebih dahulu. Kriteria yang umum digunakan untuk optimasi penjadwalan adalah waktu penyelesaian tugas-tugas terhadap sebuah jadwal. Sedangkan kriteria yang digunakan untuk melakukan optimasi pembagian beban kerja dapat menggunakan waktu yang tidak digunakan oleh node (waktu *idle* node) yang terlibat dalam sebuah penjadwalan. Dua kriteria ini yang akan digunakan dalam menentukan fungsi *fitness* dari Algoritma Genetika dalam melakukan optimasi.

B. Batasan dan Rumusan Masalah

1. Batasan Masalah

Luasnya permasalahan dalam hal penjadwalan komputasi terdistribusi dengan metoda paralelisme tugas dengan menerapkan Algoritma Genetika, maka batasan

masalah dalam penelitian ini adalah sebagai berikut: (1) sistem penjadwalan yang digunakan dalam penelitian ini adalah off-line scheduling atau penjadwalan statis; (2) tugas-tugas SDR yang diamati terbatas modulasi Gaussian Minimum Shift Keying (GMSK) dan waktu komunikasi antar node komputasi diabaikan; (3) simulasi dan pengujian tugas-tugas komputasi menggunakan Matlab, implementasi algoritma menggunakan bahasa pemrograman JAVA, dan pengujian kinerja berbentuk simulasi untuk membuktikan konsep-konsep yang diajukan.

2. Rumusan Masalah

Pertanyaan penelitian yang akan dijawab dalam penelitian ini adalah bagaimana formula fungsi fitness untuk skema penjadwalan tugas-tugas komputasi SDR? Apakah dengan skema penjadwalan menggunakan algoritma dapat menghasilkan penjadwalan yang optimal dilihat dari waktu eksekusi dan apakah penyeimbangan beban kerja menjadi lebih merata antar tiap node komputasi?

BAB 2

TINJAUAN PUSTAKA

A. Algoritma Genetika

Algoritma Genetika telah terbukti handal untuk mencari solusi permasalahan yang dikategorikan sebagai *NP-hard problem*. Dengan demikian, penjadwalan menggunakan Algoritma Genetika diyakini akan menghasilkan sebuah penjadwalan komputasi paralel atau terdistribusi yang cenderung optimal. Hal ini berdasarkan fakta bahwa Algoritma Genetika sudah terbukti handal dalam setiap penyelesaian permasalahan yang terkait dengan proses optimasi karena Algoritma Genetika berpotensi mendapatkan titik optimum global dalam mencari setiap pelacakan solusi permasalahan.

Kompleksitas waktu Algoritma Genetika yang dinyatakan dalam nilai *big-O* untuk permasalahan yang memiliki skala seragam (*uniformly scaled problem*) adalah $O(m)$, dan $O(m^2)$ adalah kompleksitas waktu Algoritma Genetika untuk permasalahan yang bersifat eksponensial (Lobo, 2000).

Bagian berikut ini akan menjelaskan konsep dasar Algoritma Genetika, implementasi Algoritma Genetika untuk penjadwalan tugas-tugas komputasi terdistribusi dengan menggunakan metoda paralelisme tugas, serta hasil pengujian dari penjadwalan Algoritma Genetika beserta pembahasannya.

Algoritma Genetika atau sering disingkat GA adalah sebuah teknik pelacakan (*searching*) yang digunakan dalam komputasi untuk mendapatkan solusi yang paling optimal. Algoritma Genetika telah dipergunakan secara luas dan telah berhasil untuk berbagai optimasi permasalahan. Algoritma Genetika adalah merupakan teknik-teknik probabilistik mulai dari pembentukan populasi sebagai calon solusi dari permasalahan, dan secara barangsur-angsur diproses menuju ke arah solusi optimal melalui perulangan aplikasi operator genetika seperti *crossover*, mutasi, seleksi dan reproduksi (Rahmani, 2008).

Optimasi permasalahan dapat dilakukan dengan maksimasi fungsi atau minimasi fungsi. Jika optimasi permasalahan adalah minimasi sebuah fungsi f , maka optimasi permasalahan sebenarnya sama dengan maksimasi sebuah fungsi g , dimana fungsi $g = -f$ yaitu:

$$\min f(x) = \max g(x) = \max \{-f(x)\} \quad (1)$$

Fungsi yang digunakan dalam Algoritma Genetika dalam melakukan optimasi sering disebut dengan **fungsi obyektif**, dan fungsi yang dimaksud dapat berupa fungsi variabel tunggal ataupun fungsi multivariabel. Kita boleh mengasumsikan bahwa fungsi obyektif mengambil nilai-nilai positif dari domain permasalahan; selain itu kita juga boleh menambahkan beberapa nilai konstanta positif C pada fungsi tersebut, sehingga:

$$\max g(x) = \max \{g(x) + C\} \quad (2)$$

Dengan demikian, tahap awal dalam implementasi Algoritma Genetika untuk optimasi sebuah permasalahan adalah menentukan fungsi yang akan dilakukan optimasi baik optimasi untuk menemukan nilai minimum (minimasi fungsi) atau untuk menemukan nilai maksimumnya (maksimasi fungsi). Fungsi obyektif tersebut akan menjadi fungsi fitness yang digunakan sebagai dasar evaluasi setiap hasil pelacakan solusi.

Fungsi *fitness* yang didefinisikan dalam representasi genetika akan menentukan ukuran kualitas dari solusi yang diajukan. Fungsi *fitness* ini bersifat unik, yaitu selalu berbeda antara satu permasalahan dengan permasalahan yang lain atau dengan kata lain fungsi fitness sangat tergantung pada permasalahan yang akan dicari solusinya. Jika ekspresi fungsi *fitness* sulit ditentukan, maka solusi permasalahan dapat menggunakan Algoritma Genetika interaktif (*interactive genetic algorithm*).

Penelitian ini memperbaiki kelemahan sistem yang dikembangkan oleh Zheng tersebut di atas, baik dalam hal optimasi penjadwalan maupun optimasi pembagian beban kerja proses komputasi terdistribusi sehingga diperoleh kinerja komputasi tugas-tugas pengolahan sinyal digital frekuensi tinggi dengan mengambil studi kasus komputasi sinyal digital untuk pengembangan SDR yang lebih baik. Metoda yang akan digunakan dalam skema penjadwalan adalah menggunakan algoritma genetika dalam melakukan optimasi penjadwalan komputasi terdistribusi pada sistem SDR. Selain itu, metoda pembagian beban kerja pemrosesan paralel yang akan diteliti menggunakan metoda paralelisme tugas sehingga pendistribusian tugas menjadi lebih fleksibel dan beban tugas menjadi lebih merata untuk semua node komputasi yang terlibat.

Algoritma genetika atau sering disingkat GA adalah sebuah teknik pelacakan (*searching*) yang digunakan dalam komputasi untuk mendapatkan solusi yang paling optimal. Algoritma genetika telah dipergunakan secara luas dan telah berhasil untuk berbagai optimasi permasalahan. Algoritma genetika adalah menerapkan teknik-teknik probabilistik mulai dari pembentukan populasi sebagai calon solusi dari permasalahan, dan

secara barangsur-angsur diproses menuju ke arah solusi optimal melalui perulangan aplikasi operator genetika seperti *crossover*, mutasi, seleksi dan reproduksi (Rahmani, 2008).

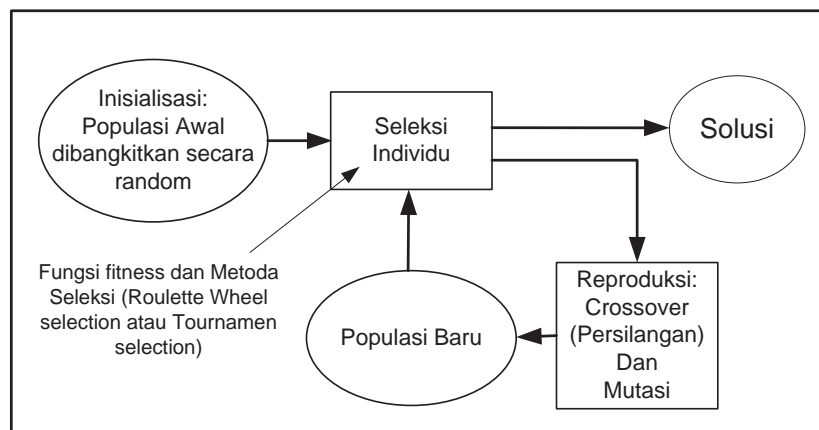
Prinsip kerja algoritma genetika adalah dalam melakukan pelacakan mencari solusi adalah berdasarkan teori Schemata yang menyatakan bahwa:

Short, low-order, above average schemata receive exponentially increasing trials in subsequent generations of a genetic algorithm (Michalewicz, 1992).

Ciri khas algoritma genetika memerlukan dua buah variabel yang harus didefinisikan, yaitu:

- sebuah representasi genetika dari domain solusi
- sebuah fungsi *fitness* untuk mengevaluasi domain solusi.

Representasi standar untuk solusi adalah sebuah array sehingga lebih cocok untuk algoritma genetika karena elemen-elemennya dapat diatur dengan mudah sehingga operasi persilangan (*crossover*) dan operasi mutasi juga dapat dilakukan dengan mudah. Secara umum proses algoritma genetika dalam mencari solusi sebuah permasalahan ditunjukkan pada Gambar 2-1.



Gambar 2-1. Proses algoritma genetika

Berdasarkan Gambar 2-1 tersebut, algoritma genetika diawali dengan inisialisasi populasi yaitu proses membangkitkan sejumlah kromosom untuk membentuk populasi awal. Selanjutnya, populasi awal ini dihitung nilai kekuatan atau nilai fitness tiap-tiap kromosom, dan dilakukan seleksi dimana kromosom memiliki nilai *fitness* yang paling kuat akan memperoleh kesempatan paling tinggi untuk terpilih dalam populasi generasi

selanjutnya. Metoda seleksi dapat menggunakan *Roulette Wheel selection* atau *Tournament selection*. Selain proses seleksi dalam membentuk populasi generasi baru, algoritma genetika juga melakukan proses *crossover* yang terjadiannya ditentukan oleh laju *crossover* (*crossover rate*) serta proses mutasi yang terjadiannya juga ditentukan oleh sebuah nilai yang disebut dengan laju mutasi (*mutation rate*). Setiap diperoleh populasi generasi baru, algoritma genetika akan menguji kekuatan (*fitness*) setiap kromosom dalam populasi tersebut. Proses algoritma genetika akan berhenti jika solusi yang diinginkan telah tercapai atau terjadi proses terminasi dimana terminasi ini dapat ditentukan oleh banyaknya generasi yang dibangkitkan atau nilai fungsi obyektif untuk kriteria tercapainya sebuah solusi telah dipenuhi. *Pseudo-code* algoritma dari algoritma genetika adalah sebagai berikut:

1. Pilih populasi awal
2. Hitung nilai *fitness* tiap-tiap individu dalam populasi
3. **Repeat:**
 - a. Pilih individu yang memiliki rangking terbaik untuk dikembangbiakkan
 - b. Biakkan generasi baru melalui operasi genetika yaitu persilangan dan mutasi dan lahirkan generasi anak
 - c. Hitung *fitness* setiap individu dari generasi anak
 - d. Gantikan rangking terjelek dari populasi dengan generasi anak yang memiliki nilai *fitness* yang lebih baik
4. **Until** terminasi.

Kriteria untuk mengakhiri iterasi algoritma genetika dalam proses mencari solusi dapat menggunakan fungsi obyektif yaitu iterasi akan berhenti jika proses evaluasi telah menghasilkan sebuah nilai tertentu yang memenuhi fungsi obyektif. Selain itu juga dapat menggunakan kriteria berdasarkan jumlah generasi yang telah dibangkitkan.

B. Software-Defined Radio (SDR)

Penelitian tentang komputasi pengolahan sinyal digital frekuensi tinggi khususnya dalam mengembangkan *Software Radio* (softradio), telah dilakukan oleh peneliti-peneliti sebelumnya. *Software-Defined Radio* (SDR) atau ada juga yang menyebut *Software Radio* (SWR) atau radio yang fungsinya didefinisikan menggunakan perangkat lunak diperkenalkan pertama kali pada tahun 1991 oleh Joseph Mitola (Reed, 2002).

Penelitian tentang arsitektur dan proses komputasi yang telah dilakukan oleh peneliti lain sebelumnya antara lain:

Vanu (1999), mengembangkan arsitektur SDR menggunakan komputer pribadi (*Personal Computer*) dan komputasi pengolahan sinyal digital masih menggunakan komputasi sekuensial (Bose, 1999). Corneloup (2002) mengimplementasikan arsitektur SDR pada board DSP SHARC, dengan pendekatan pemrosesan paralel (sistem multiprosesor) namun tidak menjelaskan permasalahan penjadwalannya. Pucker (2002) membandingkan tiga model proses pemilihan saluran yaitu *Digital Down Converter* (DDC), *Frequency Domain Filtering* (FDF) dan *Polyphase FFT Filter Bank* (PFFB) dengan melihat laju kesalahan bit (*bit error rate or BER*). Prosesor yang digunakan adalah FPGA (*Field Programmable Gate Array*). Komputasi yang digunakan dalam proses pemilihan saluran adalah komputasi sekuensial. Harris (2003) melakukan simulasi terhadap pemilihan saluran sistem AMPS, komunikasi radio FM, dan SSB dengan menggunakan bank filter polifase. Proses komputasi yang digunakan juga masih menggunakan komputasi sekuensial. Gweon (2005) mengembangkan arsitektur *software-define radio* (SDR) yang dapat dikonfigurasi ulang dengan menggunakan beberapa prosesor DSP TMS320C6416-600 MHz, sehingga pemrosesan sinyal digital dapat dilakukan secara paralel meskipun belum mengimplementasikan skema penjadwalan. Malm (2005) dan Valentin (2006) mengembangkan lapisan data link layer untuk tesbed radio perangkat lunak menggunakan *Universal Software Radio Pheripheral* (USRP) GNU Radio dengan menguji *latency* dari RTT (*round trip time*) untuk pengiriman paket.

Penelitian-penelitian tentang pengolahan sinyal digital frekuensi tinggi yang telah dilakukan sebelumnya dengan studi kasus sistem *Software-Defined Radio* (SDR) antara lain kajian aspek komputasi SDR (Marpanaji, 2006), arsitektur SDR (Marpanaji, 2006), studi simulasi dan eksperimen modulasi GMSK untuk SDR (Marpanaji, 2006), arsitektur perangkat lunak SDR (Marpanaji, 2007), studi eksperimen modulasi DQPSK untuk SDR (Marpanaji, 2007)(Marpanaji, 2008), pengukuran unjuk kerja modulasi GMSK pada platform SDR (Marpanaji, 2007), studi eksperimen pengukuran unjuk kerja modulasi DBPSK (Marpanaji, 2008), dan dekomposisi tugas-tugas SDR (Marpanaji, 2009). Penelitian-penelitian tersebut merupakan kajian awal dalam mengamati dan mengeksplorasi komputasi pengolahan sinyal digital frekuensi tinggi dalam bentuk

komputasi sekuensial, dan belum mengimplementasikan komputasi paralel ataupun komputasi terdistribusi.

C. Penjadwalan Komputasi Terdistribusi

Isu utama dalam menjalankan komputasi terdistribusi dari hasil dekomposisi tugas adalah penjadwalan (*scheduling*) tugas-tugas tersebut ke dalam masing-masing mesin komputasi (komputer). Penjadwalan dalam sistem komputasi paralel atau komputasi terdistribusi pada prinsipnya adalah mengalokasikan sekumpulan tugas kepada setiap prosesor atau komputer sedemikian rupa sehingga diperoleh kinerja komputasi yang optimal.

Ada dua jenis metoda yang digunakan dalam penjadwalan, yaitu (1) *list heuristics*, membuat daftar tugas berdasarkan prioritas dengan urutan menurun, dan prioritas paling tinggi akan diberikan pertama kali kepada prosesor yang siap melakukan eksekusi; (2) *meta-heuristics*, dikenal juga dengan Algoritma Genetika, melakukan metoda pelacakan secara acak menggunakan sifat evolusi dan genetika alami. Metoda penjadwalan meta-heuristics atau metoda algoritma genetika ini akan menghasilkan sebuah penjadwalan yang optimal (Lee, 2003).

Penjadwalan dalam sistem multiprosesor atau komputasi paralel sebenarnya merupakan sebuah *Non-deterministik Polynomial time* (NP) dan dikategorikan sebagai permasalahan NP yang sulit (*NP_Hard problem*) (Abdeyazdan, 2008). Permasalahan penjadwalan dalam sistem multiprosesor atau komputasi paralel ini sering disebut dengan *Multiprocessor Scheduling Problem* (MSP). (Tsujimura, 1997).

Zheng (2007) mengimplementasikan “*weighted-selective scheduling scheme*” dalam melakukan penjadwalan komputasi paralel SDR dalam sebuah jaringan cluster PC, sebagai perbaikan dari skema penjadwalan round robin khususnya dalam menjalankan fungsi radio pada bagian pemancar. Metoda paralelisme menggunakan paralelisme data yaitu semua node komputasi melakukan tugas yang sama untuk paket data yang berbeda. Persamaan yang digunakan dalam menentukan penjadwalan adalah:

$$l = \arg \max_{k \in \{1, 2, \dots, K\}} s_k(n) / w_k(n) \quad (3)$$

dimana $s_k(n), k = 1, 2, \dots, K$ adalah parameter status dari node pekerja yang dapat diberi tanda dengan angka “0” yang berarti “sibuk” atau angka “1” untuk menyatakan kondisi sedang “bebas” pada interval frame ke- n . Dalam permasalahan ini, sekali

manager distribusi memulai untuk mengirim sebuah satuan frame ke sebuah node pekerja pada interval frame tetap T_s , node pekerja akan diberi tanda “0” dalam tabel status; dan ketika tugas komputasi telah selesai dan umpan balik berhasil dengan baik, node pekerja ini akan diberi tanda “1” lagi dan dapat dipilih untuk diberi pekerjaan lagi. Sedangkan l adalah indeks dari node pekerja dimana frame akan didistribusikan untuk pemrosesan menurut bobot yang menyatakan kapasitas pemrosesan dari tiap node pekerja dalam tabel status. Bobot node pekerja ke- k dalam interval frame ke- n yang ditentukan oleh kemampuan komputasi dari node pekerja dan status pekerjaannya dinyatakan dengan $w_k(n)$ dengan persamaan:

$$w_k(n) = L_k(n) / T_s, k = 1, 2, \dots, K \quad (4)$$

Dimana $L_k(n)$ adalah waktu yang diperlukan CPU untuk melakukan proses pengolahan sinyal digital satu frame sinyal jika node pekerja ke- k menjadi “bebas” selama interval frame ke- n . Sebaliknya, indikator ini akan tetap tidak berubah, yaitu $w_k(n) = w_k(n-1)$. Sedangkan kemampuan rata-rata node pekerja untuk frame ke- n dinyatakan dengan $\bar{w}_k(n)$, dimana $k = 1, 2, \dots, K$.

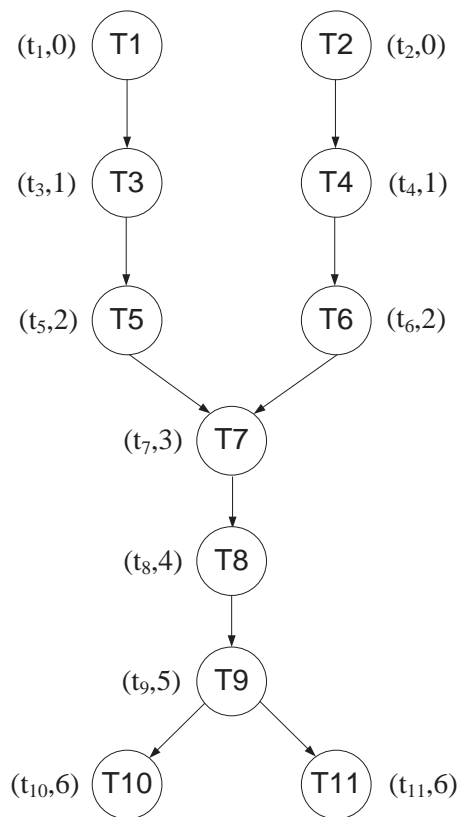
Skema penjadwalan komputasi terdistribusi dapat disusun setelah dekomposisi tugas komputasi pengolahan sinyal digital frekuensi tinggi terbentuk. Studi kasus tentang dekomposisi tugas-tugas komputasi pengolahan sinyal digital frekuensi tinggi sistem SDR untuk sistem pemancar dan penerima dengan sistem modulasi *Gaussian Minimum Shift Keying* (GMSK) telah dilakukan sebelumnya (Marpanaji, 2009). Penelitian ini selain menentukan tugas-tugas komputasi juga mengukur waktu eksekusi tiap-tiap tugas komputasi sehingga dapat digunakan sebagai dasar dalam melakukan penelitian tentang skema penjadwalan komputasi terdistribusi dari pengolahan sinyal digital frekuensi tinggi.

D. Grafik Tugas Komputasi SDR dengan Modulasi GMSK

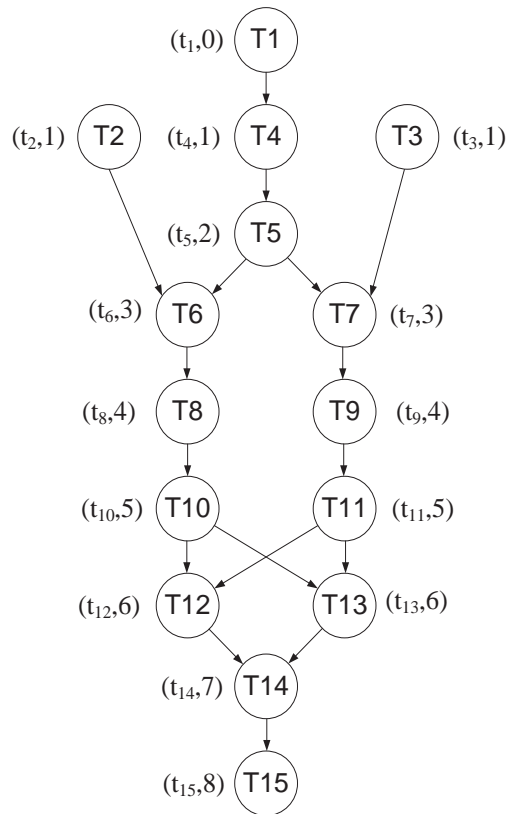
Sebagai studi kasus dalam penelitian ini digunakan grafik tugas komputasi dalam komunikasi SDR dengan skema modulasi GMSK baik untuk pemancar maupun penerima. Grafik tugas tersebut menggambarkan proses-proses yang bersifat berurutan (proses berikutnya akan berlangsung jika proses sebelumnya telah selesai) serta proses-

proses yang bisa dilaksanakan secara terpisah (dua proses atau lebih dapat dijalankan secara bersamaan dan tidak tergantung proses yang lain). Selain itu, grafik tugas juga menertakan waktu eksekusi serta tingkat urutan untuk tiap-tiap proses.

Banyaknya proses yang dapat dikerjakan secara terpisah menunjukkan derajat paralelisme komputasi paralel atau terdistribusi. Semakin banyak proses yang dapat dijalankan secara terpisah berarti semakin tinggi derajat paralelisme sistem komputasi. Grafik tugas untuk sistem pemancar dan penerima sistem SDR dengan skema modulasi GMSK dapat dilihat pada gambar berikut ini.



Gambar 2-2. Grafik Tugas Modulator GMSK



Gambar 2-3. Grafik Tugas Demodulator GMSK

Berdasarkan grafik tugas pemancar dan penerima selanjutnya dilakukan penelitian tentang aplikasi algoritma untuk penjadwalan komputasi terdistribusi sistem SDR dengan metode paralelisme tugas.

Tahap awal kegiatan penelitian ini adalah analisis kebutuhan dan studi pustaka tentang komputasi paralel, komputasi terdistribusi, Algoritma Genetika, dan penjadwalan komputasi paralel atau terdistribusi. Dasar teori yang diperoleh sangat diperlukan dalam mengembangkan sistem penjadwalan komputasi terdistribusi menggunakan Algoritma Genetika.

Skema penjadwalan *weighted-selective* yang digunakan oleh Zheng untuk komputasi paralel SDR merupakan metoda penjadwalan *list heuristic* dengan metoda paralelisme data. Sedangkan penelitian ini menggunakan metoda penjadwalan yang kedua yaitu *meta-heuristic* atau Algoritma Genetika dengan mengimplementasikan paralelisme tugas.

BAB 3

TUJUAN DAN MANFAAT PENELITIAN

A. Tujuan Penelitian

Tujuan penelitian ini adalah menghasilkan sebuah konsep dan prototipe penjadwalan komputasi terdistribusi dengan metoda paralelisme tugas dan mengimplementasikan Algoritma Genetika untuk optimasi penjadwalan tugas-tugas komputasi SDR. Konsep yang dihasilkan ini dapat memperbaiki kelemahan skema penjadwalan *weighted-selective* dengan mengimplementasikan metoda paralelisme data.

B. Manfaat Penelitian

Manfaat penelitian ini menghasilkan kontribusi keilmuan dibidang pengolahan sinyal digital, telekomunikasi radio, komunikasi data, dan pengembangan software radio/tv/radar, terutama dalam hal:

- Pembangunan grafik tugas atau DAG (*Directed Acyclic Graph*) untuk tugas-tugas SDR dalam menjalankan fungsi-fungsi radio.
- Usulan paralelisme berdasarkan tugas untuk komputasi terdistribusi pada SDR.
- Peningkatan kecepatan komputasi dan pemerataan beban kerja komputasi terdistribusi dengan skema penjadwalan menggunakan Algoritma Genetika.
- Usulan fungsi fitness Algoritma Genetika untuk komputasi terdistribusi dengan metoda paralelisma tugas .

Hasil penelitian ini diharapkan dapat menjadikan konsep dasar dalam mengembangkan komputasi terdistribusi pengolahan sinyal digital frekuensi tinggi yang memiliki lajut bit sangat tinggi sehingga dapat digunakan untuk pengembangan sofradio, softtv ataupun sofradar. Luaran penelitian secara umum adalah konsep dan metode penjadwalan komputasi terdistribusi dengan paralelisme tugas dengan menggunakan Algoritma Genetika untuk memilih jadwal komputasi yang optimal. Selain itu, penelitian ini menghasilkan sebuah prototipe komputasi terdistribusi khususnya dalam pengembangan SDR.

Penelitian ini berdasarkan penelitian-penelitian awal sebelumnya termasuk eksplorasi grafik tugas komputasi SDR dengan menggunakan skema modulasi Gaussian Minimum Shift Keying (GMSK) baik untuk sistem pengirim (pemancar) maupun sistem penerima. Grafik tugas yang dihasilkan tersebut digunakan sebagai bahan simulasi dalam mengembangkan aplikasi Algoritma Genetika dalam melakukan optimasi penjadwalan

komputasi terdistribusi dengan metode paralelisme tugas. Kegiatan penelitian dalam mengembangkan prototipe aplikasi penjadwalan komputasi terdistribusi menggunakan Algoritma Genetika dengan metode paralelisme tugas dilaksanakan dalam penelitian ini.

BAB 4

METODE PENELITIAN

A. Rencana/Disain Pelaksanaan Penelitian

Metoda yang digunakan dalam mengatasi permasalahan tentang penjadwalan komputasi terdistribusi menggunakan Algoritma Genetika dengan metoda paralelisme tugas (*task-parallelism*) diawali dengan mengidentifikasi tugas-tugas komputasi SDR untuk berbagai jenis modulasi, melakukan dekomposisi tugas-tugas tersebut sehingga dapat digambarkan dalam bentuk grafik tugas, menghitung nilai ketinggian masing-masing tugas dan menguji waktu eksekusi masing-masing tugas, sehingga dihasilkan sebuah grafik tugas SDR. Berdasarkan grafik tugas yang dihasilkan tersebut kemudian dilakukan penjadwalan dan dilanjutkan dengan optimasi penjadwalan menggunakan Algoritma Genetika. Operator genetika yang digunakan disesuaikan dengan konstrain proses komputasi terdistribusi.

Kinerja penjadwalan menggunakan Algoritma Genetika untuk komputasi terdistribusi dengan metoda paralelisme tugas diamati berdasarkan waktu eksekusi total setiap penjadwalan serta waktu istirahat (*idle*) maksimum dari semua node komputasi yang terjadwal. Waktu istirahat dari sebuah jadwal digunakan sebagai indikator pemerataan beban kerja. Batasan penelitian ini adalah tugas-tugas SDR yang diamati terbatas modulasi GMSK dengan dekomposisi tugas yang telah dilakukan sebelumnya (Marpanaji, 2009) dan waktu komunikasi antar node komputasi diabaikan.

B. Tahapan Penelitian

Penelitian ini dilaksanakan dalam dua tahap, yaitu: (1) Tahap I untuk tahun pertama; dan (2) Tahap II untuk tahun kedua. Rangkaian kegiatan penelitian disesuaikan dengan besarnya dana, waktu, dan sumber daya lainnya sehingga target penelitian secara umum dapat dibagi menjadi beberapa target yang berurutan sehingga urutan kegiatan akan menghasilkan luaran sesuai dengan target akhir adalah tujuan penelitian ini. Tahap I dilaksanakan pada Tahun I (2013) dan Tahap II dilaksanakan pada Tahun II (2014). Suatu hal yang sangat penting dalam pelaksanaan ini adalah perubahan skema penelitian antara Tahun I dan Tahun II terjadi sedikit pergeseran kegiatan mengingat biaya yang disetujui dalam penelitian ini hanya 53,3% dari dana yang diusulkan dalam proposal. Pergeseran kegiatan yang dimaksud adalah pergeseran sebagian kegiatan penelitian yang semula dilaksanakan pada tahun 1 di ubah dilaksanakan pada tahun 2.

Berdasarkan Surat Perjanjian Internal Pelaksanaan Penelitian Fundamental **Nomor: 231/Fund.-BOPTN/UN34.21/2014**, dana penelitian yang disetujui adalah Rp. 40.000.000,00 dari usulan dana sebesar Rp. 75.000.000,00 atau sebesar 53,3% dari dana yang diusulkan. Hal ini memberikan dampak yang cukup signifikan terhadap target kegiatan penelitian pada Tahap II (Tahun 2014). Keterbatasan dana dan waktu tersebut menyebabkan sebagian rencana kegiatan yang tercantum dalam usulan penelitian ada beberapa perubahan. Urutan kegiatan untuk Tahap I (Tahun 2013) dan Tahap II (Tahun 2014) disajikan dalam bentuk *fishbone* seperti ditunjukkan pada Gambar 4-1.

Berdasarkan Gambar 4-1 tersebut, maka rencana kegiatan penelitian yang harus dilaksanakan pada Tahap I (Tahun 2013) dan Tahap II (Tahun 2014) beserta luaran yang dihasilkan dari kegiatan penelitian tersebut adalah sebagai berikut.

Tahap I (Tahun 2013):

1. Studi Pustaka dan Analisis Kebutuhan

Studi pustaka merupakan kegiatan awal dan selalu dilakukan dalam setiap kegiatan (berkelanjutan). Studi pustaka ini dilakukan untuk mempertajam pengetahuan dan konsep baik Algoritma Genetika secara umum, pengolahan sinyal digital frekuensi tinggi, *Software-Defined Radio* (SDR). Selain itu, studi pustaka juga diperlukan untuk mempertajam pengetahuan dan konsep tentang algoritma-algoritma pemrograman yang terkait dengan Algoritma Genetika sebagai dasar implementasi (*coding*) dalam bahasa pemrograman. Kegiatan ini juga sekaligus melaksanakan analisis kebutuhan berdasarkan konsep yang diperoleh dari kajian pustaka.

2. Eksplorasi Algoritma Genetika

Kegiatan ini berupa simulasi dan pengujian Algoritma Genetika sebagai persiapan awal aplikasi Algoritma Genetika dalam penjadwalan. Kegiatan ini bertujuan untuk memperoleh metoda dan nilai parameter dalam setiap komponen Algoritma Genetika sehingga diperoleh metoda dan parameter yang tepat untuk keperluan optimasi penjadwalan komputasi.

3. Analisis Komputasi Parallel

Kegiatan pada tahap ini adalah melakukan analisis komputasi parallel Data-Parallelism and Weighted-Scheduling (DP-WS) dengan Task-Parallelism with Genetic Algorithm (TP-GA). Analisis dilakukan terhadap waktu total eksekusi setiap proses yang dijadwalkan serta efektifitas dalam pembagian beban kerja untuk seluruh

mesin yang tergabung dalam sistem komputasi parallel atau komputasi terdistribusi tersebut.

4. Publikasi

Hasil-hasil penelitian untuk tiap kegiatan dijadikan bahan untuk penulisan makalah seminar, artikel jurnal, dan laporan akhir untuk mempublikasikan temuan-temuan yang dapat dimanfaatkan peneliti lain yang bekerja dalam bidang yang sama.

Hasil kegiatan penelitian Tahap I (Tahun 2013) digunakan sebagai dasar pengembangan Tahap II (2014). Berdasarkan Gambar 4-1, kegiatan penelitian untuk Tahap II (2014) adalah sebagai berikut.

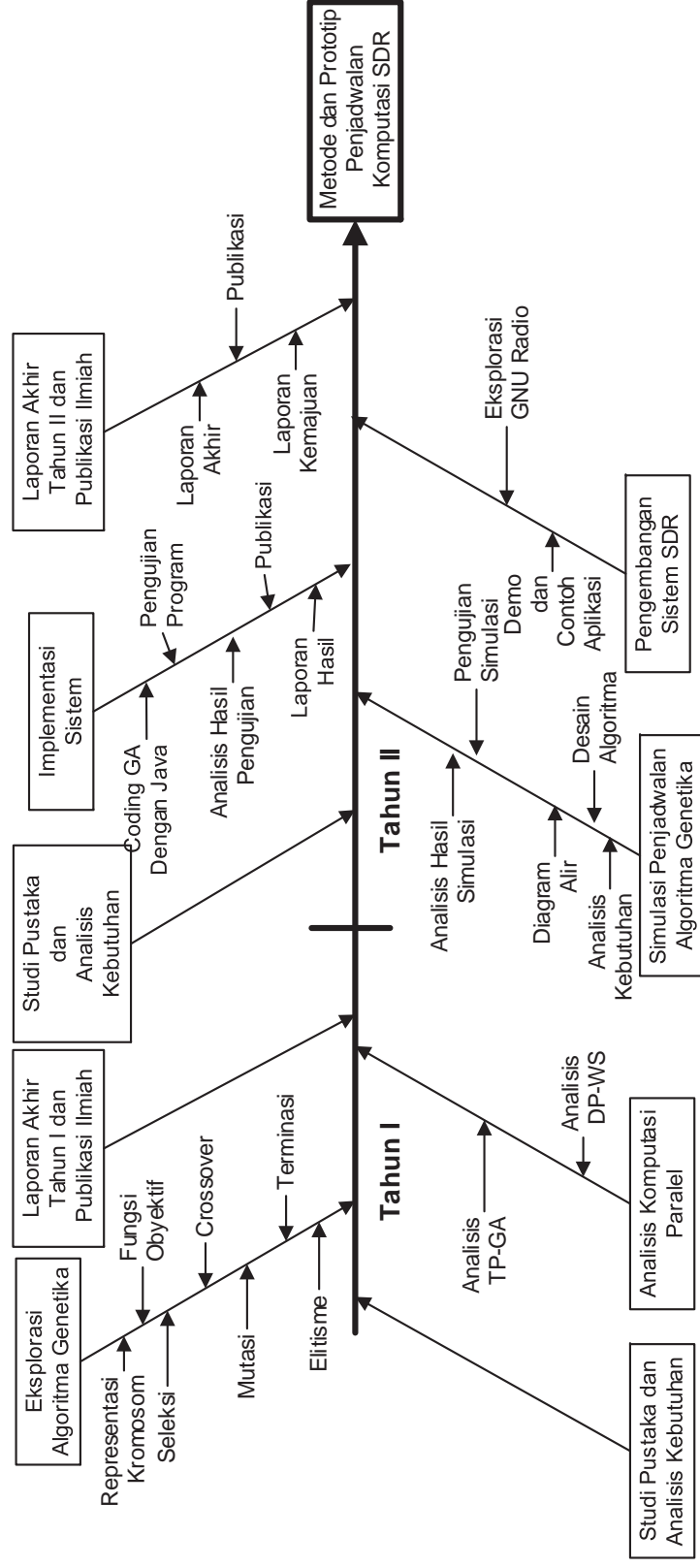
Tahap II (Tahun 2014)

1. Studi Pustaka dan Analisis Kebutuhan

Studi pustaka merupakan kegiatan awal pada Tahun II dan selalu dilakukan dalam setiap kegiatan (berkelanjutan). Studi pustaka ini dilakukan untuk mempertajam pengetahuan dan konsep penjadwalan komputasi secara umum, penjadwalan komputasi parallel dan terdistribusi, aplikasi Algoritma Genetika dalam penjadwalan komputasi, pengolahan sinyal digital frekuensi tinggi, *Software-Defined Radio* (SDR). Selain itu, studi pustaka juga diperlukan untuk mempertajam pengetahuan dan konsep tentang algoritma-algoritma pemrograman yang terkait dengan aplikasi Algoritma Genetika dalam penjadwalan komputasi terdistribusi dengan menggunakan metoda paralelisme tugas untuk pengolahan sinyal frekuensi tinggi (studi kasus komputasi SDR). Kegiatan ini juga termasuk analisis kebutuhan berdasarkan hasil kajian pustaka yang telah dilakukan.

2. Simulasi Penjadwalan Algoritma Genetika

Kegiatan ini untuk mengkaji aplikasi Algoritma Genetika untuk penjadwalan komputasi terdistribusi dengan metoda paralelisme tugas untuk komputasi pengolahan sinyal frekuensi tinggi dalam menjalankan fungsi SDR. Kegiatan ini diawali dengan kajian secara teoritis perbandingan antara sistem penjadwalan *weighted-scheduling* dengan metoda paralelisme data untuk SDR yang dilakukan oleh Zheng dengan sistem penjadwalan menggunakan Algoritma Genetika dengan metoda paralelisme tugas. Berdasarkan kajian tersebut ditentukan algoritma-algoritma penjadwalan komputasi dengan metoda paralelisme tugas sehingga mempermudah implementasi dalam bahasa pemrograman.



Gambar 4-1. Fishbone Metode Penelitian

3. Implementasi Sistem

Implementasi Sistem yang dimaksud dalam tahap ini adalah pembuatan prototipe program (*coding*) Algoritma Genetika menggunakan bahasa pemrograman Java. Kegiatan ini juga sekaligus melakukan pengujian dan simulasi untuk meyakinkan program dapat menjalankan fungsi Algoritma Genetika dengan benar.

4. Pengembangan Sistem SDR

Kegiatan ini merupakan kegiatan penelitian-penelitian selanjutnya dan lebih bersifat peningkatan pengetahuan berdasarkan perkembangan terkini tentang SDR. Tujuan kegiatan ini adalah agar tidak ketinggalan dalam pengembangan SDR saat ini, baik dari sisi perangkat keras (platform komputasi) maupun perangkat lunaknya. Selain itu, kegiatan ini sebagai sarana kajian model alternatif SDR untuk pengembangan berikutnya baik dalam bentuk algoritma, perangkat lunak, maupun perangkat keras yang dapat digunakan dalam pengembangan SDR.

C. Hasil/Sasaran Penelitian

Berdasarkan kondisi pendanaan yang disetujui pada Tahun II ini maka luaran penelitian untuk tahap kedua (Tahun II) adalah sebagai berikut:

- Analisis perbandingan komputasi terdistribusi antara komputasi dengan metoda paralelisme data dengan paralelisme tugas.
- Simulasi algoritma pemrograman untuk penjadwalan Algoritma Genetika untuk komputasi parallel menggunakan Matlab.
- Prototipe fungsi-fungsi Algoritma Genetika menggunakan bahasa pemrograman Java.
- Makalah seminar atau artikel jurnal.

BAB 5

HASIL DAN PEMBAHASAN

A. Persiapan

Persiapan meliputi persiapan peralatan (tools) baik hardware maupun software, sumber daya manusia, ruangan, dan fasilitas lain yang sangat diperlukan untuk kegiatan penelitian.

B. Pelaksanaan Kegiatan

Berdasarkan tahapan dan metode penelitian yang dijelaskan sebelumnya, kegiatan penelitian Tahun I (2013) dan Tahun II (2014) sebagian besar adalah kajian Algoritma Genetika dan aplikasinya dalam sistem penjadwalan tugas-tugas komputasi. Kegiatan diawali dengan studi pustaka dan analisis kebutuhan.

Berdasarkan analisis kebutuhan, kajian Algoritma Genetika memerlukan beberapa bahan dan alat sebagai berikut: (1) Grafik tugas komputasi SDR untuk melakukan simulasi dan uji coba; (2) Fungsi-fungsi serta kelengkapan lain untuk menjalankan Algoritma Genetika seperti representasi kromosom, fungsi obyektif untuk menentukan fitness, fungsi inisialisasi, fungsi perhitungan nilai *fitness*, fungsi seleksi, fungsi *crossover*, fungsi terminasi, dan fungsi elitisme (jika diperlukan); (3) analisis tentang perbandingan penjadwalan komputasi paralel dengan metode paralelisme data dan metode paralelisme tugas; (4) pengujian algoritma dalam membentuk perangkat lunak aplikasi Algoritma Genetika untuk penjadwalan komputasi terdistribusi dengan metode paralelisme tugas meskipun dalam bentuk simulasi dan parsial.

1. Simulasi dan Pengujian Algoritma Genetika

Sebelum dilakukan simulasi Algoritma Genetika dalam rangka meyakinkan konsep dan algoritma pemrograman dalam mengembangkan penjadwalan komputasi terdistribusi menggunakan Algoritma Genetika dengan metoda paralelisme tugas, dilakukan representasi kromosom dan simulasi tiap-tiap komponen Algoritma Genetika. Ciri khas Algoritma Genetika memerlukan dua buah variabel yang harus didefinisikan, yaitu:

- sebuah representasi genetika dari domain solusi
- sebuah fungsi fitness untuk mengevaluasi domain solusi.

Representasi standar untuk solusi adalah sebuah array dari bit atau bentuk array lainnya dengan prinsip yang sama. Sifat utama representasi genetika menggunakan jenis array menjadi lebih cocok untuk Algoritma Genetika karena elemen-elemennya dapat diatur dengan mudah sehingga operasi persilangan (*crossover*) dan operasi mutasi dapat dilakukan dengan mudah. Secara umum proses Algoritma Genetika dalam mencari solusi sebuah permasalahan seperti ditunjukkan pada Gambar 1 pada Bab 2.

Berdasarkan gambar tersebut, Algoritma Genetika terdiri beberapa komponen proses: (1) proses inisialisasi populasi; (2) seleksi individu dengan menghitung nilai *fitness* setiap individu; (3) proses *crossover* jika syarat terjadi *crossover* terpenuhi; (4) proses mutasi jika syarat mutasi terpenuhi; (5) kembali ke proses seleksi individu dari populasi yang baru hasil dari *crossover* dan mutasi atau terminasi setelah hasil optimasi (solusi) telah diperoleh. Setiap komponen proses dalam Algoritma Genetika ditentukan algoritma pemrogramannya dan dilakukan implementasi (*coding*) beserta pengujiannya.

Algoritma Genetika diawali dengan inisialisasi populasi yaitu proses membangkitkan sejumlah kromosom untuk membentuk populasi awal. Selanjutnya, populasi awal ini dihitung nilai kekuatan atau nilai *fitness* tiap-tiap kromosom, dan dilakukan seleksi dimana kromosom dengan nilai *fitness* paling kuat akan memperoleh kesempatan paling tinggi untuk terpilih dalam populasi generasi selanjutnya. Metoda seleksi dapat menggunakan *Roulette Wheel selection* atau *Tournament selection*. Selain proses seleksi dalam membentuk populasi generasi baru, Algoritma Genetika juga melakukan proses *crossover* yang kejadiannya ditentukan oleh laju *crossover* (*crossover rate*) serta proses mutasi yang kejadiannya juga ditentukan oleh sebuah nilai yang disebut dengan laju mutasi (*mutation rate*). Setiap diperoleh populasi generasi baru, Algoritma Genetika akan menguji kekuatan (*fitness*) setiap kromosom dalam populasi tersebut. Proses Algoritma Genetika akan berhenti jika solusi yang diinginkan telah tercapai atau terjadi proses terminasi dimana terminasi ini dapat ditentukan oleh banyaknya generasi yang dibangkitkan atau nilai fungsi obyektif untuk kriteria tercapainya sebuah solusi telah dipenuhi. *Pseudo-code* algoritma dari Algoritma Genetika adalah sebagai berikut:

1. Pilih populasi awal
2. Hitung nilai *fitness* tiap-tiap individu dalam populasi
3. **Repeat:**
 - a. Pilih individu yang memiliki rangking terbaik untuk dikembangbiakkan

- b. Biakkan generasi baru melalui operasi genetika yaitu persilangan dan mutasi dan lahirkan generasi anak
- c. Hitung *fitness* setiap individu dari generasi anak
- d. Gantikan ranking terjelek dari populasi dengan generasi anak yang memiliki nilai *fitness* yang lebih baik

4. **Until** terminasi.

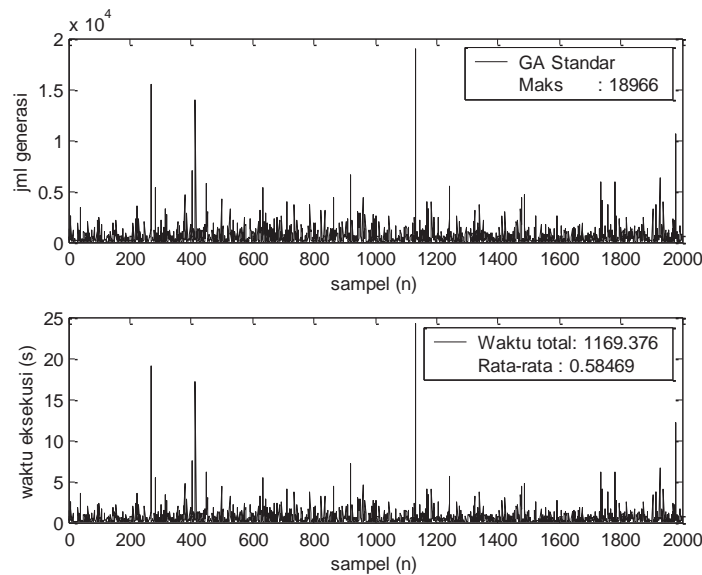
Kriteria untuk mengakhiri iterasi Algoritma Genetika dalam proses mencari solusi dapat menggunakan fungsi obyektif yaitu iterasi akan berhenti jika proses evaluasi telah menghasilkan sebuah nilai tertentu yang memenuhi fungsi obyektif. Selain itu juga dapat menggunakan kriteria berdasarkan jumlah generasi yang telah dibangkitkan. Gambar berikut ini adalah contoh pengamatan waktu eksekusi dan banyaknya generasi yang dihasilkan oleh Algoritma Genetika dalam mencari solusi.

Pengamatan tentang Algoritma Genetika dalam menjalankan proses pencarian sebuah solusi dilakukan dengan perulangan sebanyak 2000 kali. Ada dua metoda Algoritma Genetika yang diamati. Pertama, Algoritma Genetika melakukan proses terminasi berdasarkan kriteria bahwa nilai fungsi obyektif telah tercapai. Metoda ini disebut dengan metoda Algoritma Genetika Standar atau diberi nama GA Standar atau GA-Std. Kedua, Algoritma Genetika melakukan proses terminasi berdasarkan nilai fungsi obyektif dan apabila dalam sejumlah generasi tertentu solusi belum tercapai maka Algoritma Genetika melakukan penyegaran populasi awal sehingga membuka peluang baru dalam memperoleh solusi. Metoda Algoritma Genetika yang kedua ini dinamakan *GA Terminate-N*, dimana N adalah banyaknya generasi.

Hasil dari pengamatan kedua jenis metoda Algoritma Genetika dalam melakukan proses terminasi dapat dilihat pada gambar berikut ini.

Gambar 5-1 tersebut menunjukkan jumlah generasi yang dibangkitkan dan waktu iterasi pada saat GA Standar melakukan iterasi dalam mencari solusi. Nampak bahwa kadang-kadang Algoritma Genetika dalam melakukan eksplorasi titik-titik solusi dan eksploitasi sebuah populasi tidak segera menemukan solusi yang diinginkan atau kriteria terminasi menggunakan fungsi obyektif belum tercapai meskipun jumlah generasi yang dibangkitkan sudah cukup besar. Jumlah generasi maksimum dibangkitkan adalah 18.966 generasi dalam memperoleh solusi yang diinginkan. Hal ini juga akan menyebabkan waktu iterasi Algoritma Genetika menjadi sangat lama dalam memperoleh solusi yaitu

hampir 25 detik untuk jumlah generasi yang paling tinggi, dan waktu total dalam pengujian untuk 2.000 kali perulangan ini 1.169,376 detik. Waktu yang cukup lama ini sangat tidak diinginkan jika Algoritma Genetika akan digunakan dalam skema penjadwalan komputasi terdistribusi saat menerapkan metoda penjadwalan dinamis untuk keperluan komputasi sistem waktu nyata. Berdasarkan hasil pengujian tersebut, maka GA Standar (GA-Std) dalam memperoleh solusi memiliki rata-rata 18.966 generasi dan waktu iterasi rata 0.58469 detik.

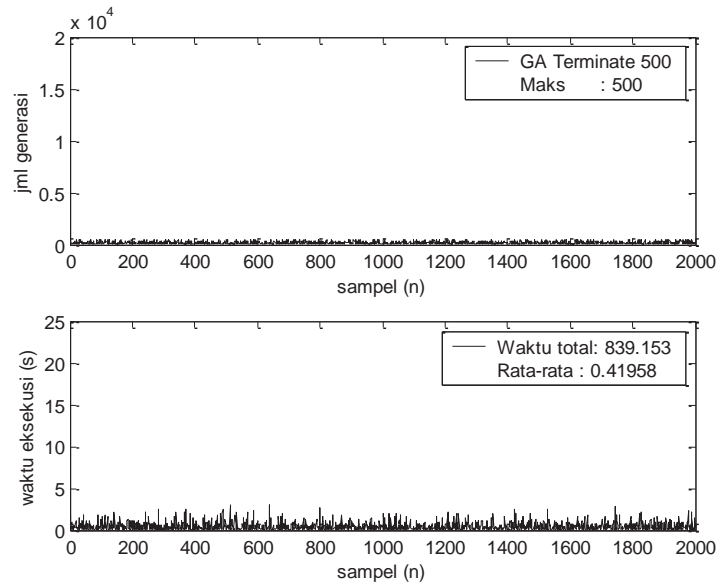


Gambar 5-1. Banyaknya generasi dan total waktu eksekusi GA Standar untuk mencapai solusi (2000 kali pengujian)

Gambar 5-2 menunjukkan jumlah generasi dan waktu iterasi GA Terminated-N dengan $N = 500$. Artinya Algoritma Genetika dibatasi 500 generasi dalam melakukan eksploitasi populasi. Jika jumlah generasi yang dibangkitkan sudah mencapai 500 dan solusi belum diperoleh, maka Algoritma Genetika melakukan inisialisasi ulang populasi awal sehingga memberikan kemungkinan baru dalam mencapai solusi yang diinginkan berdasarkan kriteria nilai fungsi obyektif. Berdasarkan hasil pengujian tentang GA *Terminated-500* ternyata waktu iterasi rata-rata menjadi lebih rendah yaitu 0,41958 dengan waktu total dalam pengujian untuk 2.000 kali perulangan hanya membutuhkan waktu 839,153 detik. Jumlah generasi maksimum tentu saja terbatas 500 generasi sesuai

dengan nilai N yang diberikan. Angka waktu rata-rata iterasi dan waktu total hasil pengujian GA *Terminated*-500 ini lebih rendah dibanding dengan hasil pengujian GA Standar yang ditunjukkan oleh Gambar 5-1.

Berdasarkan uraian hasil pengujian antara GA Standar dengan GA *Terminated*-N ini, maka untuk menghindari Algoritma Genetika terjebak dalam melakukan eksplorasi solusi dan eksploitasi populasi sebaiknya dilakukan pembatasan jumlah generasi yang dibangkitkan jika solusi yang diinginkan belum tercapai. Dengan demikian kriteria terminasi tidak hanya menggunakan fungsi obyektif tetapi juga jumlah generasi yang telah dihasilkan oleh Algoritma Genetika saat melakukan pelacakan solusi.



Gambar 5-2. Banyaknya generasi dan total waktu eksekusi GA *Terminated*-N untuk mencapai solusi (2000 kali pengujian untuk N = 500)

Dengan menggunakan GA *Terminated*-N ini, jika sejumlah generasi N sudah terbentuk namun fungsi obyektif untuk syarat terminasi belum tercapai, maka Algoritma Genetika diperbolehkan melakukan inisialisasi populasi ulang sehingga mengalami penyegaran dalam melakukan eksplorasi dan eksploitasi. Metoda ini akan menjadi perhatian pada saat membuat skema penjadwalan menggunakan Algoritma Genetika.

2. Analisis Perbandingan Metoda Paralelisme Data dan Paralelisme Tugas

Dalam membangun sebuah sistem komputasi paralel atau terdistribusi, ada beberapa model algoritma yang dapat digunakan. Pertama, model data paralel (*data-*

parallel model) yaitu tugas komputasi dipetakan ke seluruh node komputasi dan setiap node komputasi menjalankan fungsi yang sama untuk data yang berbeda. Model ini sering disebut dengan paralelisme data. Kedua, model grafik tugas yaitu tugas-tugas komputasi digambarkan dalam bentuk grafik tugas yang menyatakan hubungan ketergantungan antar tugas, dan tiap-tiap tugas tersebut disebar ke node-node komputasi untuk melakukan proses komputasi terhadap sekelompok data yang sama. Model ini dikenal dengan paralelisme tugas. Ada beberapa model algoritma lainnya yaitu *work pool model*, *master-slave model*, *pipeline* atau *producer-consumer model*, dan *hybrid model*. Penelitian ini hanya membahas dua model algoritma yang pertama yaitu paralelisme data dan paralelisme tugas.

Zheng (2007) menggunakan paralelisme data dalam melakukan komputasi terdistribusi dengan skema penjadwalan *weighted-selective scheduling* terhadap tugas-tugas komputasi dari SDR. Skema penjadwalan ini memiliki kelebihan dalam hal pembagian beban kerja pada saat node-node komputasi memiliki kapasitas komputasi yang berbeda (*heterogen*), jika dibandingkan dengan skema penjadwalan dengan algoritma *Round Robin* yang hanya cocok untuk lingkungan komputasi *homogen*. Namun demikian, dengan digunakannya paralelisme data dalam sistem komputasi terdistribusi, skema penjadwalan *weighted-selective* masih memiliki kelemahan saat perbedaan kapasitas komputasi sangat menyolok antar node-node komputasi.

Jika perbedaan kapasitas komputasi dari node-node komputasi sangat besar, maka ada kemungkinan node komputasi yang memiliki kapasitas komputasi paling rendah tidak mendapatkan tugas komputasi sama sekali. Hal ini disebabkan tugas-tugas komputasi telah diberikan kepada node-node yang memiliki kapasitas komputasi yang jauh lebih tinggi dan paralelisme data yang digunakan tidak memberikan kesempatan dalam pembagian tugas secara merata karena setiap node melakukan komputasi yang sama dan menyeluruh untuk data yang berbeda. Dengan demikian model paralelisme data yang digunakan tidak menjamin pemerataan beban kerja.

Untuk mengatasi kelemahan paralelisme data yang digunakan dalam komputasi terdistribusi menggunakan skema penjadwalan *weighted-selective* dari Zheng tersebut, penelitian ini mengusulkan sistem komputasi terdistribusi dengan sistem paralelisme tugas dengan skema penjadwalan menggunakan algoritma genetika. Dengan menggunakan paralelisme tugas maka penjadwalan diberi kesempatan untuk mengatur

jadwal secara optimal dengan pembagian tugas keseluruhan node-node komputasi yang ada berdasarkan kapasitasnya masing-masing. Setiap node akan melakukan tugas komputasi yang berbeda untuk menyelesaikan pemrosesan sebuah data. Fungsi algoritma genetika adalah melakukan optimasi penjadwalan untuk memperoleh waktu eksekusi yang paling kecil dengan pembagian beban kerja sedapat mungkin lebih merata dengan ditunjukkan oleh waktu *idle* tiap-tiap node komputasi sekecil mungkin.

Analisis waktu penyelesaian tugas (*finishing time*) dan pembagian beban kerja yang ditunjukkan oleh persentase waktu idle terhadap waktu penyelesaian tugas yang telah dilakukan dalam penelitian ini antara paralelisme data yang digunakan oleh Zheng dengan paralelisme tugas dengan menggunakan algoritma genetika dapat dilihat pada analisis berikut ini.

Misalkan tugas komputasi yang akan diselesaikan adalah:

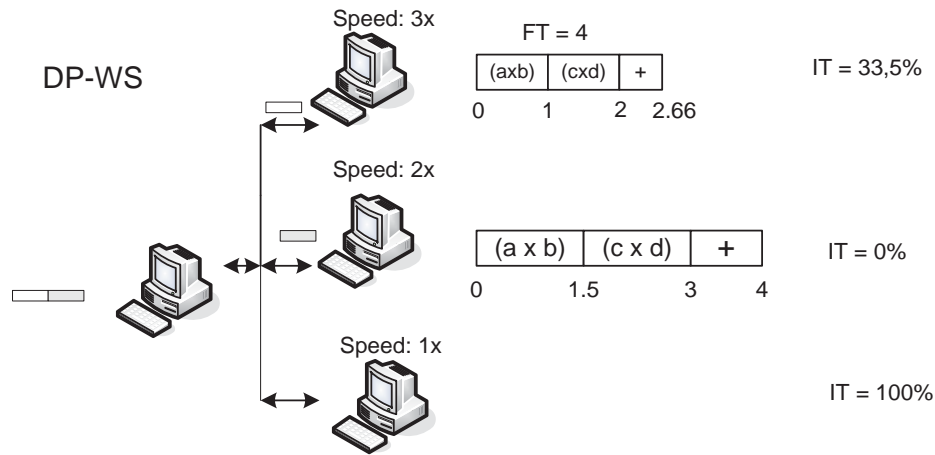
$$(a \times b) + (c \times d) \quad (5)$$

Diketahui waktu eksekusi untuk perkalian (\times) adalah 3 satuan waktu, sedangkan waktu eksekusi penjumlahan (+) adalah 2 satuan waktu. Berikut ini perbandingan waktu eksekusi total hasil penjadwalan antara komputasi terdistribusi menggunakan paralelisme data dengan penjadwalan *weighted-selective* (DP-WS) dengan komputasi terdistribusi menggunakan paralelisme tugas dengan algoritma genetika (TP-GA).

Seandainya ada 2 buah paket data dan sistem komputasi terdistribusi terdiri dari 3 buah *client*, dimana *client-1* memiliki kemampuan komputasi sebesar 1x, *client-2* memiliki kemampuan komputasi 2x, dan *client-3* memiliki kemampuan komputasi 3x, maka waktu total proses komputasi untuk masing-masing proses komputasi terdistribusi dapat dijelaskan sebagai berikut.

Gambar 5-3 menunjukkan proses komputasi terdistribusi DP-WS beserta perhitungan waktu total eksekusi untuk masing-masing *client* serta waktu idle untuk masing-masing *client*. Sifat penjadwalan WS (*weighted-scheduling*) adalah selalu memprioritaskan node yang memiliki kemampuan komputasi yang paling tinggi, sehingga untuk paket data pertama proses komputasi diserahkan ke *client-3* yang memiliki kecepatan 3x. Dengan adanya metoda DP maka *client-3* akan melakukan seluruh proses komputasi (perkalian dan penjumlahan) untuk paket data pertama.

Berdasarkan metoda DP-WS tersebut, maka waktu eksekusi total proses komputasi pada client-3 adalah 2,66 satuan waktu. Sedangkan untuk paket yang kedua, karena client-3 sedang mengerjakan komputasi paket data pertama, maka WS akan menyerahkan proses komputasi paket data kedua ini kepada client-2 karena prioritas komputasinya masih lebih tinggi dibanding client-1. Dengan adanya metoda DP maka client-2 akan mengerjakan seluruh proses komputasi untuk paket data kedua (perkalian dan penjumlahan) juga, sehingga waktu eksekusi total penyelesaian jadwal adalah 4 satuan waktu.



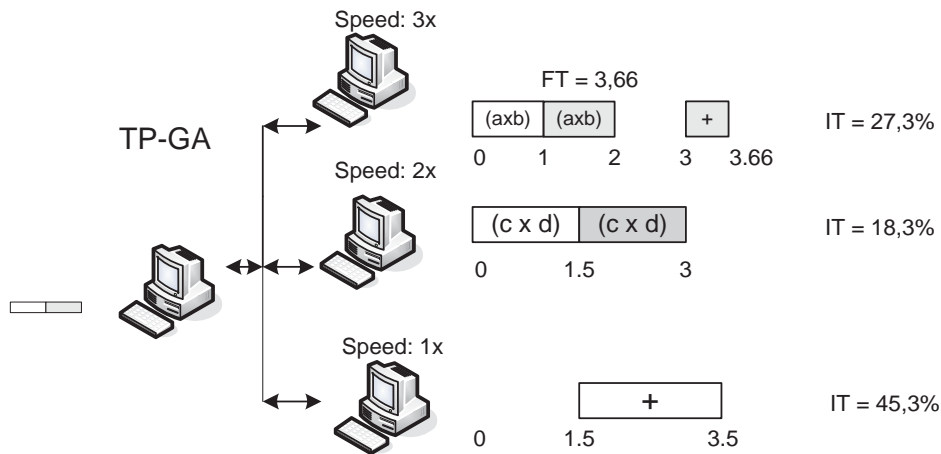
Gambar 5-3. Waktu eksekusi dan waktu idle metoda DP-WS (2 paket, 3 client)

Jika proses komputasi berhenti untuk dua paket data tersebut, maka dapat diketahui bahwa *client-1* sama sekali tidak mengerjakan tugas sama sekali. Jika waktu *idle* (IT) sebuah node dinyatakan sebagai persentase waktu yang tidak digunakan oleh node tersebut untuk melakukan komputasi terhadap waktu eksekusi total maksimum penyelesaian jadwal atau dapat dinyatakan:

$$IT(p) = \frac{FT_{\max} - FT_p}{FT_{\max}} \times 100\% \quad (6)$$

dimana, $IT(p)$ adalah waktu *idle* dari node p , FT_p adalah waktu eksekusi total yang diperlukan oleh node p dalam menjalankan komputasi, dan FT_{\max} adalah waktu eksekusi total maksimum proses komputasi.

Berdasarkan persamaan tersebut diatas, maka waktu *idle* IT(p) untuk *client-1* adalah 100%, sedangkan waktu *idle client-2* adalah 0%, dan waktu *idle client-3* adalah 33,5%. Beban kerja dapat dikatakan merata jika semua node memiliki nilai waktu IT yang relatif sama. Jika waktu *idle* IT diambil nilai maksimumnya, maka proses komputasi terdistribusi di atas memiliki IT = 100%.



Gambar 5-4. Waktu eksekusi dan waktu *idle* metoda TP-GA (2 paket, 3 *client*)

Berbeda dengan metoda TP-GA seperti yang ditunjukkan pada Gambar 5-4. Metoda TP akan membagi semua tugas komputasi untuk memproses sebuah paket data. Dengan demikian, fungsi penjadwalan adalah mencari jadwal yang paling optimal yaitu menghasilkan sebuah penjadwalan yang memiliki waktu eksekusi total paling rendah. Dengan 2 buah paket data dan 3 buah *client* dengan kecepatan komputasi seperti tersebut di atas, maka waktu eksekusi total penyelesaian jadwal adalah FT = 3,66 satuan waktu untuk metoda TP-GA (lebih rendah dibanding metoda DP-WS yaitu membutuhkan waktu eksekusi total 4 satuan waktu). Waktu *idle* IT metoda TP-GA juga menjadi lebih merata untuk semua node komputasi, dengan nilai paling besar adalah 45,3% (lebih rendah dibanding metoda DP-WS yaitu IT maksimumnya adalah 100%).

Selanjutnya dengan melakukan analisis secara enumerasi untuk jumlah paket yang berbeda dengan 2 buah *client*, maka waktu total eksekusi untuk menyelesaikan sebuah jadwal dan waktu *idle* ditunjukkan pada tabel dan grafik berikut ini.

Tabel 1. Waktu Eksekusi Total (satuan waktu) Penyelesaian Penjadwalan Komputasi Terdistribusi DP-WS dan TP-GA untuk 2 *client*

Jml Data	FT		% TP/DP
	Data-Paralleism	Task-Parallelism	
1	4	4	100,00%
2	8	5,5	68,75%
3	8	8	100,00%
4	12	11	91,67%
5	16	13,5	84,38%
6	16	16	100,00%
7	20	19	95,00%
8	24	22	91,67%
9	24	24	100,00%
10	28	27	96,43%
Rata-rata:			92,789%

Tabel 1 menunjukkan waktu eksekusi total penyelesaian sebuah penjadwalan komputasi terdistribusi DP-WS dan TP-GA untuk jumlah paket data yang berbeda, mulai dari 1 paket data sampai dengan 10 paket data dengan jumlah node komputasi sebanyak 2 *client*. Kolom ke-4 menyajikan perbandingan waktu eksekusi total penyelesaian jadwal TP-GA dengan DP-WS dalam persen. Berdasarkan tabel tersebut maka persentase TP-GA terhadap DP-WS tidak pernah melebihi 100% artinya waktu eksekusi total penyelesaian jadwal metoda TP-GA selalu lebih rendah atau paling tidak sama dengan waktu eksekusi total penyelesaian jadwal metoda DP-WS. Waktu eksekusi total penyelesaian jadwal metoda TP-GA sama dengan waktu eksekusi total penyelesaian jadwal metoda DP-WS terjadi pada saat jumlah paket data yang diproses adalah 1, 3, 6, dan 9 paket, sehingga perbandingan waktu eksekusi total penyelesaian jadwal menjadi 100%. Sedangkan untuk jumlah data lainnya menunjukkan bahwa waktu eksekusi total penyelesaian jadwal TP-GA selalu lebih rendah dibanding waktu eksekusi total penyelesaian jadwal DP-WS. Rata-rata persentase perbandingan waktu eksekusi total penyelesaian jadwal tersebut adalah 92,789%.

Sedangkan pengamatan waktu *idle* IT untuk komputasi terdistribusi menggunakan metoda DP-WS dan TP-GA dengan 2 buah *client* untuk variasi jumlah paket data yang diproses ditunjukkan pada tabel berikut ini.

Tabel 2. Waktu *Idle* untuk Menyatakan Pemerataan Beban Kerja Komputasi Terdistribusi DP-WS dan TP-GA

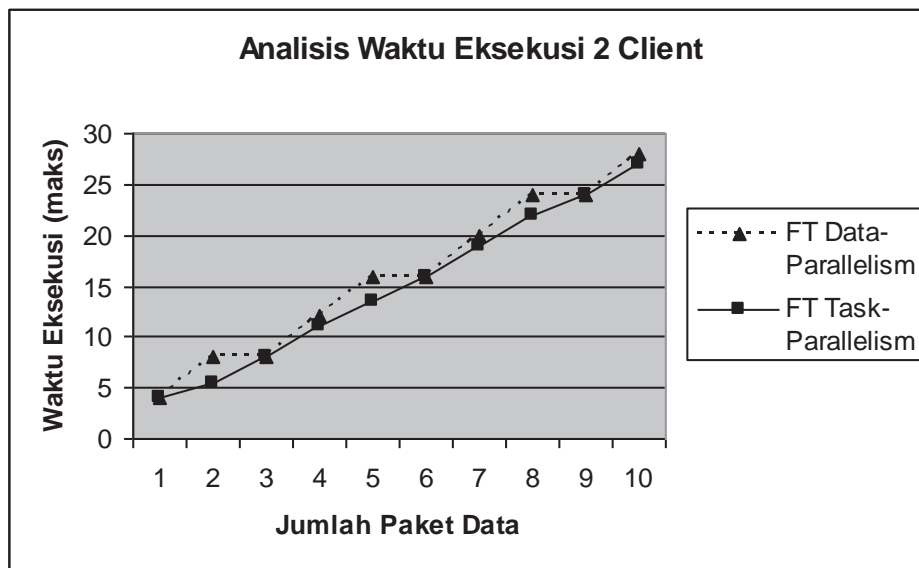
Jml Data	IT (%)	
	Data- Parallism	Task- Parallelism
1	100,00	37,50
2	50,00	9,09
3	0,00	0,00
4	33,30	10,00
5	25,00	3,70
6	0,00	16,00
7	20,00	5,26
8	16,67	4,54
9	0,00	0,00
10	14,29	1,85
Rata-rata:	25,93	8,79
Maks:	100,00	37,50

Tabel 2 menunjukkan waktu *idle* IT penjadwalan komputasi terdistribusi DP-WS dan TP-GA untuk jumlah paket data yang berbeda, mulai dari 1 paket data sampai dengan 10 paket data dengan jumlah node komputasi sebanyak 2 *client*. Waktu *idle* ini digunakan sebagai indikator pemerataan beban kerja keseluruhan node komputasi yang terlibat dalam sebuah penjadwalan. Berdasarkan tabel tersebut maka waktu *idle* untuk DP-WS dan TP-GA dapat dijelaskan sebagai berikut.

Waktu *idle* maksimum komputasi terdistribusi DP-WS adalah 100%, artinya ada sebagian node komputasi yang tidak melakukan komputasi sama sekali. Rata-rata waktu *idle* DP-WS untuk beberapa paket data dengan 2 buah *client* adalah 25,93%.

Waktu idle maksimum komputasi terdistribusi TP-GA adalah 37,5%, artinya seluruh node komputasi yang terlibat dalam penjadwalan melakukan komputasi. Sedangkan rata-rata waktu *idle* komputasi terdistribusi TP-GA adalah 8,79%.

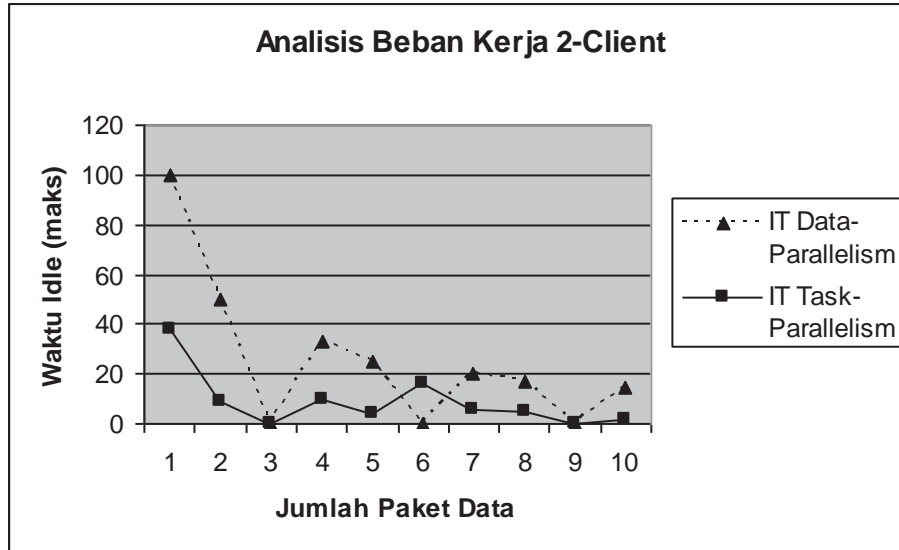
Berdasarkan nilai rata-rata waktu *idle* komputasi terdistribusi DP-WS dan TP-GA, maka dapat dinyatakan bahwa komputasi terdistribusi TP-GA memiliki pembagian beban kerja yang lebih merata dibanding komputasi terdistribusi DP-WS. Hal ini terbukti dari nilai rata-rata waktu *idle* TP-GA 8,79% adalah lebih rendah dibanding nilai rata-rata waktu *idle* DP-WS 25,93%.



Gambar 5-5. Analisis *Finishing Time* (FT) antara Paralelisme Data (*Data-Parallelism*) dengan Paralelisme Tugas (*Task-Parallelism*)

Gambar 5-5 menunjukkan hubungan jumlah paket data terhadap waktu eksekusi maksimum dalam menyelesaikan sebuah penjadwalan komputasi terdistribusi untuk metoda DP-WS dan TP-GA. Berdasarkan grafik tersebut nampak bahwa waktu eksekusi maksimum sistem komputasi terdistribusi TP-GA lebih rendah dibanding waktu eksekusi maksimum sistem komputasi terdistribusi DP-WS. Namun, ada beberapa titik yang menyatakan bahwa waktu eksekusi maksimum TP-GA dan DP-WS bernilai sama, yaitu pada saat banyaknya paket adalah 1, 3, 6, dan 9 paket. Berdasarkan grafik tersebut maka dapat dinyatakan bahwa model komputasi terdistribusi dengan metoda paralelisme tugas menggunakan skema penjadwalan algoritma genetika memiliki nilai waktu eksekusi

maksimum yang lebih rendah atau sama dengan model komputasi terdistribusi dengan metoda paralelisme data menggunakan skema penjadwalan *weighted-selective*.



Gambar 5-6. Analisis beban kerja antara Paralelisme Data (*Data-Parallelism*) dengan Paralelisme Tugas (*Task-Parallelism*)

Gambar 5-6 menunjukkan hubungan jumlah paket data terhadap waktu *idle* node komputasi dari penjadwalan komputasi terdistribusi untuk metoda DP-WS dan TP-GA. Waktu *idle* digunakan sebagai indikator keseimbangan beban kerja dalam penjadwalan komputasi terdistribusi. Berdasarkan grafik tersebut nampak bahwa waktu *idle* sistem komputasi terdistribusi TP-GA lebih rendah dibanding waktu *idle* sistem komputasi terdistribusi DP-WS. Namun, ada beberapa titik yang menyatakan bahwa waktu *idle* TP-GA dan DP-WS bernilai sama, yaitu pada saat banyaknya paket adalah 3, 6, dan 9 paket.

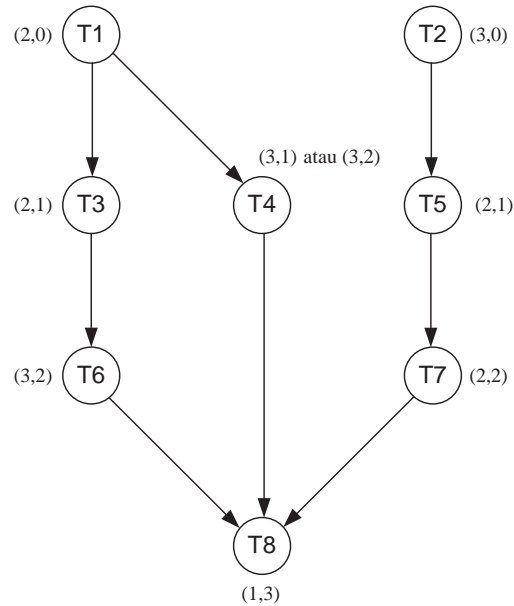
Berdasarkan grafik tersebut maka dapat dinyatakan bahwa model komputasi terdistribusi dengan metoda paralelisme tugas menggunakan skema penjadwalan algoritma genetika memiliki nilai waktu *idle* lebih rendah dibanding dengan model komputasi terdistribusi dengan metoda paralelisme data menggunakan skema penjadwalan *weighted-selective*. Hal ini membuktikan bahwa komputasi terdistribusi dengan metoda paralelisme tugas menggunakan skema penjadwalan algoritma genetika akan menghasilkan beban kerja komputasi terdistribusi yang lebih merata dibanding dengan metoda paralelisme data menggunakan skema penjadwalan *weighted-selective*.

3. Grafik Tugas untuk Penjadwalan Komputasi Parallel/Terdistribusi Menggunakan Algoritma Genetika.

Aplikasi Algoritma Genetika untuk penjadwalan komputasi parallel/terdistribusi pada prinsipnya memiliki komponen yang sama seperti Algoritma Genetika pada umumnya, yaitu terdiri dari (1) Struktur data untuk representasi kromosom; (2) Inisialisasi populasi; (3) evaluasi fitness menggunakan fungsi obyektif; (4) seleksi; (5) persilangan (cross over); (6) mutasi; (7) terminasi. Perbedaan utama aplikasi Algoritma Genetika untuk penjadwalan komputasi dengan aplikasi lainnya adalah terletak pada teknik representasi kromosom dan teknik dalam melakukan proses persialngan dan mutasi.

Proses penjadwalan komputasi parallel/terdistribusi memerlukan Grafik Tugas yang menggambarkan urutan dan hubungan input-output antar tugas untuk seluruh proses komputasi. Grafik Tugas ini akan memberikan informasi tugas-tugas mana saja yang dapat dikerjakan secara parallel dan tugas-tugas mana saja yang harus dikerjakan secara sekuensial. Grafik Tugas yang digunakan untuk eksplorasi Algoritma Genetika untuk penjadwalan komputasi parallel/terdistribusi menggunakan contoh Grafik Tugas seperti yang ditunjukkan pada Gambar 5-7 berikut ini.

Berdasarkan Grafik Tugas tersebut maka kajian tentang aplikasi Algoritma Genetika untuk skema penjadwalan komputasi parallel dapat dilakukan terutama dalam hal simulasi dan pengujian tentang representasi kromosom, perhitungan fitness, proses persilangan, dan mutasi dengan menggunakan perangkat lunak bantu Matlab. Hasil dari kajian ini adalah sebuah protipe algoritma dan metoda dari aplikasi Algoritma Genetika untuk penjadwalan komputasi parallel yang dapat digunakan sebagai dasar dalam pengembangan aplikasi menggunakan bahasa pemrograman JAVA dan pengembangan selanjutnya adalah untuk komputasi terdistribusi.



Gambar 5-7. Grafik tugas DAG dengan nilai bobot dan ketinggian tiap node

Gambar 5-7 tersebut menunjukkan sebuah grafik tugas yang terdiri 8 (delapan) node atau 8 (delapan) buah tugas komputasi, dengan nilai parameter berupa waktu eksekusi diikuti nilai ketinggian setiap node ditulis disamping masing-masing node. Sebagai contoh, node T_1 memiliki parameter (2,0) artinya waktu eksekusi yang diperukan untuk menjalankan tugas T_1 adalah 2 satuan waktu dan nilai ketinggian T_1 adalah 0. Parameter node T_4 ada dua yaitu (3,1) dan (3,2), yang berarti bahwa nilai ketinggian dari T_4 dapat bernilai 1 yaitu setingkat dengan nilai ketinggian T_3 dan T_5 atau nilai ketinggian T_4 dapat bernilai 2 yaitu setingkat dengan nilai ketinggian T_6 dan T_7 . Hal ini berarti bahwa posisi node T_4 dapat disejajarkan dengan letak T_3 dan T_5 atau dapat juga disejajarkan dengan node T_6 dan T_7 . Sedangkan bobot tiap-tiap garis berarah atau waktu komunikasi dalam grafik tugas ini diabaikan.

4. Operator Genetika untuk Skema Penjadwalan Komputasi Paralel

Salah satu fungsi dari operator genetika adalah untuk menghasilkan titik lacak baru berdasarkan titik lacak populasi saat itu. Titik-titik lacak baru secara khusus terbentuk dengan mengkombinasikan atau mengatur kembali bagian-bagian representasi string dari titik lacak yang lama. Ide tersebut menyatakan bahwa dengan pemilihan string

yang sesuai dalam representasi akan menggambarkan ”*goodness*” dari titik lacak tersebut. Dengan demikian, dengan mengkombinasikan (menggabungkan) struktur yang bagus dari dua buah titik lacak, mungkin akan menghasilkan titik lacak baru yang lebih baik. Berkaitan dengan gagasan ini terhadap penjadwalan komputasi terdistribusi, bagian tertentu dari penjadwalan mungkin menjadi milik penjadwalan yang optimal. Dengan menggabungkan beberapa bagian optimal ini, kita dapat menemukan penjadwalan optimal secara lebih efisien.

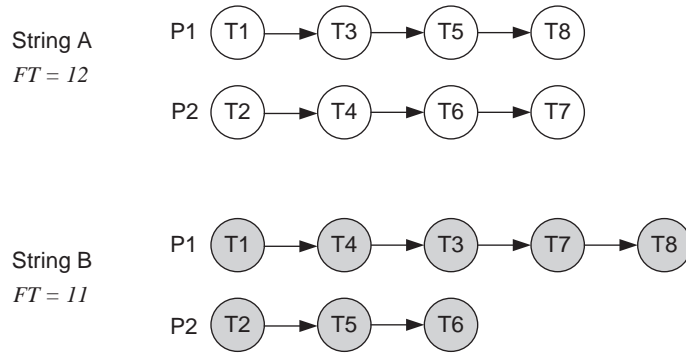
Jika ruang representasi string dan ruang lacak tidak dalam korespondensi satu-satu, maka kita harus merancang operator genetika secara hati-hati. Jika jumlah string yang berhubungan dengan titik lacak yang tidak diijinkan relatif sedikit, maka diperbolehkan operator genetika membuat string ilegal dan kemudian nanti dapat dibuang dengan menggunakan tes kelegalannya. Jika jumlah string yang berkaitan dengan titik lacak ilegal cukup besar jika dibandingkan dengan titik lacak legal, maka pembangkitan string ilegal akan membuang-buang waktu komputasi saja baik dalam hal pembangkitan maupun pengujiannya. Dalam hal ini, operator genetika yang baik akan selalu membangkitkan string yang menggambarkan titik lacak legal. Namun demikian kondisi ini tidak bisa selalu dipenuhi karena beberapa faktor seperti kesulitan implementasi dan waktu komputasi yang tinggi saat operasi.

Untuk permasalahan penjadwalan komputasi terdistribusi, operator genetika yang digunakan harus menjalankan hubungan preseden intraprosesor, sesesuai dengan kelengkapan dan keunikan tugas dalam penjadwalan. Hal ini akan menjamin bahwa string baru yang dihasilkan akan selalu menunjukkan titik lacak legal.

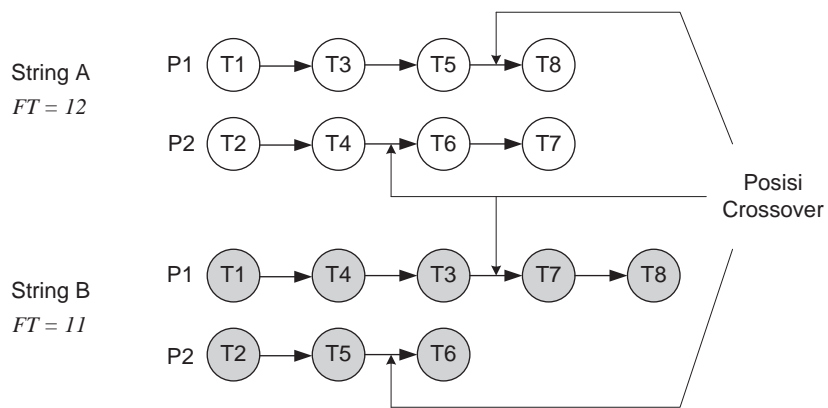
Seleksi biasanya menggunakan *roulette wheel*. Operator genetika yang diperlukan dalam penjadwalan multiprosesor menggunakan algoritma genetika adalah *crossover* dan mutasi. Ada beberapa algoritma yang dikembangkan untuk operator *crossover* dan mutasi.

Algoritma Crossover

Anggap dua buah string (jadwal) seperti ditunjukkan pada gambar berikut ini.



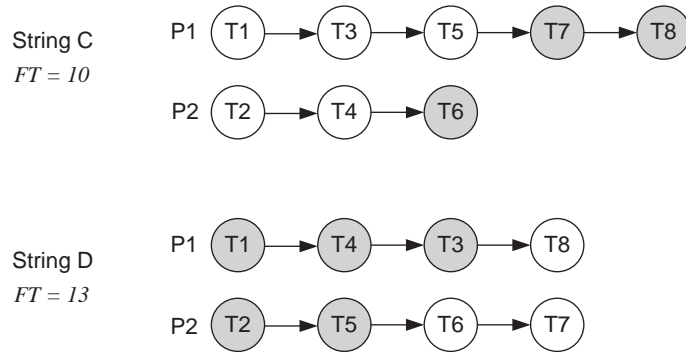
Gambar 5-8. Dua buah string induk yang akan mengalami *crossover*



Gambar 5-9. Letak titik pemotongan tiap-tiap kromosom

Kita dapat membuat string baru dengan menukarkan bagian dari dua buah string tersebut dengan menggunakan metoda berikut ini.

1. Pilih letak (posisi *crossover*) dimana kita dapat memotong string menjadi dua buah bagian
2. Tukarkan potongan akhir P1 string A dengan potongan akhir P1 string B
3. Tukarkan potongan akhir P2 string A dengan potongan akhir P2 string B



Gambar 5-10. Hasil *crossover* dengan nilai *fitness* yang baru

String baru yang dihasilkan adalah string C dan String D. Nampak bahwa string C memiliki waktu penyelesaian yang lebih kecil dibanding string A dan B. Dalam kenyataannya, waktu penyelesaian ini merupakan waktu penyelesaian optimal dari grafik tugas dengan menggunakan 2 buah prosesor. Operasi yang dijelaskan di atas dapat dikembangkan dengan mudah untuk sejumlah p prosesor dan nampak cukup efektif. Tetapi, kita masih harus mendefinisikan metoda dalam memilih letak *crossover*, dan menunjukkan bahwa string baru yang dihasilkan adalah legal.

Aturan menentukan posisi titik *crossover*

Legalitas string baru yang dihasilkan adalah sangat berhubungan erat dengan pemilihan posisi *crossover*. Catatan bahwa letak *crossover* yang digunakan contoh di atas selalu terletak antara tugas-tugas dengan dua perbedaan nilai ketinggian ($height(T_5) \neq height(T_8)$, $height(T_4) \neq height(T_6)$, dan seterusnya). Kita dapat membuktikan teori tersebut sebagai berikut:

Teorema 1.

Jika letak *crossover* dipilih sedemikian rupa sehingga kondisi berikut ini muncul:

1. Nilai ketinggian dari tugas-tugas setelah letak *crossover* adalah berbeda
2. Nilai ketinggian dari seluruh tugas-tugas tepat di depan letak *crossover* adalah sama, maka string baru yang dibangkitkan akan selalu legal.

Bukti:

Kita perlu menunjukkan bahwa hubungan presenden tidak dilanggar dan kelengkapan dan keunikan tugas masih dipertahankan. Anggap kondisi yang ditunjukkan pada Gambar 2-10(a). Kita memiliki kondisi sebagai berikut:

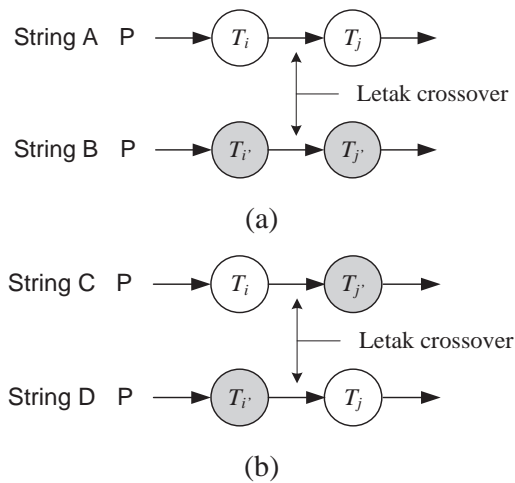
$$height(T_i) < height(T_j), height(T_{i'}) < height(T_{j'}), height(T_i) = height(T_{i'})$$

Berdasarkan kondisi tersebut, jelas bahwa seluruh tugas-tugas dengan ketinggian lebih dari $height(T_i)$ ditukar antara dua string, maka dijamin tidak ada tugas yang terhapus atau terduplikasi. Hal ini berarti bahwa kelengkapan dan keunikan tugas tetap dipertahankan.

Setelah proses operasi *crossover* (Gambar 2-10(b)), hubungan berikut ini tetap valid:

$$height(T_i) < height(T_{j'}), height(T_{i'}) < height(T_j).$$

Oleh karena itu, string baru yang dibangkitkan masih memenuhi kondisi pengurutan nilai ketinggian, sehingga string baru yang dibangkitkan adalah jadwal yang legal.



Gambar 5-11. Letak titik *crossover* dan hasil proses *crossover*

Operasi *crossover* menggunakan fakta di atas dan memilih letak *crossover* sedemikian rupa sehingga kondisi 1) dan 2) selalu terpenuhi, dan dapat diringkas dalam algoritma berikut ini.

Algoritma Crossover

Algoritma ini menjalankan operasi *crossover* pada dua buah string (A dan B) dan membangkitkan dua buah string baru.

1. Pilih lokasi *crossover*. Bangkitkan sebuah bilangan random c , antara 0 dan nilai maksimum ketinggian dalam grafik tugas.
2. *Loop* untuk tiap prosesor. Untuk tiap-tiap prosesor P_i dalam string A dan string B lakukan langkah 3.

3. Temukan lokasi *crossover*. Cari tugas terakhir T_{ji} dalam prosesor P_i yang memiliki ketinggian c , dan T_{ki} adalah tugas-tugas setelah T_{ji} . Yaitu, $c = height'(T_{ji}) < height'(T_{ki})$ dan $height'(T_{ji})$ adalah sama untuk seluruh nilai i .
4. *Loop* untuk setiap prosesor. Untuk setiap prosesor P_i dalam string A dan string B lakukan langkah 5.
5. *Crossover*. Dengan menggunakan lokasi *crossover* yang dipilih pada langkah 3, tukarkan bagian terakhir dari string A dan B untuk setiap prosesor P_i .

Meskipun operasi *crossover* sangat kuat, operasi tersebut adalah acak secara alami dan mungkin menghilangkan solusi optimal. Tepatnya, aplikasinya dikendalikan oleh probabilitas *crossover* yang nilainya ditentukan secara eksperimental. Lebih jauh lagi, kita selalu menjaga solusi terbaik yang telah ditemukan dengan cara memasukkan solusi terbaik yang telah ditemukan tersebut ke dalam generasi berikutnya.

Tsujimura (1997) mengembangkan algoritma operator genetika, yaitu dua buah operator genetika *crossover* dalam bentuk operator-1 dan operator-2 seperti yang dijelaskan berikut ini.

Operator-1:

1. Bangkitkan sebuah bilangan random c dengan jangkah $[1, \max\{height'\}]$
2. Tempatkan cut point pada setiap prosesor sedemikian rupa sehingga ketinggian tugas sebelum cut point adalah lebih kecil dari c dan tugas-tugas setelah cut point memiliki nilai ketinggian lebih besar atau sama dengan c . Kemudian, kita mendapatkan dua penjadwalan parsial di setiap prosesor.

Contoh.

$c = 2$;

$$height'(T_6), height'(T_7), height'(T_8) \geq 2$$

Sehingga:

$$v = \begin{cases} P_1(T_1 & T_5 & T_4 & | \mathbf{T}_7) \\ P_2(T_2 & T_3 & | \mathbf{T}_6 & \mathbf{T}_8) \end{cases}$$

Dan hasil operator-1 adalah:

$$v' = \begin{cases} P_1(T_1 & T_5 & T_4 & | \mathbf{T}_6 & \mathbf{T}_8) \\ P_2(T_2 & T_3 & | \mathbf{T}_7) \end{cases}$$

Operator-2:

1. Bangkitkan sebuah bilangan random c dengan jangkah $[1, \max\{height'\}]$
2. Untuk setiap prosesor, ambil semua tugas yang $height'$ sama dengan c
3. Gantikan posisi dari seluruh tugas secara random

Contoh:

$c = 1$;

Sehingga,

$$v = \begin{cases} P_1(T_1 \quad \mathbf{T}_5 \quad \mathbf{T}_4 \quad T_6 \quad T_8) \\ P_2(T_2 \quad \mathbf{T}_3 \quad T_7) \end{cases}$$

Dan hasil operator adalah:

$$v' = \begin{cases} P_1(T_1 \quad \mathbf{T}_3 \quad T_6 \quad T_8) \\ P_2(T_2 \quad \mathbf{T}_4 \quad \mathbf{T}_5 \quad T_7) \end{cases}$$

Algoritma Mutasi

Mutasi dapat dianggap sebagai sebuah pergantian acak yang jarang terjadi (dengan probabilitas yang cukup kecil) dari nilai sebuah string. Orang dapat membayangkan mutasi sebagai sebuah mekanisme pelarian untuk konvergensi dini. Untuk permasalahan penjadwalan multiprosesor, mutasi dilakukan dengan jalan menukarkan dua buah tugas yang memiliki nilai ketinggian yang sama. Operasi mutasi diringkas dalam sebuah algoritma berikut ini.

Algoritma ini menjalankan operasi mutasi pada sebuah string dan membangkitkan sebuah string baru.

1. Ambil sebuah tugas. Secara acak ambil sebuah tugas T_i .
2. Cocokkan nilai ketinggiannya. Lacak string tersebut untuk sebuah tugas T_j yang memiliki nilai ketinggian sama.
3. Tukarkan posisi tugas. Bentuk sebuah string baru dengan menukarkan dua buah tugas, T_i dan T_j dalam penjadwalan.

Biasanya, frekuensi penggunaan operator mutasi dikendalikan oleh probabilitas mutasi yang nilainya ditentukan secara eksperimen.

Tsujimura (1997) membuat operator mutasi yang dinyatakan dalam bentuk operator-3 sebagai berikut:

Operator-3:

Pada saat membangkitkan populasi awal, kita menetapkan nilai *height'* untuk setiap tugas dan nilainya tidak ditukar untuk seluruh generasi, karena ada sebuah *broad range*. Untuk menghilangkan ketidaknyamanan ini, kita mengajukan operator ketiga, yang tujuannya adalah membuat ruang lacak yang lebih besar untuk menemukan kromosom yang lebih baik. Operator-3 ini sebenarnya mengantisipasi adanya nilai *height'* lebih dari satu untuk sebuah tugas yaitu hasil persamaan dalam menghitung *height* dan *height'*. Sebagai contoh adalah T_4 yang memiliki nilai *height'* 1 atau 2. Langkah-langkah yang digunakan dalam operator-3 adalah:

1. Ambil tugas-tugas yang nilai *height'* dapat ditukar.
2. Pilih beberapa tugas-tugas ini secara random dengan sebuah probabilitas kecil
3. Tukar nilai *height'* dari tugas yang terpilih dengan nilai yang tugas tersebut dapat diperoleh, dan pindahkan ke posisi yang sesuai menurut *height'* yang ditukar.

Contoh.

Kita dapat memilih tugas T_4 sebagai calon yang ditukar, yaitu nilai *height'* dari T_4 dapat bernilai 1 atau 2. Setelah penukaran, pindahkan posisi tugas T_4 dari sebelum T_5 ke posisi setelah T_5 .

$$height'(T_4) = 1 \rightarrow 2$$

$$height'(T_5) = 1$$

$$v = \begin{pmatrix} P_1(T_1 & T_3 & T_6 & T_8) \\ P_2(T_2 & \mathbf{T_4} & \mathbf{T_5} & T_7) \end{pmatrix}$$

$$v' = \begin{pmatrix} P_1(T_1 & T_3 & T_6 & T_8) \\ P_2(T_2 & \mathbf{T_5} & \mathbf{T_4} & T_7) \end{pmatrix}$$

Semua aturan operator Algoritma Genetika yang telah dijelaskan di atas akan digunakan dalam mengembangkan aplikasi Algoritma Genetika untuk skema penjadwalan komputasi paralel dan selanjutnya akan dikembangkan untuk skema penjadwalan komputasi terdistribusi khususnya dalam ruang lingkup komputasi Software-Defined Radio (SDR).

5. Fungsi Fitness Algoritma Genetik untuk Penjadwalan Komputasi Terdistribusi

Salah satu komponen Algoritma Genetika dalam menjalankan proses optimasi adalah fungsi fitness yang diperoleh dari fungsi obyektif. Kriteria yang digunakan dalam proses optimasi penjadwalan komputasi dengan Algoritma Genetika dalam penelitian ini menggunakan nilai waktu eksekusi dalam menyelesaikan tugas komputasi secara keseluruhan dan waktu idle sebagai kriteria untuk menentukan beban kerja komputasi. Berdasarkan kriteria tersebut, maka fungsi fitness yang digunakan dalam proses optimasi penjadwalan dapat dinyatakan sebagai berikut ini.

$$Fitness = \frac{1}{(FT(S) + WB(S))^2} \quad (7)$$

dimana:

- $FT(S) = \max_{P_j} fip(P_j)$, dengan $fip(P_j)$ adalah waktu penyelesaian tugas dari prosesor P_j , dan
- $WB(S) = \overline{itp(P_i)}$, dengan $itp(P_i)$ adalah waktu idle dari prosesor P_i
- serta $i = j = 1, 2, \dots, N_p$ dan N_p adalah banyaknya prosesor.

Berdasarkan persamaaan (7) tersebut nampak bahwa sebuah jadwal komputasi yang diinginkan adalah jadwal komputasi untuk beberapa prosesor yang memiliki waktu penyelesaian paling rendah dan beban kerja yang merata dilihat dari rata-rata waktu idle yang rendah. Fungsi fitness ini akan digunakan dalam melakukan proses optimasi penjadwalan menggunakan Algoritma Genetika.

C. Faktor-Faktor Pendukung

Beberapa faktor eksternal yang mendukung kelancaran pelaksanaan kegiatan penelitian ini antara lain:

1. Ketersediaan ruangan, komputer, akses internet, dan listrik di Jurusan Pendidikan Teknik Elektronika dan tim peneliti memiliki hak akses secara penuh untuk semua sarana pendukung yang ada.
2. Beberapa algoritma pemrograman yang diperlukan relatif mudah ditemukan baik secara on-line maupun off-line, meskipun perlu dilakukan seleksi dan modifikasi sehingga diperoleh algoritma yang sesuai.

3. Komunitas pengembang SDR GNU-Radio telah memiliki fasilitas informasi dan komunikasi sehingga memudahkan untuk mencari sumber acuan dan diskusi jika diperlukan.
4. Algoritma Genetika merupakan sebuah algoritma optimasi yang banyak digunakan di bidang kendali dan banyak dibahas di bidang Softcomputing sehingga memudahkan dalam memperoleh materi dan penjelasan tentang Algoritma Genetika.

D. Faktor-Faktor Penghambat

Beberapa faktor penghambat (internal dan eksternal) yang dihadapi dalam menjalankan kegiatan penelitian ini antara lain:

1. Perubahan besarnya dana yang disetujui lebih rendah dari dana yang diusulkan dan waktu pelaksanaan penelitian yang lebih singkat sangat berpengaruh dalam proses kegiatan penelitian ini. Beberapa komponen dasar (perangkat keras dan perangkat lunak) yang sangat diperlukan dalam penelitian ini tidak bisa diadakan sehingga beberapa kegiatan yang membutuhkan perangkat tersebut tidak dapat dilaksanakan.
2. GA sarat akan kebolehjadian (probabilitas) sehingga berbagai macam kemungkinan bisa berpengaruh dalam menghasilkan solusi. Untuk itu, berbagai macam metode yang digunakan dalam tiap komponen GA harus dikaji secara cermat sehingga GA yang digunakan benar-benar sesuai dengan kasus yang akan dicari solusinya (dalam hal ini solusi tentang penjadwalan komputasi). Hal ini sesuai dengan karakteristik GA yaitu bersifat kasuistik terhadap penyelesaian sebuah masalah. Artinya, penyelesaian sebuah kasus menggunakan GA tidak akan selalu sama dengan penyelesaian kasus lainnya yang sama-sama menggunakan GA dalam hal metode dan parameter yang digunakan dalam menjalankan GA. Dengan kata lain, setiap penyelesaian kasus menggunakan GA memerlukan kajian yang mendalam sehingga aplikasi GA benar-benar optimal sesuai dengan permasalahan yang akan dicarikan solusinya. Permasalahan ini tentu saja akan mendorong proses penelitian menjadi lama karena kasus yang dihadapi selalu berkembang dan waktu yang digunakan untuk eksplorasi juga semakin lama.
3. Padatnya jadwal kegiatan akademik selain penelitian bagi tim peneliti sangat membatasi kegiatan penelitian Tahun I (2013) yang harus dilaksanakan di Jurusan Pendidikan Teknik Elektronika FT UNY.

E. Jalan Keluar/Solusi

1. Perubahan anggaran yang mengakibatkan dana yang disetujui lebih rendah dari biaya yang diusulkan dan waktu pelaksanaan yang lebih singkat disikapi dengan menggeser sebagian target atau luaran penelitian ke Tahun II (2014), dengan harapan dapat teratasi dalam pelaksanaan kegiatan Tahap II (Tahun II).
2. Untuk mengatasi permasalahan tentang karakteristik GA dalam mencari solusi sebuah permasalahan yang dijelaskan pada Bagian D.2 adalah dengan memilih salah satu metode yang dianggap cukup untuk proses optimasi meskipun belum menghasilkan GA yang benar-benar optimal. Kajian metoda lain yang lebih mendalam dan memiliki unjuk kerja yang lebih baik dimasukkan dalam kegiatan tindak lanjut penelitian ini, yaitu kegiatan untuk mengkaji tentang optimasi metoda dan parameter GA yang digunakan dalam menyelesaikan sebuah permasalahan (dalam hal ini terkait dengan penjadwalan komputasi terdistribusi).
3. Kegiatan penelitian yang tidak bisa dilaksanakan dalam jam kerja, diberikan alokasi waktu pada hari Sabtu dan Minggu dan kegiatan yang tidak bisa dilaksanakan di laboratorium (ruangan di lingkungan Jurusan Pend. Teknik Elektronika) karena terbatas jam kerja dilaksanakan di tempat lain yang masih memungkinkan untuk melaksanakan penelitian.

F. Ketercapaian

Berdasarkan hasil kegiatan penelitian Tahun I (2013) dan Tahun II (2014) yang telah dilaksanakan sampai dengan Laporan Akhir ini di buat, seluruh target penelitian yang direncanakan (sesuai dengan revisi tahapan Tahun I dan Tahun II) sudah terpenuhi 100% .

BAB 6

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil simulasi dan pengujian tentang Algoritma Genetika khususnya dalam melakukan proses terminasi, maka Algoritma Genetika yang dipilih GA Terminated-N agar proses optimasi memiliki waktu komputasi yang lebih singkat. Analisis tentang perbedaan WS-DP dengan TP-GA membuktikan bahwa skema penjadwalan TP-GA memiliki beban kerja yang lebih merata dilihat dari waktu istirahat (*idle time*) node-node komputasi.

B. Saran

Berdasarkan keterbatasan tentang pelaksanaan kegiatan penelitian ini, maka tindak lanjut dari hasil penelitian ini adalah kajian yang lebih mendalam terhadap:

- Analisis yang lebih mendalam tentang perbedaan WS-DP dengan TP-GA
- Eksplorasi operasi genetika (perhitungan nilai fitness, seleksi, crossover, dan mutasi) dari Algoritma Genetika yang sesuai dengan kebutuhan penjadwalan komputasi paralel atau komputasi terdistribusi.
- Eksperimen terhadap kasus komputasi SDR dalam bentuk grafik tugas (Task Graph) dalam melakukan optimasi penjadwalan komputasi terdistribusi.

DAFTAR PUSTAKA

- Abdeyazdan, M., dan Rahmani, A. M. (2008) : *Multiprocessor Task Scheduling Using a New Prioritizing Genetic Algorithm based on number of Task Children*, Distributed and Parallel Systems, Springer US, pp. 105 – 114. [On-line], Available at <http://www.springerlink.com/content/v57t546136578451/fulltext.pdf>.
- Bose, V. G. (1999) : Design and Implementation of Software Radios Using a General Purpose Processors, [Online] <http://www.sigmobility.org/phd/1999/theses/bose.pdf>.
- Corneloup, M. (2002) : Open Architecture for Software Defined Radio Systems, [Online] <http://www.csem.ch/slats/files/AmblesideCom-Final.pdf> - Similar pages.
- Gweon-Do JO. (2005) : A DSP-Based Reconfigurable SDR Platform for 3G Systems, *Journal of IEICE Transaction on Communication*, **Vol. E88-B No.2 Feb. 2005**, 678 – 686.
- Harris, F.J., Dick, C., dan Rice, M. (2003) : Digital Receivers and Transmitter Using Polyphase Filter Banks for Wireless Communications, *Journal of IEEE Transaction on Microwave Theory and Techniques*, **Vol. 51, No.4**, 1395 – 1412.
- Lee, Y. H., dan Chen, C. (2003) : A Modified Genetic Algorithm For Task Scheduling In Multiprocessor Systems, [On-line], Available at <http://parallel.iis.sinica.edu.tw/cthpc2003/papers/CTHPC2003-18.pdf>.
- Malm, H. V. (2005) : Implementing Physycal and Data Link Control Layer in the GNU Radio Software-Defined Radio Platform, [On-line] http://typo3.cs.uni-paderborn.de/fileadmin/Informatik/AG-Karl/Pubs/vmalm05-sa-gsr_aloha.pdf.
- Marpanaji, E., Riyanto T., B., Langi., A. Z. R., dan Kurniawan, A., (2007) : Pengukuran Unjuk Kerja Modulasi GMSK pada Software-Defined Radio Platform, *Jurnal Telkomnika*, **Vol. 5, No. 2 Agustus 2007**, 73 – 84.
- Marpanaji, E., T., B., Langi., A. Z. R., dan Kurniawan, A., (2008) : Studi Eksperimen Unjuk-Kerja Modulasi DBPSK pada Platform Software-Defined Radio (SDR), *Jurnal Technoscientia*, **Vol. 1 No. 1 Agustus 2008**, 14 – 22.
- Marpanaji, E., Riyanto T., B., Langi., A. Z. R., dan Kurniawan, A., (2008) : Experimental Study of DQPSK Modulation on SDR Platform, *ITB Journal of Information and Communication Technology*, **Vol. 1 No. 2 November 2008**, 84 – 98.
- Marpanaji, E., dkk. (2009) : Dekomposisi Tugas-tugas Software-Defined Radio (SDR), *Jurnal Technoscientia*, **Vol. 2 No. 1 Agustus 2009**, 50 – 60.
- Marpanaji, E.**, Riyanto T., B., Langi., A. Z. R., Kurniawan, A., Mahendra, A., dan Liung, T., (2007), Experimental Study of DQPSK Modulation on SDR Platform. *Proceeding of International Conference on Rural ICT 2007*, Bandung, August, 6th, 2007.
- Riyanto T., B., **Marpanaji, E.**, Langi., A. Z. R., Kurniawan, A., Mahendra, A., dan Liung, T., (2007) : Software Architecture of Software-Defined Radio (SDR). *Proceeding of International Conference on Rural ICT 2007*, Bandung, August, 6th, 2007.
- Mahendra, A., Langi, A. Z. R., Riyanto T., B., **Marpanaji, E.**, dan Liung, T., (2007) : Optimasi dan Sintesis Desain FPGA pada Software-Defined Radio Menggunakan Synopsys. *Proceeding of National Conference on Computer Science & Information Technology University of Indonesia 2007 (NACSIT 2007)*, Depok, January, 29th – 30th, 2007.
- Marpanaji, E.**, Riyanto T., B., Langi, A. Z. R., Kurniawan, A., Mahendra, A., dan Liung, T., (2006) : Simulation and Experimental Study of GMSK Modulation on SDR Platform. *Proceeding of TSSA and WSSA 2006. The 3rd Conference on Telematics System, Services and*

Applications and 1st Conference on Wireless System, Service and Applications, ITB 2006, Bandung, 8th – 9th December 2006.

Marpanaji, E., Riyanto T., B., Langi, A. Z. R., Kurniawan, A., dan Mahendra, A., (2006) : Arsitektur Software-Defined Radio (SDR). *Prosiding Seminar Nasional Ilmu Komputer dan Aplikasinya (SNIKA) 2006 Universitas Katolik Parahyangan*, November 2006.

Marpanaji, E., Riyanto T., B., Langi, A. Z. R., Kurniawan, A., dan Mahendra, A., (2006) : Aspek Komputasi Platform Software-Defined Radio. *Prosiding Seminar Nasional Indonesian Conference on Telecommunications (ICTel) 2006, STT Telkom Bandung*, 20 – 22 September 2006.

Michalewics, Z. (1996) : *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, New York, pp. 45 – 55.

Pucker, L. (2002) : Channelization Techniques for Software Radio, [Online] http://www.spectrumsignal.com/channel_techniques/Channelization_Paper_SDR_forum.pdf.

Rahmani, A. M., dan Vahedi, M. A. (2008) : A Novel Task Scheduling In Multiprocessor Systems With Genetic Algorithm By Using Elitism Stepping Method, *Journal of Computer Science*, **Vol. 7 No. 2 June 2008**. [On-line]. Available at <http://www.dcc.ufla.br/infocomp/artigos/v7.2/art08.pdf>.

Reed, J. H. (2002) : *Software Radio: A Modern Approach to Radio Engineering*, New Jersey, Prentice Hall.

Tsujimura, Y., dan Gen M. (1997) : *Genetic Algorithms for Solving Multiprocessor Scheduling Problems, Simulated Evolution and learning*, Heidelberg, Springer Berlin, pp. 106 – 115. [On-line], Available at <http://www.springerlink.com/content/w461447527q15086/fulltext.pdf>.

Valentin, S., Malm, H. V., dan Karl, H. (2006) : Evaluating The GNU Software Radio Platform for Wireless Testbeds, [On-line] http://typo3.cs.uni-paderborn.de/fileadmin/Informatik/AG-Karl/Pubs/TR-RI-06-273-gnuradio_testbed.pdf, February, 2006.

Zheng, K., Li G., dan Huang L. (2007) : A weighted Scheduling Scheme in an Open Software Radio Environment, Communication, Computer and Signal Processing, PacRim 2007, *Proceedings of IEEE Pacific Rim Conference, Canada, August, 2007*, pp. 561 – 564.

LAMPIRAN

A. Surat Kontrak Penelitian



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS NEGERI YOGYAKARTA

LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT

Alamat : Karangmalang, Yogyakarta 55281

Telp. (0274) 550840, Fax (0274) 518617, 550839, email: lppm.uny@gmail.com

**SURAT PERJANJIAN INTERNAL
PELAKSANAAN PENELITIAN DESENTRALISASI BOPTN SKIM:
FUNDAMENTAL
NOMOR : 12/FUNDAMENTAL-BOPTN/UN.34.21/2014**

Pada hari ini Senin tanggal tujuh belas bulan Maret tahun dua ribu empat belas, kami yang bertanda tangan di bawah ini:

1. Dr. Widarto, M.Pd : Sekretaris Lembaga Penelitian dan Pengabdian kepada Masyarakat Universitas Negeri Yogyakarta yang berkedudukan di Yogyakarta dalam hal ini bertindak untuk dan atas nama perguruan tinggi tersebut; selanjutnya disebut PIHAK PERTAMA.
2. Dr. EKO MARPANAJI : Ketua Tim Peneliti dari Skim Penelitian *Fundamental*, yang beralamat di *FT* Universitas Negeri Yogyakarta, selanjutnya disebut PIHAK KEDUA.

Kedua belah pihak berdasarkan:

- 1) Daftar Isian Pelaksanaan Pekerjaan Anggaran (DIPA) Universitas Negeri Yogyakarta No.: SP DIPA-023.04.2.189946/2014 tanggal 5 Desember 2013;
- 2) Surat Keputusan Rektor UNY No. : 915 Tahun 2013 tanggal 19 Desember 2013 tentang Penetapan Pemenang Penelitian (On Going dan Usulan Baru) Skim Penelitian Desentralisasi Dosen Universitas Negeri Yogyakarta Tahun 2014;
- 3) Surat Perjanjian Pelaksanaan Kegiatan Penelitian Fundamental Universitas Negeri Yogyakarta Tahun 2014 antara Ketua Lembaga Penelitian dan Pengabdian kepada Masyarakat Universitas Negeri Yogyakarta dengan Sekretaris Lembaga Penelitian dan Pengabdian kepada Masyarakat UNY Nomor: 231/Fund.-BOPTN/UN34.21/2014 tanggal 17 Maret 2014.

PIHAK PERTAMA dan PIHAK KEDUA secara bersama-sama bersepakat mengikatkan diri dalam suatu Perjanjian Pelaksanaan Penelitian Fundamental dengan ketentuan dan syarat-syarat sebagai berikut:

Pasal 1

PIHAK PERTAMA memberi tugas kepada PIHAK KEDUA, dan PIHAK KEDUA menerima tugas tersebut sebagai penanggung jawab dan mengkoordinasikan pelaksanaan Penelitian Fundamental dengan judul dan nama Ketua/Anggota Peneliti sebagai berikut:

- Judul : APLIKASI ALGORITMA GENETIKA PADA SKEMA PENJADWALAN KOMPUTASI TERDISTRIBUSI (STUDI KASUS PADA SOFTWARE-DEFINED RADIO) TAHUN II
- Ketua Peneliti : Dr. EKO MARPANAJI
- Anggota : 1. Adi Dewanto, M.Kom.
2. Didik Hariyanto, MT.
3. -

Pasal 2

- 1) PIHAK PERTAMA memberikan dana penelitian yang tersebut pada pasal 1 sebesar Rp 40.000.000(Empat puluh juta rupiah) yang dibebankan pada Daftar Isian Pelaksanaan Anggaran (DIPA) No.: SP DIPA-023.04.2.189946/2014 tanggal 5 Desember 2013.
- 2) PIHAK KEDUA berhak menerima dana tersebut pada ayat (1) dan berkewajiban menggunakan sepenuhnya untuk pelaksanaan penelitian sebagaimana pasal 1 sampai selesai sesuai ketentuan pembelanjaan keuangan negara.

Pasal 3

Pembayaran dana Penelitian Fundamental ini akan dilaksanakan melalui Lembaga Penelitian dan Pengabdian kepada Masyarakat UNY dan dibayarkan secara bertahap dengan ketentuan sebagai berikut:

- (1) **Tahap Pertama 70%** sebesar Rp. 28.000.000 (Dua puluh delapan juta rupiah) setelah Surat Perjanjian ini ditandatangani oleh kedua belah pihak,
- (2) **Tahap Kedua 30%** sebesar Rp12.000.000 (Dua belas juta rupiah) setelah PIHAK KEDUA menyerahkan Laporan Akhir Hasil Pelaksanaan Penelitian Fundamental kepada PIHAK PERTAMA dalam bentuk hardcopy sebanyak 6 (enam) eksemplar dan softcopy (CD dalam format "pdf") paling lambat tanggal **31 Oktober 2014**.
- (3) PIHAK KEDUA wajib membuat Laporan Kemajuan Pelaksanaan Penelitian dan Laporan Penggunaan Keuangan sejumlah termin I (70%) yang diserahkan kepada PIHAK PERTAMA dalam bentuk hardcopy masing-masing 2 (dua) eksemplar paling lambat tanggal 30 Juni 2014 serta **mengunggahnya (upload) ke SIM-LITABMAS antara tanggal 15-30 Juni 2014**.
- (4) PIHAK KEDUA wajib membuat Laporan Hasil dan Laporan Penggunaan Keuangan sejumlah (100%) yang diserahkan kepada PIHAK PERTAMA dalam bentuk hardcopy masing-masing 2 (dua) eksemplar paling lambat tanggal 31 Oktober 2014 serta **mengunggahnya (upload) ke SIM-LITABMAS antara tanggal 1-10 November 2014**.
- (5) PIHAK KEDUA berkewajiban mempertanggungjawabkan pembelanjaan dana yang telah disesuaikan dengan ketentuan pembelanjaan keuangan negara.
- (6) PIHAK KEDUA berkewajiban mengembalikan sisa dana yang dibelanjakan kepada PIHAK PERTAMA untuk selanjutnya disetorkan ke Kas Negara.

Pasal 4

PIHAK KEDUA berkewajiban untuk :

- (1) Mempresentasikan hasil penelitiannya pada seminar yang dilaksanakan oleh Direktorat Penelitian dan Pengabdian kepada Masyarakat Direktorat Jenderal Pendidikan Tinggi, Kementerian Pendidikan dan Kebudayaan Jakarta;
- (2) Mendaftarkan hasil penelitiannya untuk memperoleh HKI;
- (3) Memanfaatkan hasil penelitian untuk proses belajar mengajar dan bahan mengajar;
- (4) Mempublikasikan hasil penelitiannya ke dalam jurnal ilmiah;
- (5) Membayar/menyetorkan PPh pasal 21, PPh pasal 22, PPN pasal 23, dan PPN sesuai ketentuan yang berlaku;
- (6) Wajib menyelenggarakan dan mengikuti Seminar awal (proposal/instrumen) dan seminar akhir (hasil) baik secara sendiri-sendiri atau bersama-sama sesuai dengan jadwal pelaksanaan kegiatan penelitian.

Pasal 5

- (1) PIHAK KEDUA bertanggungjawab atas keaslian judul penelitian sebagaimana disebutkan dalam Pasal 1 Surat Perjanjian Kontrak Penelitian ini (bukan duplikat/jiplakan/plagiat) dari penelitian orang lain
- (2) PIHAK KEDUA menjamin bahwa judul penelitian tersebut bebas dari ikatan dengan pihak lain atau sedang didanai oleh pihak lain.
- (3) PIHAK KEDUA menjamin bahwa judul penelitian tersebut bukan merupakan penelitian yang SEDANG ATAU SUDAH selesai dikerjakan, baik didanai oleh pihak lain maupun oleh sendiri

- (4) PIHAK PERTAMA tidak bertanggungjawab terhadap tindakan plagiat yang dilakukan oleh PIHAK KEDUA
- (5) Apabila dikemudian hari diketahui ketidakbenaran pernyataan dalam diktum (1) s.d. (4), maka kontrak penelitian DINYATAKAN BATAL, dan PIHAK KEDUA wajib mengembalikan dana sejumlah nilai kontrak kepada kas negara.

Pasal 6

- (1) Jangka waktu pelaksanaan penelitian yang dimaksud Pasal 1 ini selama **8 (delapan)** bulan terhitung mulai **17 Maret 2014 s.d 31 Oktober 2014**, dan PIHAK KEDUA harus menyelesaikan Penelitian Fundamental yang dimaksud dalam Pasal 1 selambat-lambatnya **10 hari setelah tanggal pelaksanaan penelitian**.
- (2) PIHAK KEDUA harus menyerahkan kepada PIHAK PERTAMA berupa:
 - a. Laporan Akhir Hasil Penelitian dalam bentuk hardcopy sebanyak 6 (enam) eksemplar, dan dalam bentuk soft copy (CD dalam format “*.pdf”) sebanyak 1 (satu) keping atau 8 (delapan) eksemplar bagi yang akan menyertifikasikan dengan membayar biaya sebesar Rp 120.000,- perjudul ke Bagian Keuangan LPPM UNY.
 - b. Artikel Ilmiah dimasukkan ke Jurnal melalui Lembaga Penelitian dan Pengabdian kepada Masyarakat UNY, yang terpisah dari laporan sebanyak 2 (dua) eksemplar dan softcopy.
 - c. Upload laporan baik laporan Kemajuan Penelitian maupun laporan Hasil penelitian.
 - d. Upload laporan pertanggungjawaban Keuangan baik penggunaan dana 70% dan penggunaan dana 100%.
- (3) Laporan hasil penelitian dalam bentuk hard copy harus memenuhi ketentuan sebagai berikut:
 - a. Bentuk/ukuran kertas kuarto
 - b. Warna cover Abu-abu
 - c. Di bagian bawah kulit ditulis:
Dibiayai oleh DIPA Universitas Negeri Yogyakarta dengan Surat Perjanjian Penugasan dalam rangka Pelaksanaan Program Penelitian Desentralisasi BOPTN Skim: Fundamental Tahun Anggaran 2014 Nomor: 231/Fund.-BOPTN/UN34.21/2014 Tanggal 17 Maret 2014.
- (4) Selanjutnya laporan tersebut akan disampaikan ke:
 - a. Perpustakaan Nasional republik Indonesia Jakarta sebanyak 1 (satu) eks;
 - b. PDII LIPI Jakarta sebanyak 1 (satu) eks;
 - c. BAPPENAS c.q. Biro APKO Jakarta sebanyak 1 (satu) eks;
 - d. Perpustakaan Lembaga Penelitian dan Pengabdian kepada Masyarakat UNY sebanyak 3 (tiga) eks.
- (5) Apabila batas waktu habisnya masa penelitian ini PIHAK KEDUA belum menyerahkan Laporan Hasil Penelitian kepada PIHAK PERTAMA, maka PIHAK KEDUA dikenakan denda sebesar **1 % (satu persmil)** setiap hari keterlambatan sampai dengan setinggi-tingginya **5% (lima persen)** dari nilai **surat Perjanjian Pelaksanaan Hibah Penelitian**, terhitung dari tanggal jatuh tempo yang telah ditetapkan sampai dengan berakhirnya pembayaran dana Hibah Penelitian oleh Lembaga Penelitian dan Pengabdian kepada Masyarakat Universitas Negeri Yogyakarta yang dibuktikan dengan Surat Pernyataan/Berita Acara Keterlambatan yang disepakati/disetujui Pihak Pertama dengan mencantumkan tanggal Jatuh Tempo Penyerahan Laporan Hasil Penelitian.

Pasal 7

- (1) Apabila Ketua Peneliti sebagaimana dimaksud pasal 1 tidak dapat menyelesaikan pelaksanaan penelitian ini, maka PIHAK KEDUA wajib menunjuk pengganti Ketua Pelaksana sesuai dengan bidang ilmu yang diteliti dan merupakan salah satu anggota tim;
- (2) Apabila PIHAK KEDUA tidak dapat melaksanakan tugas sebagaimana dimaksud pada pasal 1 maka harus mengembalikan seluruh dana yang telah diterimanya kepada PIHAK PERTAMA, untuk selanjutnya disetor ke Kas Negara.

Pasal 8

Hak Kekayaan Intelektual yang dihasilkan dari pelaksanaan penelitian tersebut diatur dan dikelola sesuai dengan peraturan dan perundang-undangan yang berlaku.

Pasal 9

Hasil penelitian berupa peralatan dan / atau alat yang dibeli dari kegiatan penelitian ini adalah milik negara yang dapat dihibahkan kepada Universitas Negeri Yogyakarta atau Lembaga Pemerintah lain melalui Surat Keterangan Hibah.

Pasal 10

PIHAK PERTAMA maupun PIHAK KEDUA tidak bertanggung jawab atas keterlambatan atau tidak terlaksananya kewajiban seperti tercantum dalam kontrak sebagai akibat *Force Majeure* yang secara langsung mempengaruhi terlaksananya kontrak, antara lain: perang, perang saudara, blockade ekonomi, revolusi, pemberontakan, kekacauan, huru-hara, kerusuhan, mobilisasi, keadaan darurat, pemogokan, epidemis, kebakaran, banjir, gempa bumi, angin ribut, gangguan navigasi, tindakan pemerintah di bidang moneter.

Force Majeure di atas harus disahkan kebenarannya oleh Pejabat yang berwenang.

Pasal 11

Surat Perjanjian Internal Pelaksanaan Penelitian Fundamental ini dibuat rangkap 2 (dua), dan dibubuhi materai sesuai dengan ketentuan yang berlaku, dan biaya materainya dibebankan kepada PIHAK KEDUA.

Pasal 12

Hal-hal yang belum diatur dalam perjanjian ini akan ditentukan kemudian oleh kedua belah pihak secara musyawarah.

PIHAK KEDUA
Ketua Peneliti,



Dr. EKO MARPANAJI

PIHAK PERTAMA
Sekretaris LPPM
Universitas Negeri Yogyakarta



Dr. Widarto, M.Pd
NIP 19631230 198812 1 001

B. Artike Jurnal

EXECUTION TIME AND WORKLOAD COMPARISON BETWEEN WEIGHTED-SELECTIVE SCHEDULING AND GENETIC ALGORITHM SCHEDULING

Eko Marpanaji¹

¹Pend. Teknik Elektronika, Fakultas Teknik, Universitas Negeri Yogyakarta
Jl. Colombo No. 1, Karangmalang, Yogyakarta, 55281

Telp : (0274) 586168, Fax : (0274) 586168, website: <http://uny.ac.id>

¹eko@uny.ac.id

Abstrak

Skema penjadwalan *weighted-selective* dengan metoda paralelisme data (*data-parallelism*) atau DP-WS akan selalu memprioritaskan *node* yang memiliki kemampuan komputasi paling tinggi, sehingga beban kerja komputasi paralel menjadi tidak merata. Tujuan penelitian ini adalah mengembangkan komputasi terdistribusi untuk tugas-tugas komputasi dalam SDR dengan menggunakan metoda paralelisme tugas (*task-parallelism*) dengan optimasi penjadwalan menggunakan Algoritma Genetika (GA) atau TP-GA untuk memperbaiki skema penjadwalan DP-WS. Sistem penjadwalan TP-GA ini diharapkan menghasilkan penjadwalan yang optimal dengan pembagian beban kerja lebih merata. Kinerja penjadwalan diamati berdasarkan waktu eksekusi total penyelesaian setiap jadwal serta waktu idle sebagai indikator pemerataan beban kerja. Makalah ini membahas hasil penelitian khususnya dalam kajian tentang analisis waktu komputasi dan beban kerja antara skema penjadwalan DP-WS dengan TP-GA serta formula yang dapat digunakan untuk menentukan fungsi fitness untuk keperluan penjadwalan tugas-tugas komputasi *software-defined radio* menggunakan algoritma genetika. Hasil penelitian ini diharapkan dapat menjadikan konsep dasar dalam mengembangkan komputasi terdistribusi pengolahan sinyal digital frekuensi tinggi yang memiliki laju bit sangat tinggi sehingga dapat digunakan untuk pengembangan softradar, softtv ataupun softradar.

Kata kunci : *algoritma genetika, komputasi terdistribusi, paralelisme tugas, penjadwalan, paralelisme data, software-define radio, waktu eksekusi, beban kerja komputasi.*

Abstract

The weighted-selective scheduling scheme with data parallelism method or DP-WS always prioritized the node that had the highest computational capability. Hence, the parallel computing workload became uneven. The purpose of this research was to develop the distributed computation for SDR computational tasks by using the task parallelism method with scheduling optimization using Genetic Algorithm (GA) or TP-GA to improve the DP-WS scheduling scheme. This TP-GA scheduling system is expected to produce optimal scheduling by sharing the workload more evenly. The scheduling performance was observed by the total execution time of each schedule completion and the idle time as an indicator of workload equalization. This paper discussed the results of the research, especially in the studies of the computing time analysis and workload between DP-WS and TP-GA as well as the formula that can be used to determine the fitness function for software-defined radio computing tasks scheduling using genetic algorithm. The results of this study were expected to be the basic concepts in developing the distributed computation of the high-frequency digital signal process that has a very high bit rate. Those can be used for the softradar, softtv or softradar development.

Keywords: *genetic algorithms, distributed computation, task- parallelism, scheduling, data parallelism, software-define radio, execution time, computational workload.*

1. Introduction

The main obstacle in realizing a digital signal processing system of high frequency (radio signal, tv, and radar) is the computation for processing a very high bit rate, especially in developing radio software (sofradio), television software (softtv), and radar software (sofradar), or other high-frequency digital systems that require real-time computational processes. The sofradio, softtv, and sofradar here is a radio / tv / radar which function of the system is determined by the software that works on hardware used. This method is expected to increase the flexibility of the system, because the function of the system is determined by the software that can be changed any time without changing the hardware architecture used. The term Software-Defined Radio was first proposed by J. Mitola [12]. The hardware used in developing SDR can use a minimum system [2] or Personal Computer (PC) [1]. Distributed computation method can be used to address the fairly large computing problem so that the execution time becomes shorter. A major issue in parallel computation or distributed computation is the scheduling of computational tasks division to the machines that run the computational process. The desired scheduling is an optimal scheduling which the total execution time is shorter and the workload of computing resources is more evenly.

Digital signal processing by applying computing parallel with data parallelism method and implementing "weighted-selective scheduling scheme" for parallel computational scheduling of Software-Defined Radio (SDR) in a PC cluster network is an improvement of round robin scheduling scheme [13]. Parallel computational method using parallelism data is that all of computing nodes perform the same task for different data packets. The weighted-selective scheduling scheme will always prioritize the node that has the highest computational capability for each data packet to be processed. Thus, there is the possibility of several computing nodes not to perform at all. It is because the lowest value of the priority will result to the inequality of distributed computing workload.

This paper will describe the results of studies on parallel computation scheduling and parallelism method towards execution time total and all machines workload which run parallel computing process.

This research was conducted in order to improve the weakness of WS-DP scheduling system (Weighted-Selective with Data parallelism) towards the computing tasks in SDR. The proposing model is scheduling scheme with Genetic Algorithm for optimizing the distributed computational scheduling, and applying task parallelism method in the partition of computing tasks. Furthermore, the proposed scheduling scheme called TP-GA (Task parallelism with Genetic Algorithm). This study is a continuation of Software-Define Radio (SDR) studies that has been done previously [3-9].

2. Parallel Computational Scheduling

The main issue in running parallel computing or distributed computing from decomposition scheduling is the task (scheduling) these tasks into each computer. Scheduling in a parallel computing system in principle is to allocate a set of tasks to each processor or a computer in ways that minimize obtained optimal computational performance.

There are two types of methods used in scheduling, namely (1) list of heuristics, make a list of tasks by priority in descending rank, and the highest priority will be given to the first processor is ready to execute; (2) the meta-heuristics, known as Genetic Algorithms, perform tracking method using the random nature of evolution and natural genetics. Meta-heuristic scheduling method or methods of genetic algorithm will produce an optimal scheduling (Lee, 2003).

Scheduling algorithms can be categorized into 2 (two) types (Yue, 1996), namely:

- Static scheduling or prescheduling is performed during the compile-time. The main advantage of this approach is that there is no extra scheduling overhead during execution time (run time) since the iteration is done entirely at compile time. However, one of the problems in scheduling static is processor load imbalance. The iterations from different loop often require different execution time. Since the static scheduling gives that iteration to the processor at compile-time, the provision cannot be set to push the dynamic variation of the processor workload.
- Dynamic scheduling is the scheduling that provides iterations to the processor at the execution time. Therefore, the workload scheduling of each processor can be set. There are several kinds of dynamic scheduling algorithms which are commonly used, namely: (1) Self-scheduling (SS); (2) Chunk-scheduling (CS); (3) guided self-scheduling (GSS); (4) Factoring-scheduling (FS); and (5) Trapezoid self-scheduling (TSS).

Weighted-selective scheduling scheme used by Zheng for SDR parallel computing is a list heuristic scheduling method with data parallelism. Meanwhile, this research used the second scheduling method called the meta-heuristic or genetic algorithms to implement task parallelism.

Scheduling in multiprocessor systems or parallel computing is a non-deterministic polynomial time (NP) and categorized as NP hard problems (Abdeyazdan, 2008). Scheduling problem in multiprocessor systems or parallel computing is often called the Multiprocessor Scheduling Problem (MSP). (Tsujimura, 1997).

A genetic algorithm is reliable to seek the problems solutions which are categorized as NP-hard problem.

Thus, the scheduling with genetic algorithm will result an optimal parallel or distributed computing scheduling. Based on the fact that the genetic algorithm has been proven reliable in any problem solving related to the optimization process, genetic algorithm is potential to get global optimum point in finding the tracking of the solutions.

The complexity time of genetic algorithm expressed in big-O value for uniformly scaled problems is $O(m)$, and $O(m^2)$ is complexity time of genetic algorithm for the exponential problem (Lobo, 2000).

The following section will explain the basic concepts of genetic algorithms, the implementation of a genetic algorithm for scheduling tasks of distributed computing with task parallelism method, and the testing results of genetic algorithm scheduling completed with the discussion.

3. Results and Discussion

3.1 The Comparison between data Parallelisms method and Task Parallelism

There are several algorithms models that can be used in establishing a parallel or distributed computing system. The first is data-parallel model in which the computational task mapped into the whole of computing nodes, and each computing node runs the same function for different data. This model is often called data parallelism. The second is the task graphic model in which the computing tasks described in the form of a task graphic that shows the dependency among the tasks, and each task is distributed to the computing nodes to run computing process on the same group data. This model is known as task parallelism. There are several models of other algorithms namely, work pool model, the master-slave model, the pipeline or producer-consumer model, and hybrid model. This study is limited to discuss the first two algorithms models, namely data parallelism and task parallelism.

Zheng (2007) uses the data parallelism in running a distributed computation with weighted-selective scheduling scheme with data parallelism (WS-DP) towards the computing tasks of the SDR. The advantage of scheduling scheme is in the term of the workload partition when the computing nodes have different computing capacity (heterogeneous). It is compared to the scheduling scheme with Round Robin algorithm which is only suitable for homogeneous computing environments. However, with the use of data parallelism in a distributed computing system, the weighted-selective scheduling schemes still have limitation when the computing capacity among the computing nodes is significantly different.

If the computing capacity difference among the computing nodes is very large, there will be a possibility that the computing nodes which have the lowest computational capacity will not get to work. This is due

to the computing tasks were given to the nodes that have much higher computing capacity. The data parallelism used did not give a chance to run the task partition evenly. The nodes will run the same computation for different data comprehensively. Thus, the data parallelism model used does not ensure the equal workload.

To overcome the weakness of the data parallelism used in the distributed computing model of WS-DP, this research proposes a distributed computing system with task parallelism system with scheduling scheme using genetic algorithm (TP-GA). By using task parallelism, the scheduling has the opportunity to set the optimal schedule with task partition through the available computing nodes based on their capacities. Each node will run different computational tasks to complete processing a data. The function of genetic algorithm is to optimize the scheduling to obtain the smallest execution time with the workload partition more evenly. It is shown by idle time of each computing node as small as possible.

The finishing time analysis and the workload partition which is shown by the percentage of idle time towards the task completion time which has been made in this study between the data parallelism used by Zheng with task parallelism by using a genetic algorithm can be seen in the following numerical analysis.

Suppose the computing tasks to be completed are:

$$(a \times b) + (c \times d) \quad (1)$$

It is known that the execution time for multiplication (\times) is 3 units of time, while the execution time for summation ($+$) is 2 units of time. Here is the comparison of the execution time total of scheduling result between distributed computing using data parallelism with weighted-selective scheduling (DP-WS) with distributed computing using task parallelism with genetic algorithm (TP-GA).

If there are 2 pieces of data packets and distributed computing system consists of 3 pieces of the clients, where the client-1 has once (1x) computing capability, client-2 has twice (2x) computing capability, and client-3 has three times (3x) computing capability. So, the total time of computing process for each distributed computing process can be explained as follows.

Figure 1 shows the process of distributed computing DP-WS together with calculations of total execution time for each client as well as the idle time for each client. The nature of WS scheduling (weighted-scheduling) is always prioritized the highest computational capability node, so the first data packets of computing process is directed to the client-3 which has three times (3x) speed. By using the DP method, the client-3 will run the whole process of computation (multiplication and summation) for the first data packet.

Based on the WS-DP method, the total execution time of computing processes on the client-3 is 2.66 units of time. While for the second package, because the client-3 is working on the first data packet computation, WS will hand over the process of computing to the second data packet to the client-2. It is because the computing priority is still higher than the client-1. With the DP method, the client-2 will run the whole process of computing for the second data packets (multiplication and summation) as well, so that the total execution time of the completion schedule is 4 units of time.

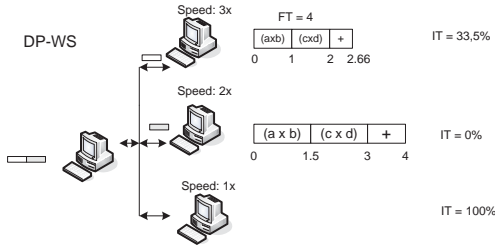


Figure 1. The execution time and idle time method of DP-WS (2 packets, 3 clients)

If the computing process stops for the two data packets, it can be seen that the client-1 did not work at all. If the idle time (IT) of a node is stated as the percentage of time that is not used by the nodes to compute the maximum total execution time of the completion schedule, the idle time for a processor $IT(p)$ can be stated as follows:

$$IT(p) = \frac{FT_{\max} - FT_p}{FT_{\max}} \times 100\% \quad (2)$$

where, $IT(p)$ is the idle time of a node p , FT_p is the total execution time required by node p in running the computation, and FT_{\max} is the maximum total execution time of the computing process.

Based on the equation above, then the idle time $IT(p)$ for client-1 is 100%, whereas the time idle of client-2 is 0%, and the time idle of client-3 is 33.5%. The workload can be said to be evenly if all nodes have relatively the same value of IT . If the IT idle time is taken its maximum value, the distributed computing process above has IT value= 100%.

Unlike the TP-GA method as shown in Figure 2 TP method will divide all the computing tasks to process a data packet. Thus, the function of scheduling is to find the most optimal schedule that produces a schedule that has the lowest total of execution time.

With 2 pieces of data packet and 3 pieces of client with computing speed as mentioned above, the total execution time of the completion schedule is $FT = 3.66$ unit time for TP-GA method (lower than DP-WS method that requires the total execution time 4 unit

time). Idle time (IT) of TP-GA method is also becoming more equitable for all computing nodes, with the greatest value is 45.3% (lower than the DP-WS method that its IT maximum is 100%).

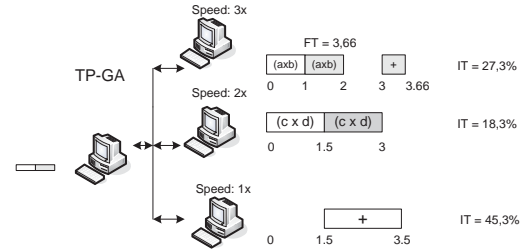


Figure 2. The execution time and idle time of TP-GA method (2 packets, 3 clients)

Furthermore, by analyzing the enumeration for the number of different packets with 2 pieces of the client, the total execution time to complete a schedule and idle time are shown in the following tables and graphs.

Table 1 The Total Execution Time (units of time) Completion of DP-WS Scheduling and TP-GA Distributed Computing for 2 clients.

Number of Data	FT		% TP/DP
	Data-Parallelism	Task-Parallelism	
1	4	4	100,00%
2	8	5,5	68,75%
3	8	8	100,00%
4	12	11	91,67%
5	16	13,5	84,38%
6	16	16	100,00%
7	20	19	95,00%
8	24	22	91,67%
9	24	24	100,00%
10	28	27	96,43%

Average:

92,789%

Table 1 shows the total execution time of a distributed computing scheduling completion of a DP-WS and TP-GA for a number of different data packets, ranging from 1 to 10 data packets of data packets with 2 clients computing nodes. The 4th column presents a comparison of the total execution time for scheduling completion of the TP-GA with DP-WS in percent. Based on this table, the percentage of TP-GA against the DP-WS never exceeds 100%. It means that the total execution time of scheduling completion of the TP-GA method is always lower or at least equal to the total execution time of scheduling completion of DP-WS method. Total execution time of the TP-GA method is

equal to the total execution time of scheduling completion of the DP-WS method when the number of data packets processed are 1, 3, 6, and 9 packages. So, the ratio of the total execution time of scheduling completion became 100%. Meanwhile, for the number of other data indicates that the total execution time of scheduling completion of the TP-GA is always lower than the total execution time of scheduling completion of the DP-WS. The average percentage ratio of the total execution time of scheduling completion is 92.789%.

While the observation of IT idle time for distributed computing using DP-WS method and TP-GA with 2 clients to vary the number of data packets processed are shown in the following table:

Table 2 The Idle Time to Show Equitable Distributed Computing Workload of DP-WS and TP-GA

Number of Data	IT (%)	
	Data-Parallelism	Task-Parallelism
1	100,00	37,50
2	50,00	9,09
3	0,00	0,00
4	33,30	10,00
5	25,00	3,70
6	0,00	16,00
7	20,00	5,26
8	16,67	4,54
9	0,00	0,00
10	14,29	1,85
Average:	25,93	8,79
Max:	100,00	37,50

Table 2 shows the IT idle time of DP-WS and TP-GA distributed computational scheduling for a number of different data packets, ranging from 1 to 10 data packets with 2 clients of computing nodes. This idle time is used as an indicator of the workload equalization through the computing nodes involved in a scheduling. Based on this table, the idle time for DP-WS and TP-GA can be explained as follows.

Maximum idle time of DP-WS distributed computing is 100%. It means that there are some computing nodes which do not run the computing at all. The average DP-WS idle time for multiple data packets with 2 clients is 25.93%.

Maximum idle time of TP-GA distributed computing is 37.5%. It means that all computing nodes get involved in the computational scheduling. Meanwhile, the average idle time of TP-GA distributed computing is 8.79%.

Based on the average value of idle time of DP-WS and TP-GA distributed computation, it can be stated that the TP-GA distributed computation has workload partition more evenly than DP-WS distributed

computation. It is stated in the average value of TP-GA idle time is 8.79% lower than the average value of DP-WS idle time 25.93%.

Figure 3 shows the relationship between the number of data packets and the maximum execution time in resolving a scheduling method for DP-WS and TP-GA distributed computing. Based on the graph, it shows that the maximum execution time of a TP-GA distributed computing system is lower than what DP-WS distributed computing system has. However, there are some points which state that the TP-GA and DP-WS maximum execution time are the same. It happens when the number of packets is 1, 3, 6, and 9 packages. Based on the graph, it can be stated that the model of distributed computing with task parallelism method using genetic algorithm scheduling scheme has lower or equal maximum execution time to the distributed computing model with data parallelism method using weighted-selective scheduling scheme.

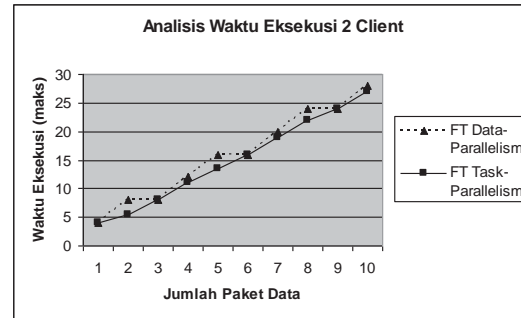


Figure 3 Finishing Time (FT) Analysis between Data Parallelism with Task Parallelism.

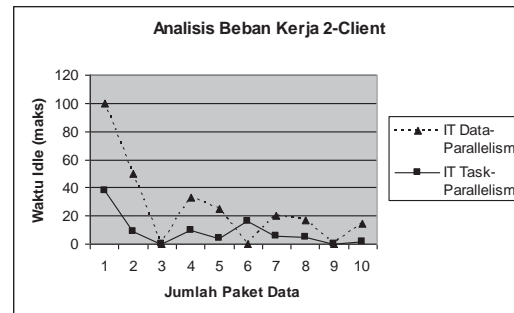


Figure 4 The workload analysis between data parallelism with Task Parallelism.

Figure 4 shows the relationship between the number of data packets with the computing node idle time of distributed computing for DP-WS and TP-GA scheduling method. Idle time is used as an indicator of workload balancing in distributed computational scheduling. Based on the graph, it shows that the idle time of TP-GA distributed computing system is lower than the idle time of DP-WS distributed computing system. However, there are some points which states

that the idle time of the TP-GA and DP-WS are same. It happens when the number of packets is 3, 6, and 9 packages.

Based on the graph, it can be stated that the model of distributed computing with task parallelism method using genetic algorithm scheduling scheme has a lower value than the idle time of the distributed computing model with data parallelism method using weighted-selective scheduling scheme. It is proved that the distributed computing with task parallelism method using genetic algorithm scheduling scheme will result to the distributed computational workload is more evenly compared to data parallelism method using weighted-selective scheduling scheme.

4. Conclusion and Future Works

4.1 Conclusion

Based on the execution time and the idle time analysis above, it can be concluded that the parallel computing or distributed computing using task parallelism method using genetic algorithm scheduling scheme has a lower execution time in task completion than data parallelism method using weighted-selective scheduling scheme. In addition, workload analysis also shows that the task parallelism method distribute the workload more evenly than the data parallelism method. It can be seen from the comparison of the idle time percentage towards the maximum completion time in executing the task. In other words, an analysis of the differences between WS-DP and TP-GA prove that the TP -GA scheduling schemes have a more evenly workload seen from the computing nodes idle time.

4.2 Future Works

The follow-up of the results of this study is more in-depth study about Genetic Algorithms used in the scheduling scheme of parallel computing / distributed computing tasks for Software-Defined Radio (SDR).

References:

- [1] Bose, V. G., 1999, *Design and Implementation of Software Radios Using a General Purpose Processors*, [Online] <http://www.sigmobility.org/phd/1999/theses/bose.pdf>.
- [2] Gweon-Do JO., 2005, *A DSP-Based Reconfigurable SDR Platform for 3G Systems*, Journal of IEICE Transaction on Communication, Vol. E88-B No.2 Feb. 2005, 678 – 686.
- [3] Marpanaji, E., Riyanto T., B., Langi., A. Z. R., dan Kurniawan, A., (2007) : Pengukuran Unjuk Kerja Modulasi GMSK pada Software-Defined Radio Platform, *Jurnal Telkomnika*, Vol. 5, No. 2 Agustus 2007, 73 – 84.
- [4] Marpanaji, E., T., B., Langi., A. Z. R., dan Kurniawan, A., (2008) : Studi Eksperimen Unjuk-Kerja Modulasi DBPSK pada Platform Software-Defined Radio (SDR), *Jurnal Technoscientia*, Vol. 1 No. 1 Agustus 2008, 14 – 22.
- [5] Marpanaji, E., Riyanto T., B., Langi., A. Z. R., dan Kurniawan, A., (2008) : Experimental Study of DQPSK Modulation on SDR Platform, *ITB Journal of Information and Communication Technology*, Vol. 1 No. 2 November 2008, 84 – 98.
- [6] Marpanaji, E., dkk. (2009) : Dekomposisi Tugas-tugas Software-Defined Radio (SDR), *Jurnal Technoscientia*, Vol. 2 No. 1 Agustus 2009, 50 – 60.
- [7] Marpanaji, E., Riyanto T., B., Langi., A. Z. R., Kurniawan, A., Mahendra, A., dan Liung, T., (2007), Experimental Study of DQPSK Modulation on SDR Platform. *Proceeding of International Conference on Rural ICT 2007*, Bandung, August, 6th, 2007.
- [8] Riyanto T., B., Marpanaji, E., Langi., A. Z. R., Kurniawan, A., Mahendra, A., dan Liung, T., (2007) : Software Architecture of Software-Defined Radio (SDR). *Proceeding of International Conference on Rural ICT 2007*, Bandung, August, 6th, 2007.
- [9] Marpanaji, E., Riyanto T., B., Langi, A. Z. R., Kurniawan, A., Mahendra, A., dan Liung, T., (2006) : Simulation and Experimental Study of GMSK Modulation on SDR Platform. *Proceeding of TSSA and WSSA 2006. The 3rd Conference on Telematics System, Services and Applications and 1st Conference on Wireless System, Service and Applications*, ITB 2006, Bandung, 8th – 9th December 2006.
- [10] Michalewics, Z. (1996) : *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, New York, pp. 45 – 55.
- [11] Rahmani, A. M., dan Vahedi, M. A. (2008) : A Novel Task Scheduling In Multiprocessor Systems With Genetic Algorithm By Using Elitism Stepping Method, *Journal of Computer Science*, Vol. 7 No. 2 June 2008. [On-line]. Available at <http://www.dcc.ufla.br/infocomp/artigos/v7.2/art08.pdf>.
- [12] Reed, J. H. (2002) : *Software Radio: A Modern Approach to Radio Engineering*, New Jersey, Prentice Hall.
- [13] Zheng, K., Li G., dan Huang L. (2007) : A weighted Scheduling Scheme in an Open Software Radio Environment, Communication, Computer and Signal Processing, PacRim 2007, *Proceedings of IEEE Pacific Rim Conference, Canada, August, 2007*, pp. 561 – 564.

C. Buku Catatan Harian Penelitian (Log Book)

BUKU CATATAN HARIAN PENELITIAN (LOGBOOK)

JUDUL PENELITIAN
APLIKASI ALGORITMA GENETIKA
PADA SKEMA PENJADWALAN KOMPUTASI TERDISTRIBUSI
(STUDI KASUS PADA *SOFTWARE-DEFINED RADIO*)

JENIS/SKIM PENELITIAN	BIDANG PENELITIAN
Penelitian Fundamental	Teknik Elektro dan Informatika

KETUA PENELITIAN		ANGGOTA
Nama	: Dr. Eko Marpanaji	1. Adi Dewanto, M. Kom
Jurusan	: Pendidikan Teknik Elektronika	2. DidikHariyanto, M. T.
Fakultas	: Teknik	

NILAI KONTRAK
Rp. 40.000.000,00

HASIL/SASARAN AKHIR TAHUN 2014

1. Simulasi Matlab Skema Penjadwalan Komputasi Paralel menggunakan Algoritma Genetika.
2. Metoda Penjadwalan menggunakan Algoritma Genetika dengan metoda Paralelisme Tugas untuk Komputasi SDR
3. Prototipe Skema Penjadwalan menggunakan Algoritma Genetika dengan metoda Paralelisme Tugas untuk Komputasi SDR
4. Makalah seminar atau artikel jurnal.

CATATAN KEMAJUAN/PELAKSANAAN PENELITIAN

No.	Tanggal *)	Kegiatan/Aktivita	Catatan Kemajuan/Hasil Aktivitas**)
1.	17 – 20 Maret 2014	Studi pustaka konsep komputasi paralel dan komputasi terdistribusi, serta metoda paralelisme	Konsep komputasi paralel dan komputasi terdistribusi, perbedaan paralelisme tugas dan paralelisma data dalam komputasi paralel
2.	20 – 25 Maret 2014	Studi Pustaka Komputasi SDR dan platform SDR	Perkembangan platform USRP versi 2 dan informasi PSoC
3.	25 – 27 Maret 2014	Persiapan pengadaan barang platform SDR	Pengadaan main board dan daughterboard USRP GNU Radio
4.	26 – 29 Maret 2014	Analisis kebutuhan Algoritma Genetika	Fungsi-fungsi yang digunakan untuk menjalankan Algoritma Genetika
5.	29 – 31 Maret 2014	Review tentang Task Graph (DAG)	Aturan dalam menggambarkan tugas-tugas komputasi dalam bentuk Grafik Tugas serta cara menentukan nilai ketinggian dan bobot komputasi
6.	1 – 3 April 2014	Studi Pustaka tentang operator genetika untuk skema penjadwalan (cross over dan mutasi) berdasarkan Grafik Tugas dalam komputasi paralel	Aturan dan algoritma crossover dan mutasi untuk keperluan penjadwalan tugas-tugas komputasi berdasarkan Grafik Tugas komputasi paralel
7.	3 – 5 April 2014	Persiapan contoh grafik tugas untuk keperluan uji coba dan pengembangan aplikasi Algoritma Genetika untuk penjadwalan komputasi paralel	Contoh Grafik Tugas dengan 8 tugas dengan nilai ketinggian dan bobot komputasi untuk masing-masing tugas yang akan digunakan sebagai acuan dalam pengembangan dan uji coba Algoritma Genetika
8.	9 – 12 April 2014	Analisis Representasi Kromosom Algoritma Genetika (GA) untuk sebuah penjadwalan berdasarkan Grafik Tugas	Format kebutuhan struktur data kromosom sebuah jadwal komputasi paralel
9.	10 – 13 April 2014	Pembuatan struktur data kromosom jadwal komputasi paralel	Struktur data kromosom jadwal komputasi paralel

No.	Tanggal *)	Kegiatan/Aktivita	Catatan Kemajuan/Hasil Aktivitas**)
10.	14 – 16 April 2014	Implementasi (<i>coding</i>) struktur data kromosom jadwal komputasi parallel	Skrip Matlab struktur data kromosom jadwal komputasi parallel
11.	17 – 20 April 2014	Simulasi dan pengujian struktur data kromosom jadwal komputasi parallel	Hasil pengujian skrip Matlab untuk fungsi struktur data kromosom penjadwalan komputasi parallel
12.	20 – 25 April 2014	Analisis Proses Inisialisasi Populasi GA dan penentuan jumlah populasi dan perancangan algoritma Inisialisasi Populasi untuk penjadwalan komputasi parallel	Algoritma dan diagram alir proses inisialisasi populasi penjadwalan komputasi parallel
13.	24 – 27 April 2014	Implementasi (<i>coding</i>) fungsi Inisialisasi populasi penjadwalan komputasi parallel	Skrip Matlab untuk fungsi inisialisasi populasi penjadwalan komputasi parallel
14.	27 – 28 April 2013	Simulasi dan pengujian fungsi Inisialisasi Populasi penjadwalan komputasi parallel	Hasil pengujian simulasi fungsi Inisialisasi Populasi penjadwalan komputasi parallel
	29 – 30 April 2014	Penentuan fungsi obyektif penjadwalan komputasi parallel untuk perhitungan fitness	Fungsi obyektif dan fungsi fitness penjadwalan komputasi parallel
15.	1 – 2 Mei 2014	Analisis Proses Perhitungan <i>Fitness</i> dan perancangan algoritma perhitungan <i>fitness</i> sebuah jadwal komputasi parallel	Persamaan fungsi obyektif dan algoritma perhitungan fitness jadwal komputasi parallel
16.	3 – 4 Mei 2014	Implementasi (<i>coding</i>) fungsi perhitungan nilai <i>fitness</i> berdasarkan fungsi obyektif penjadwalan komputasi parallel	Skrip Matlab fungsi fitness penjadwalan komputasi parallel
17.	5 - 7 Mei 2014	Simulasi dan pengujian fungsi perhitungan <i>fitness</i> penjadwalan komputasi parallel	Hasil pengujian skrip fungsi perhitungan fitness penjadwalan komputasi parallel
18.	6 – 7 Mei 2014	Analisis Proses Seleksi GA dan perancangan algoritma untuk penjadwalan komputasi parallel	Algoritma dan diagram alir fungsi seleksi GA untuk penjadwalan komputasi parallel
19.	7 – 8 Mei 2014	Perancangan algoritma dan diagram alir perhitungan probabilitas tiap individu berdasarkan nilai <i>fitness</i> untuk penjadwalan komputasi parallel	Algoritma perhitungan probabilitas individu untuk penjadwalan komputasi parallel menggunakan GA

No.	Tanggal *)	Kegiatan/Aktivita	Catatan Kemajuan/Hasil Aktivitas**)
20.	8 – 10 Mei 2014	Implementasi (<i>coding</i>) perhitungan probabilitas tiap individu berdasarkan nilai <i>fitness</i> untuk penjadwalan komputasi paralel	Skrip Matlab perhitungan probabilitas untuk penjadwalan komputasi paralel
21.	10 – 11 Mei 2014	Simulasi dan pengujian fungsi penentuan probabilitas individu dan fungsi pengurutan individu berdasarkan nilai probabilitas untuk penjadwalan komputasi paralel	Hasil pengujian perhitungan fitness, perhitungan probabilitas dan pengurutan fitness untuk penjadwalan komputasi paralel
22.	11 – 12 Mei 2014	Studi pustaka Proses Seleksi Metoda <i>Roulette Wheel</i> dan <i>Tournament Selection</i> untuk penjadwalan komputasi paralel	Konsep metoda seleksi GA untuk penjadwalan komputasi paralel
23.	12 Mei 2014	Analisis proses seleksi dengan metoda <i>Roulette Wheel</i> untuk penjadwalan komputasi paralel	Konsep mekanisme seleksi menggunakan <i>Roulette Wheel</i> untuk penjadwalan komputasi paralel
24.	13 Mei 2014	Perancangan algoritma dan diagram alir proses seleksi populasi dengan metoda <i>Roulette Wheel</i> untuk penjadwalan komputasi paralel	Algoritma dan diagram alir seleksi menggunakan metoda <i>Roulette Wheel</i> untuk penjadwalan komputasi paralel
25.	13 – 15 Mei 2014	Implementasi (<i>coding</i>) fungsi seleksi individu berdasarkan nilai probabilitas dengan metoda <i>Roulette Wheel</i> untuk penjadwalan komputasi paralel	Skrip Matlab proses seleksi untuk penjadwalan komputasi paralel
26.	15 – 17 Mei 2014	Simulasi dan pengujian proses seleksi untuk penjadwalan komputasi paralel	Hasil pengujian simulasi fungsi seleksi untuk penjadwalan komputasi paralel
27.	16 Mei 2014	Analisis proses Kawin Silang (<i>Crossover</i>) dan nilai laju Kawin Silang (<i>Crossover rate</i>) untuk penjadwalan komputasi paralel	Konsep kawin silang dan fungsi nilai laju kawin silang untuk penjadwalan komputasi paralel
28.	17 – 18 Mei 2014	Perancangan algoritma dan diagram alir proses Kawin Silang (<i>Crossover</i>) untuk penjadwalan komputasi paralel	Algoritma dan diagram alir fungsi kawin silang untuk penjadwalan komputasi paralel
29.	18 – 20 Mei 2014	Implementasi (<i>coding</i>) fungsi Kawin Silang (<i>Crossover</i>) untuk penjadwalan komputasi paralel	Skrip Matlab fungsi kawin silang untuk penjadwalan komputasi paralel

No.	Tanggal *)	Kegiatan/Aktivita	Catatan Kemajuan/Hasil Aktivitas**)
30.	20 Mei 2014	Simulasi dan pengujian proses Kawin Silang (<i>Crossover</i>) untuk penjadwalan komputasi parallel	Hasil pengujian simulasi kawin silang untuk penjadwalan komputasi parallel
31.	21 Mei 2014	Analisis proses Mutasi dan nilai laju mutasi (<i>Mutation rate</i>) untuk penjadwalan komputasi parallel	Konsep proses mutasi dan fungsi nilai laju mutasi untuk penjadwalan komputasi parallel
32.	21 – 22 Mei 2014	Perancangan algoritma dan diagram alir proses Mutasi untuk penjadwalan komputasi parallel	Algoritma dan diagram fungsi mutasi untuk penjadwalan komputasi parallel
33.	22 – 23 Mei 2014	Implementasi (<i>coding</i>) fungsi Mutasi untuk penjadwalan komputasi parallel	Skrip Matlab fungsi mutasi untuk penjadwalan komputasi parallel
34.	23 Mei 2014	Simulasi dan pengujian proses Mutasi untuk penjadwalan komputasi parallel	Hasil pengujian simulasi mutasi untuk penjadwalan komputasi parallel
35.	24 – 25 Mei 2014	Analisis proses Terminasi untuk penjadwalan komputasi parallel	Konsep terminasi untuk penjadwalan komputasi parallel
36.	25 Mei 2014	Perancangan algoritma dan diagram alir proses Terminasi untuk penjadwalan komputasi parallel	Algoritma dan diagram alir fungsi terminasi untuk penjadwalan komputasi parallel
37.	25 – 28 Mei 2014	Implementasi (<i>coding</i>) fungsi Terminasi untuk penjadwalan komputasi parallel	Skrip Matlab fungsi terminasi untuk penjadwalan komputasi parallel
39.	29 – 30 Mei 2014	Simulasi dan pengujian proses Terminasi untuk penjadwalan komputasi parallel	Hasil pengujian simulasi fungsi terminasi untuk penjadwalan komputasi parallel
41.	30 – 31 Mei 2014	Simulasi dan pengujian Algoritma Genetika terintegrasi untuk penjadwalan komputasi parallel	Hasil pengujian simulasi Algoritma Genetika secara umum untuk penjadwalan komputasi parallel
42.	1 – 3 Juni 2014	Analisis hasil pengujian Algoritma Genetika untuk penjadwalan komputasi parallel	Kemungkinan-kemungkinan perbaikan skrip Matlab Algoritma Genetika untuk penjadwalan komputasi parallel
	3 – 15 Juni 2014	Implementasi Skema Penjadwalan Algoritma Genetika untuk Komputasi Parallel menggunakan JAVA	Skrip JAVA untuk prototipe Skema Penjadwalan Komputasi parallel

No.	Tanggal *)	Kegiatan/Aktivita	Catatan Kemajuan/Hasil Aktivitas**)
45.	16-17 Juni 2014	Simulasi dan pengujian Parsial fungsi-fungsi Skema Penjadwalan Menggunakan Algoritma Genetika untuk tugas-tugas komputasi SDR	Hasil pengujian tiap-tiap fungsi yang digunakan dalam prototipe Skema Penjadwalan Komputasi Paralel menggunakan Algoritma Genetika
46.	18 – 20 Juni 2014	Simulasi dan Pengujian Terintegrasi Prototipe Skema Penjadwalan Komputasi Paralel Menggunakan Algoritma Genetika untuk Tugas-tugas Komputasi SDR	Hasil pengujian secara keseluruhan fungsi dalam prototipe Skema Penjadwalan Komputasi Paralel menggunakan Algoritma Genetika
47.	13 – 25 Juni 2014	Penyusunan bahan publikasi (makalah/artikel)	Draft makalah seminar/artikel jurnal
48.	15 – 30 Juni 2014	Pembuatan Laporan Kemajuan dan Laporan Penggunaan Dana	Draft Laporan Kemajuan dan Laporan Penggunaan Dana
49.	30 Juni 2014	Penyerahan laporan kemajuan	Laporan Kemajuan dan Laporan Penggunaan Dana
50.	1 – 7 Juli 2014	Penyusunan Draft Artikel Jurnal	Draft Artikel Jurnal
51.	5 – 20 Juli 2014	Browsing Jurnal Online Internasional dan Nasional Akreditasi	Situs jurnal online yang akan dituju dalam mengirimkan artikel jurnal
52.	2 -20 Agustus 2014	Penyusunan Artikel Jurnal dalam bahasa Inggris	Tersedia artikel jurnal dalam bahasa inggris
53.	10 September 2014	Mendaftar sebagai Author di Jurnal Telkomnika UAD	Akun author di www.telkomnika.uad.ac.id
54.	12 – 15 September 2014	Penyusunan Draft Laporan Akhir	Tersedia Draft Laporan Akhir
55.	16 September 2014	Monev Eksternal	Mengikuti Monev Eksternal dan memperoleh beberapa masukan dari Reviewer
56.	17 – 20 September 2014	Revisi Draft Laporan Akhir	Draft Laporan Akhir yang sudah di revisi
57.	20 – 30 September 2014	Revisi artikel jurnal	Artikel jurnal yang sudah direvisi dan siap di unggah ke jurnal online
58.	24 Oktober 2014	Seminar Hasil	Draft Laporan Akhir dan beberapa masukan oleh Reviewer
59.	25 – 28 Oktober 2014	Revisi Draft Laporan Akhir	Tersusun Laporan Akhir (Laporan Akademik dan Laporan Penggunaan Dana)

No.	Tanggal *)	Kegiatan/Aktivita	Catatan Kemajuan/Hasil Aktivitas**))
60.	30 Oktober 2014	Mengunggah Laporan Akhir	Terunggah Laporan Akademik dan Laporan Penggunaan Dana
61.	30 Oktober 2014	Pengumpulan Laporan Akhir	Terkumpul Laporan Akhir dan Laporan Penggunaan Dana

Notasi:

*) jika perlu diisiikan pula jam

**) Berisi data yang diperoleh, keterangan data, sketsa, gambar, analisis singkat, dsb.

Tambahan halaman ini sesuai kebutuhan

Pemonitor

Ketua Peneliti

.....
NIP.

Dr. Eko Marpanaji
NIP.19670608 199303 1 001

D. Seminar Instrumen



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS NEGERI YOGYAKARTA
LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT
Alamat : Kampus Karangmalang Yogyakarta 55281
Telepon: (0274) 550839 Fax. (0274) 518617
Web: lppm.uny.ac.id, e-mail: lppmuny@uny.ac.id, lppm.uny@gmail.com

BERITA ACARA PELAKSANAAN SEMINAR PROPOSAL/INSTRUMEN PENELITIAN

1. Nama Peneliti : Eko Marpanis
2. Jurusan/Prodi :
3. Fakultas : Teknik
4. Jenis penelitian : Penelitian Fundamental
5. Judul Penelitian : Aplikasi Algoritma Genetika pada Sistem Pengolahan Komputer Terdistribusi (Studi Kasus pada Software Defined Radio)
6. Pelaksanaan : Tanggal 03. Mei. 2019 Jam 07.30 - 11.00
7. Tempat : Eko Marpanis
8. Dipimpin oleh : Ketua Eko Marpanis
Sekretaris Triatmanto
9. Peserta yang hadir : a. Reviewer : orang
b. Notulis : orang
c. Peserta lain : orang
Jumlah : orang

SARAN-SARAN DARI PESERTA

1. Yakinkan dan sesuaikan aplikasi
2. Berikan input = metode penelitian
3. Per pengisian nama untuk yg genetika dan logika

10. Hasil Seminar;

Setelah mempertimbangkan penyajian, penjelasan, argumentasi serta sistematika dan tata tulis, seminar berkesimpulan:

- a. Diterima, tanpa revisi/pembenahan usulan/instrumen/hasil
- b. Diterima, dengan revisi/pembenahan
- c. Dibenahi untuk diseminarkan ulang

Ketua Sidang

[Signature]
NIP:

Reviewer

[Signature]
NIP: 197002201995021001

Notulis


[Signature]
NIP: 19651231991011001

**LEMBAR PENILAIAN
KESIAPAN PELAKSANAAN PENELITIAN FUNDAMENTAL UNY**

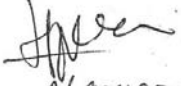
1. Nama Peneliti : Elio Marpanqaji
 2. Jurusan/Prodi/Fakultas : FT
 3. Jenis penelitian : Penelt. Fundamental

No.	Kriteria	Komentar/Saran-saran
1	Langkah-langkah pelaksanaan penelitian : Kejelasan dan kelengkapan	
2	Prototipe produk penelitian : kejelasan, keunikan dan kebaruan	
3	Instrumen penelitian yang digunakan : Kelengkapan	
4	Persiapan memasuki lapangan penelitian	
5	Kelayakan : Biaya, peralatan dan waktu	
6	Kemungkinan penelitian ini dapat diselesaikan	
7	Kesungguhan/keseriusan peneliti dalam penyiapan penelitian	

Divalidasi dan disahkan oleh
Ketua LPPM,


 Prof. Dr. Anik Ghufro
 NIP. 19621111 198803 1 001

Yogyakarta, 03-5-2014
 Reviewer,


 Agus Alaman
 NIP. 19700828 1995021 001

SARAN-SARAN DARI REVIEWER SECARA KESELURUHAN:

- Uraian penelitian berupa jurnal nasional / internasional
& teori / model.
 - Metode penelitian dijelaskan.











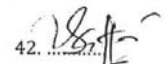
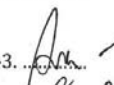
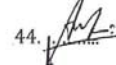

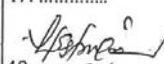


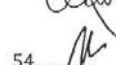



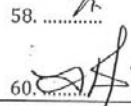


KEMENTERIAN PENDIDIKAN NASIONAL
UNIVERSITAS NEGERI YOGYAKARTA
LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT
Alamat : Karangmalang, Yogyakarta 55281
Telp. (0274) 550839. Fax. (0274) 550839, e-mail: lppm@uny.ac.id

DAFTAR HADIR SEMINAR INSTRUMEN/PROPOSAL PENELITIAN

Jenis Seminar :
Hari, Tanggal : Sabtu, 3 Mei 2014
Waktu : 08.00 WIB - Selesai
Tempat : Ruang Sidang LPPM UNY

No.	N A M A	Fakultas/Dari	TANDA TANGAN
1	Dr. Heru Kuswanto	FMIPA	1.
2	Saptomo, M.Hum	FBS	2.
3	Dr. Hari Sutrisno	FMIPA	3.
4	Dr. Ariswan	FMIPA	4.
5	Prof. Dr. Nurfini Aznam, SU, Apt	FMIPA	5.
6	Prof. Dr. Sri Atun	FMIPA	6.
7	Jaslin Ikhsan, Ph.D	FMIPA	7.
8	Siti Sulastri, M.S	FMIPA	8.
9	Dr. Hartono	FMIPA	9.
10	Kus Prihantoso K, M.Si	FMIPA	10.
11	Karyati, M.Si	FMIPA	11.
12	Dr. Dhoriva Urwatul Wutsqa	FMIPA	12.
13	Prof. Dr. AK. Prodjosantoso	FMIPA	13.
14	Prof. Dr. Jumadi	FMIPA	14.
15	Dr. Insih Wilujeng	FMIPA	15.
16	Dr. Muhsinatun Siasah M	FMIPA	16.
17	Suwardi, M.Si	FMIPA	17.
18	Retno Arianingrum, M.Si	FMIPA	18.
19	Aloysius Suyitno, M.S	FMIPA	19.
20	Suhartini, M.S	FMIPA	20.
21	Crys Fajar Partana, M.Si	FMIPA	21.
22	Prof. Dr. Yoyon Suryono	FIP	22.
23	Dr. Suparno	FIP	23.
24	S. Wisni Septiarti, M.Pd	FIP	24.
25	Nur Djazifah ER, M.Si	FIP	25.

No.	N A M A	Fakultas/Dari	TANDA TANGAN
26	Pujiriyanto, M.Pd	FIP	26. 
27	Dr. Mumpuniarti	FIP	27. 
28	Dr. Sari Rudiwati, M.Pd	FIP	28. 
29	Dr. Sugito	FIP	29.
30	Dr. Puji Yanti Fauziah	FIP	30.
31	Dr. Pratiwi Pujiastuti	FIP	31. 
32	Ikhlasul Ardi Nugroho, M.Pd	FIP	32.
33	Dr. Sri Winarni	FIK	33. 
34	Dr. Dimyati	FIK	34.
35	Prof. Dr. Wawan S. Suherman	FIK	35. 
36	Soni Nopemberi, M.Pd	FIK	36. 
37	Agus Susworo Dwi M, M.Pd	FIK	37. 
38	Dr. Sugeng Purwanto	FIK	38. 
39	Ermawan Susanto, M.Pd	FIK	39.
40	Dr. Siswantoyo	FIK	40. 
41	Nanang Primasoni, M.Or	FIK	41.
42	Yeni Artanti, M.Hum	FBS	42. 
43	Rumi Wiharsih, M.Pd	FBS	43. 
44	Suwarda Zebua, M.Sn	FBS	44. 
45	Else Liliani, M.Hum	FBS	45.
46	Prof. Dr. Haryadi	FBS	46. 
47	Dr. Tadkirioatun Musfiroh	FBS	47.
48	Sri Harti Widyastuti, M.Hum	FBS	48. 
49	Hesti Mulyani, M.Hum	FBS	49.
50	Endang Sutiyati, M.Hum	FBS	50. 
51	Bambang Setiyo HP, M.Pd	FT	51.
52	Dr. Margana, M.Hum	FBS	52. 
53	Drs. Samsul Ma'arif, M.A	FBS	53.
54	Marwanti, M.Pd	FT	54. 
55	Agus Santoso, M.Pd	FT	55. 
56	Retna Hidayah, Ph.D	FT	56. 
57	Dr. Rita Eka Izzaty, M.Si	FT	57.
58	Zamtinah, M.Pd	FT	58. 
59	Herlambang Sigit P, M.Sc	FT	59.
60	Dr. Eko Marpanaji	FT	60. 

E. Seminar Hasil

	LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT UNIVERSITAS NEGERI YOGYAKARTA			
	BERITA ACARA SEMINAR HASIL PENELITIAN			
	No. FRM/LPPM-PNL/314	Revisi : 00	Tgl 1 September 2014	Hal 1 dari 2

1. Nama Peneliti : Dr. Eko Marpanaji
 2. Jurusan/Prodi : Pendidikan Teknik Elektronika
 3. Fakultas : Teknik
 4. Skim Penelitian : Fundamental
 5. Judul Penelitian : Aplikasi Algoritma Genetika pada Skema Penjadwalan Komputer Terdistribusi
 6. Pelaksanaan : Tanggal 24 Okt. 2014 Jam 08.00 - Selesai
 7. Tempat : Ruang Sidang LPPM, Universitas Negeri Yogyakarta
 8. Dipimpin oleh : Ketua Dik. E. Purwanto, M.Pd.
 Sekretaris Triadanto, M.S.
 9. Peserta yang hadir : a. Konsultan : orang
 b. Nara sumber : orang
 c. Reviewer : orang
 d. Peserta lain : orang
 Jumlah : orang

SARAN-SARAN

- ① Perhatikan format dan angkasakan go-sheet.
- ② Apakah format dan penelitian fundamental dan penelitian terapan?
- ③ SVPK adalah, dan penelitian untuk apa?
- ④


	LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT UNIVERSITAS NEGERI YOGYAKARTA		
	BERITA ACARA SEMINAR HASIL PENELITIAN		
	No. FRM/LPPM-PNL/314	Revisi : 00	Tgl 1 September 2014

10. Hasil Seminar;


Setelah mempertimbangkan penyajian, penjelasan, argumentasi serta sistematika dan tata tulis, seminar berkesimpulan: hasil penelitian tersebut di atas:

- Diterima, tanpa revisi/pembenahan usulan/instrumen/hasil
- Diterima, dengan revisi/pembenahan
- Dibenahi untuk diseminarkan ulang

Ketua Sidang



Dr. Edi PURANDONO, M.Pd.
 NIP:

Mengetahui
Reviewer Internal
Penelitian

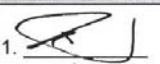

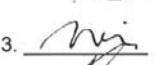
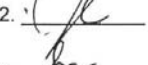


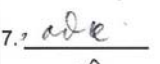
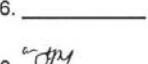

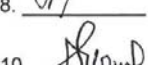



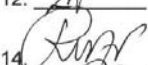




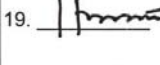

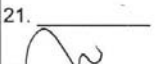






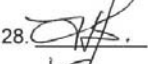
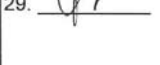


Prof. Dr. H. H. H. H.
 NIP:

Sekretaris Sidang


Triat M. M. M.
 NIP:

	LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT		
	UNIVERSITAS NEGERI YOGYAKARTA		
	FORM DAFTAR HADIR SEMINAR AKHIR		
	PENELITIAN & PPM DANA DIKTI & DANA DIPA UNY TAHUN 2014		
No. FRM/LPPM-PPM/416	Revisi : 00	Tgl: 1 September 2014	Hal 1 dari 2

Hari/tanggal : Jum'at, 24 Oktober 2014
 Waktu : pk. 08.30 WIB - selesai
 Tempat : Ruang Sidang Lt. 2 LPPM UNY

No.	Nama	Skim Penelitian	Tanda Tangan	
1	Dr. Sutrisna Wibawa, M.Pd	Hibah UPT	1. 	2. 
2	Dr. Heru Kuswanto, M.Si	Hibah UPT	3. 	4. 
3	Dr. Sri Winarni, M.Pd	Hibah UPT	5. 	6. 
4	Slamet Widodo, MT.	Hibah UPT	7. 	8. 
5	Prof. Dr. Suwarna, M.Pd	Hibah UPT	9. 	10. 
6	Dr. Mochammad Alip	Hibah UPT	11. 	12. 
7	Dr. Widyastuti Purbani	Hibah UPT	13. 	14. 
8	Dr. Paidi	Hibah UPT	15. 	16. 
9	Dr. Iis Prasetyo	Hibah UPT	17. 	18. 
10	Prof. Sugirin, Ph.D	Hibah UPT	19. 	20. 
11	Retna Hidayah, Ph.D	Hibah UPT	21. 	22. 
12	Dr. Sugito	Hibah Pasca Sarjana	23. 	24. 
13	Prof. Sugirin, Ph.D	Hibah STRANAS	25. 	26. 
14	Dr. Mami Hajaroh, M.Pd	Hibah STRANAS	27. 	28. 
15	Prof. Dr. Moerdiyanto, M.Pd	Hibah STRANAS	29. 	30. 
16	Dr. Heru Kuswanto, M.Si	Hibah STRANAS		
17	Prof. Dr. C. Asri Budiningsih	Hibah STRANAS		
18	Muhammad Munir, M.Pd	Hibah Bersaing		
19	Dr. Haryanto	Hibah Bersaing		
20	Nelva Rolina, M.Si	Hibah Bersaing		
21	Dr. Samsul Hadi, M.Pd., MT	Hibah Bersaing		
22	Dr. Dwi Rahdiyanta	Hibah Bersaing		
23	Dr. Wagiran	Hibah Bersaing		
24	Agus Murdyastomo, M.Hum	Hibah Bersaing		
25	Dr. Hartono	Hibah Fundamental		
26	Endang Sutiati, M.Hum	Hibah Fundamental		
27	Dr. Siswantoyo, M.Kes	Hibah Fundamental		
28	Dr. Eko Marpanaji	Hibah Fundamental		
29	Dr. Cahyorini K., M.Si	Hibah KLN		
30	Dr. Kastam Syamsi	Unggulan PT		

No.	Nama	Skim Penelitian	Tanda Tangan	
31	Dr. Dimiyati, M.Si	Unggulan PT	31. _____	
32	Penny Rahmawaty, M.Si	PPM PUSLIT PPK	32. _____	
33	Suranto, M.Pd., M.Si	Hibah KKN-PPM	33. _____	
34	Triatmanto, M.Si	Hibah KKN-PPM	34. _____	
35	Dr. Ariswan, M.Si. DEA	Pembahas	35. _____	
36	Prof. AK. Prodjosantoso, Ph.D	Pembahas	36. _____	
37	Prof. Dr. Herminarto Sofyan	Pembahas	37. _____	
38	Dr. Edi Purwanta, M.Pd	Pembahas	38. _____	
39	Nur Rohmah Muktiani, M.Pd	Notulis	39. _____	
40	Dr. Enny Zubaidah, M.Pd	Notulis	40. _____	
41	Prof. AK. Prodjosantoso Ph.D	Reviewer	41. _____	
42			42. _____	
43			43. _____	
44			44. _____	
45			45. _____	

Ketua LPPM,

 Prof. Dr. Anik Ghufroon, M.Pd
 NIP. 19621111 198803 1 001

