

**EFISIENSI APLIKASI PENJADWALAN MATAKULIAH JURUSAN  
TEKNIK INFORMATIKA DI UIN MAULANA MALIK IBRAHIM  
MALANG DENGAN GPU (*GRAPHIC PROCESSING UNIT*)  
DAN *MULTITHREAD***

**SKRIPSI**

Oleh :  
**DIMAS ARI SETYAWAN**  
NIM. 13650064



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2018**

**HALAMAN PENGAJUAN**

**EFISIENSI APLIKASI PENJADWALAN MATAKULIAH JURUSAN  
TEKNIK INFORMATIKA DI UIN MAULANA MALIK IBRAHIM  
MALANG DENGAN GPU (*GRAPHIC PROCESSING UNIT*) DAN  
*MULTITHREAD***

**SKRIPSI**

**Diajukan Kepada : Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan Dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh :**

**DIMAS ARI SETYAWAN**

**NIM. 13650064**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2018**

LEMBAR PERSETUJUAN

EFISIENSI APLIKASI PENJADWALAN MATAKULIAH JURUSAN  
TEKNIK INFORMATIKA DI UIN MAULANA MALIK IBRAHIM  
MALANG DENGAN GPU (*GRAPHIC PROCESSING UNIT*) DAN  
*MULTITHREAD*

SKRIPSI

Oleh :

DIMAS ARI SETYAWAN  
NIM. 13650064

Telah Diperiksa dan Disetujui Untuk Diuji

Tanggal: 12 Desember 2017

Dosen Pembimbing I



H. Fatchurrochman, M.Kom  
NIP. 19700731 200501 1 002

Dosen Pembimbing II



Khadijah F.H. Holle, M.Kom  
NIDT. 19900626 20160801 2 077

Mengetahui,

Ketua Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Cahyo Crys dian  
NIP. 19740424 200901 1 008

## LEMBAR PENGESAHAN

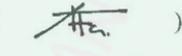
EFISIENSI APLIKASI PENJADWALAN MATAKULIAH JURUSAN  
TEKNIK INFORMATIKA DI UIN MAULANA MALIK IBRAHIM  
MALANG DENGAN GPU (*GRAPHIC PROCESSING UNIT*) DAN  
*MULTITHREAD*

## SKRIPSI

Oleh :

DIMAS ARI SETYAWAN  
NIM. 13650064

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)  
Tanggal : 5 Januari 2018

Susunan Dewan Penguji :	Tanda Tangan
Penguji Utama : <u>Ainatul Mardhiyah, M.CS</u> NIDT. 19860330 20160801 2 075	(  )
Ketua Penguji : <u>Irwan Budi Santoso, M.Kom</u> NIP. 19770103 201101 1 004	(  )
Sekretaris Penguji : <u>H. Fatchurrochman, M.Kom</u> NIP. 19700731 200501 1 002	(  )
Anggota Penguji : <u>Khadijah F.H. Holle, M.Kom</u> NIDT. 19900626 20160801 2 077	(  )

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Zahyo Crys dian  
NIP. 19740424 200901 1 008

### PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini :

Nama : Dimas Ari Setyawan

NIM : 13650064

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 8 Januari 2018

Yang Membuat Pernyataan



Dimas Ari Setyawan  
NIM. 13650064

## MOTTO

BANYAK MAIN, BANYAK MANFAAT, BANYAK PAHALA,

SEDIKIT DOSA



## KATA PENGANTAR

*Assalamu'alaikum Wr.Wb.*

Puji syukur kami haturkan kehadiran Allah S.W.T atas segala kekuatan, rahmat dan hidayahNya sehingga penulis dapat menyelesaikan skripsi dengan lancar. Shalawat serta salam semoga tetap tercurahkan kepada junjungan kita, Nabi Muhammad S.A.W yang dengan kerendahan hati beliau selalu mendoakan umat-umatnya hingga akhir zaman.

Banyak pihak yang telah membantu dan menyelesaikan penulisan dalam pembuatan skripsi ini. Untuk itu, penulis ingin menyampaikan iringan doa dan ucapan terimakasih kepada :

1. Bapak Dr. Cahyo Crysdian, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Bapak Fatchurrochman, M.Kom, sebagai dosen pembimbing I sekaligus Sekertaris Jurusan Teknik Informatika yang telah meluangkan banyak waktu untuk membimbing dan memberikan arahan kepada penulis dalam menyelesaikan skripsi hingga selesai
3. Ibu Khadijah F.H. Holle, M.Kom, sebagai dosen pembimbing II yang telah memberi banyak masukan terhadap penulisan skripsi ini
4. Ayah, Ibu dan keluarga yang selalu memberi motivasi dan doa yang tiada terputus kepada penulis

5. Bapak/Ibu Dosen Teknik Informatika yang telah sabar dan ikhlas dalam memberikan bimbingan keilmuan kepada penulis selama masa studi
6. Segenap sivitas akademika jurusan Teknik Informatika, terimakasih atas segala ilmu dan pengalaman yang berharga.
7. Imamah Adilah, yang telah berkontribusi banyak dalam proses penyelesaian skripsi.
8. Teman-teman jurusan Teknik Informatika angkatan 2013, dan kontrakan P'Jumaat yang banyak memberikan dukungan kepada penulis dan teman-teman yang lain yang tidak dapat penulis tuliskan satu persatu.

Penulis menyadari bahwa tidak ada sesuatupun yang sempurna dimuka bumi kecuali Allah S.W.T . Atas segala kekurangan dalam laporan penulisan skripsi ini, penlis memohon maaf dan mengharap kritik serta saran dari pembaca. Semoga dari kekurangan pada skripsi ini dapat lebih dikembangkan lagi oleh peneliti selanjutnya dan selalu memberikan manfaat kepada pembaca. Amiin.

*Wassalamualaikum Wr.Wb*

Malang, 8 Januari 2018

Penulis

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGAJUAN.....	ii
LEMBAR PERSETUJUAN .....	iii
LEMBAR PENGESAHAN .....	iv
PERNYATAAN KEASLIAN TULISAN .....	v
MOTTO .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR .....	xi
DAFTAR TABEL.....	xii
ABSTRAK .....	xiii
ABSTRACT.....	xiv
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	3
1.3. Tujuan.....	4
1.4. Batasan Masalah.....	4
<b>BAB II STUDI LITERATUR.....</b>	<b>5</b>
2.1. <i>Green Software</i> .....	5
2.2. <i>Green Software Engineering</i> .....	6
2.3. <i>Maximize Idle</i> (Memaksimalkan kondisi <i>idle</i> suatu program) .....	7
2.4. <i>Arraylist</i> .....	8
2.5. Komputasi Paralel .....	10
2.6. GPU.....	11
2.7. OpenCL .....	12
2.8. <i>Multithread</i> .....	13
2.9. Penjadwalan Matakuliah versi 1.0.....	16
<b>BAB III METODE PENELITIAN .....</b>	<b>20</b>
3.1. Ruang Lingkup Penelitian .....	20
3.1.1. <i>Multithread</i> .....	22
3.1.2. <i>Performance Libraries</i> .....	22
3.2. Data .....	23
3.3. Analisis Kebutuhan Non-Fungsional .....	23
3.4. Perancangan Sistem.....	24
3.4.1. Blok Masukan .....	25
3.4.2. Pencarian Jadwal.....	29
3.4.3. Validasi Konstrain.....	32
3.4.4. GPU.....	34
3.4.5. Multithread.....	42
3.4.6. Blok Keluaran .....	48
3.5. Rancangan Uji Coba.....	49
3.6. Desain Interface.....	51
<b>BAB IV UJI COBA DAN HASIL.....</b>	<b>54</b>

4.1.	Deskripsi Program .....	54
4.2.	Hasil Implementasi Sistem .....	55
4.2.1.	Hasil Implementasi Kelas Aturan .....	55
4.2.2.	Hasil Implementasi <i>Arraylist</i> .....	57
4.2.3.	Hasil Implementasi <i>Multithread</i> .....	59
4.3.	Pengujian Sistem Tahap Waktu Komputasi .....	62
4.3.1.	Uji Coba dengan GPU .....	62
4.3.2.	Uji Coba dengan <i>Multithread</i> .....	64
4.4.	Pengujian Sistem Tahap <i>Speedup</i> .....	67
4.4.1.	Perbedaan jumlah <i>thread</i> .....	67
4.4.2.	Perbedaan jumlah data jadwal matakuliah .....	68
4.5.	Pembahasan Sistem .....	69
4.5.1.	Hasil dengan Perbedaan Perangkat .....	69
4.5.2.	Pembahasan Hasil Jadwal yang Terbentuk .....	72
4.5.3.	Hasil dengan Perbedaan Jumlah Ruang .....	75
4.5.4.	Hasil <i>Speedup</i> Aplikasi paralel dengan sekuensial .....	79
4.6.	Kajian Al-Qur'an dan Hadis .....	80
BAB V PENUTUP .....		84
5.1.	Kesimpulan .....	84
5.2.	Saran .....	85
DAFTAR PUSTAKA .....		86

## DAFTAR GAMBAR

Gambar 2.1 Pembagian data Collection.....	9
Gambar 2.2. Model Multithread Many-to-Many .....	14
Gambar 2.3 Model Multithread Many-to-Many .....	15
Gambar 2.4 Model Multithread One-to-One .....	16
Gambar 2.5 Class Diagram Software Penjadwalan Versi 1.0.....	19
Gambar 3.1 Lingkup Penelitian .....	21
Gambar 3.2 Perancangan Sistem GPU.....	24
Gambar 3.3 Perancangan Sistem Multithread.....	25
Gambar 3.4 Model Pencarian.....	29
Gambar 3.5 Flowchart Proses Pencarian .....	30
Gambar 3.6 Contoh Implementasi Pencarian Slot Jadwal .....	32
Gambar 3.7 Flowchart Validasi Aturan .....	35
Gambar 3.8 Activity Diagram GPU .....	36
Gambar 3.9 Flowchart Multithread.....	42
Gambar 3.10 Activity Diagram Multithread.....	44
Gambar 3.11 Contoh Proses Pencarian Multithread .....	46
Gambar 3.12 Hasil Data Uji.....	47
Gambar 3.13 Contoh Output Jadwal Versi 1.0 .....	48
Gambar 3.14 Panel Periode Akademik .....	51
Gambar 3.15 Panel Plotting Pengajar .....	52
Gambar 3.16 Panel Prefensi Mengajar.....	52
Gambar 3.17 Panel Jadwal.....	53
Gambar 4.1 Data Ujicoba Multithread.....	60
Gambar 4.2 Interface Hasil Ujicoba Multithread.....	62
Gambar 4.3 Hasil Ujicoba GPU dengan 7 Ruangan .....	63
Gambar 4.4 Hasil Ujicoba GPU dengan 6 Ruangan.....	63
Gambar 4.5 Hasil Ujicoba GPU dengan 5 Ruangan.....	64
Gambar 4.6 Contoh Output Jadwal Multithread dengan 7 Ruangan .....	65
Gambar 4.7 Contoh Output Jadwal Multithread dengan 6 Ruangan .....	66
Gambar 4.8 Contoh Output Jadwal Multithread dengan 5 Ruangan .....	67
Gambar 4.9 Hasil Jam Perkuliahan versi 1.0 .....	74
Gambar 4.10 Hasil Jam Perkuliahan Multithread .....	74
Gambar 4.11 Data Jadwal Matakuliah GPU dan Versi 1.0.....	76
Gambar 4.12 Waktu Komputasi GPU dan Versi 1.0 .....	77
Gambar 4.13 Data Jadwal Matakuliah Multithread dan Versi 1.0.....	77
Gambar 4.14 Waktu Komputasi Multithread dan Versi 1.0 .....	78

## DAFTAR TABEL

Tabel 2.1 Data Arraylist .....	10
Tabel 2.2 Perbandingan Waktu Eksekusi Program dari Beberapa Database.....	17
Tabel 3.1 Contoh Data Plot Mengajar.....	45
Tabel 3.2 Contoh Data Preferensi Mengajar.....	45
Tabel 3.3 Catatan Waktu Proses Komputasi.....	49
Tabel 4.1 Ujicoba Multithread dengan 7 Ruangan .....	65
Tabel 4.2 Ujicoba Multithread dengan 6 Ruangan .....	65
Tabel 4.3 Ujicoba Multithread dengan 5 Ruangan .....	66
Tabel 4.4 Hasil Ujicoba dengan Perbedaan Jumlah Thread .....	68
Tabel 4.5 Hasil Ujicoba Multithread dengan Perbedaan Komputasi.....	68
Tabel 4.6 Data Ujicoba dengan Laptop.....	70
Tabel 4.7 Data Ujicoba dengan dekstop .....	70
Tabel 4.8 Perbandingan Konstrain Preferensi Dosen .....	72
Tabel 4.9 Perbandingan Konstrain Ruang .....	73
Tabel 4.10 Hasil Perbandingan Konstrain Bentrok Dosen .....	75
Tabel 4.11 Hasil Speedup Aplikasi Versi 1.0 dengan Multithread.....	79
Tabel 4.12 Hasil Speedup Aplikasi Versi 1.0 Arraylist dengan Multithread.....	79

## ABSTRAK

Setyawan, Dimas Ari. 2018. **Efisiensi Aplikasi Penjadwalan Matakuliah Jurusan Teknik Informatika di UIN Maulana Malik Ibrahim Malang dengan GPU (*Graphic Processing Unit*) dan *Multithread***. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing (I) Fatchurrochman, M. Kom. (II) Khodijah Fahmi H. H., M.kom

---

---

**Kata Kunci :** *Green Software Engineering*, Efisiensi, Komputasi Paralel, GPU, *Multithread*.

Proses efisiensi komputasi perangkat lunak merupakan salah satu perwujudan *green software engineering*. Efisiensi komputasi dapat dilakukan dengan berbagai cara, diantaranya dengan melakukan komputasi secara paralel yang dapat diwujudkan dengan GPU dan *multithread*. Objek pada penelitian ini adalah aplikasi penjadwalan perkuliahan versi 1.0 yang dalam proses komputasinya membutuhkan waktu sekitar 13 menit. Konsep komputasi paralel dengan GPU pada aplikasi versi 1.0 membutuhkan waktu yang semakin lama dikarenakan kurang tepatnya penempatan konsep paralel dan pembagian akses memori di GPU. Pada tahap selanjutnya dilakukan proses komputasi paralel dengan *multithread*, sehingga waktu yang dihasilkan dalam proses komputasi sekitar 3 detik dan susunan urutan jadwal terbentuk secara *random*. Hasil *speedup* komputasi paralel 3 kali lebih cepat dibandingkan dengan aplikasi yang sekuensial, hal ini menunjukkan konsep komputasi paralel dengan *multithread* lebih cepat dibandingkan dengan sekuensial versi 1.0.

## ABSTRACT

Setyawan, Dimas Ari. 2018. **Application Efficiency Course Scheduling Informatics Department at UIN Maulana Malik Ibrahim Malang with GPU (Grapich Processing Unit) and Multithread**. Undergraduate Thesis. Informatics Engineering Department. Faculty of Science and Technology. State Islamic University of Maulana Malik Ibrahim Malang.

Advisors : (I) Fatchurrochman, M. Kom. (II) Khodijah Fahmi H. H., M.kom

---

---

**Kata Kunci :** Green Softwere Engineering, Efficiency, Parallel Computing, GPU, Multithread.

The process of software computing efficiency is one of the embodiment of green software engineering. Computing efficiency can be done in various ways, such as by computing in parallel that can be realized with GPU and multithread. The object of this research is the application of lecture scheduling version 1.0 which in the computation process takes about 13 minutes. The concept of parallel computing with GPUs in version 1.0 applications takes a long time due to the improper placement of parallel concepts and the sharing of memory access in the GPU. In the next stage is done parallel computing process with multithread, so the time generated in the computation process about 3 seconds and the order of the sequence is formed randomly. The result of parallel computing speedup 3 times faster than sequential application, it shows the concept of parallel computing with multithread faster than sequential version 1.0

## مستخلص البحث

ديماس آري ستياوان 2017.فعالية استمارة تخطيط المادة ب GPU (Grapich Processing Unit) و Multithread في قسم تكنيك معلوماتية في جامعة مولانا مالك إبراهيم الإسلامية الحكومية بمالانق . بحث جامعي. قسم تكنيك معلوماتية ، كلية العلوم و التكنولوجيا ، جامعة مولانا مالك إبراهيم الإسلامية الحكومية بمالانق. المشرف الأول: فتح الرحمان الماجستير. المشرفة الثانية : خديجة فهمي الماجستير

الكلمات الأساسية: Green Softwere Engineering، فعالية، حسابية متوازية، GPU ، maximize ، idle ، Multithread.

عملية فعالية حسابية البرمجيات من شكل تحقيق Green Softwere Engineering. تحصل فعالية حسابية بكيفيات كثيرة، منها حسابية متوازا ب GPU و Multithread. و الموضوع لهذا البحث استمارة تخطيط المادة صيغة 1.0 التي عملية حسابيتها حول 13 دقائق. و تصوّر حسابية متوازية ب GPU بصيغة 1.0 خائب بسبب قليل استطلاع الباحث. و الوقت في عملية حسابيتها أكثر بسبب خطأ تصوّر المتواز و تقسيم الاستفادة الذاكرة في GPU . و في دور التالي عملية حسابية متوازية ب Multithread ، فالوقت الذي تحتاجه عملية حسابية حول 3 ثانياً. و ينظّم ترتيب الجدول جزافاً. و استعمالة أسرع ثلاثة أضعاف باستمارة sekuensial، فلذلك نعرف أنّ تصوّر حسابية متوازية ب Multithread أسرع باستمارة sekuensial.

## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang

Semakin maju teknologi berdampak negatif terhadap keadaan bumi. Tindakan manusia yang menggunakan teknologi secara tidak bijaksana mengakibatkan bumi rusak dan banyak polusi (Pratama, 2015). Mulai dari polusi udara, polusi air, polusi tanah, radiasi, perubahan iklim yang drastis, dan pencairan es di kutub yang mengakibatkan tinggi air laut bertambah. Pemakaian sumber daya (*resource*) secara berlebihan selain berdampak pada alam, juga mempercepat habisnya sumber daya tersebut terutama bahan bakar fosil yang tidak dapat diperbarui (minyak bumi, gas alam, batu bara). Mengakibatkan pemborosan pengeluaran untuk pemenuhan sumber daya. Data dari badan dunia bersifat non profit untuk kelestarian lingkungan di dunia, yaitu Greenpeace, menyebutkan bahwa layanan dan pemanfaatan Teknologi Informasi berperan terhadap kerusakan lingkungan dikarenakan tingginya kadar emisi karbon yang dilepaskan di lingkungan.

Teknologi yang berkembang saat ini dituntut untuk lebih ramah lingkungan dikarenakan banyaknya dampak negatif yang diakibatkan. Menurut (Agarwal, 2012) teknologi ramah lingkungan adalah teknologi yang prosesnya cepat serta membutuhkan energi yang sedikit. Semakin cepat proses *software* dijalankan maka waktu yang dibutuhkan akan lebih sedikit, dan efisien. Hal inilah yang mengharuskan sebuah aplikasi bisa menyelesaikan masalah secara efisien. Ada

beberapa faktor yang mempengaruhi tingkat efisiensi seperti kecepatan mesin, sistem operasi, bahasa pemrograman dan ukuran input. Penggunaan konsep komputasi paralel dan penyimpanan data secara terorganisir juga menambah tingkat efisiensi suatu program.

Efisiensi komputasi berarti menyelesaikan pekerjaan dengan cepat, serta dapat menghemat waktu dan menghemat energi. Semakin cepat proses komputasi dan proses komputer ke keadaan *idle*, maka akan banyak energi yang bisa dihemat (Steigerwald, 2012). Proses efisiensi bertujuan untuk mempercepat proses komputasi yang dapat diukur dengan dari beberapa faktor seperti waktu, pemanfaatan sumber daya dan kesesuaian (Yuniastari, 2015).

Kecepatan komputasi suatu program bisa dipengaruhi oleh algoritma, akses data, dan banyaknya memori (Agarwal, 2012). Akses data merupakan masalah pada aplikasi penjadwalan versi 1.0. Karena data yang digunakan masih mengakses dari database langsung, sehingga membutuhkan waktu yang lama. Jenis komputasi juga mempengaruhi kecepatan suatu program. Program penjadwalan versi 1.0 masih menggunakan komputasi sekuensial, sehingga dibutuhkan perubahan dalam proses komputasi. Menurut (Mohammad Zarkasi, 2013) penggunaan komputasi paralel digunakan untuk memperoleh *speedup* dari program yang menggunakan komputasi sekuensial ke program paralel.

Proses penjadwalan secara manual memiliki banyak sekali kekurangan mulai dari waktu perancangan jadwal yang lama, data mudah hilang, belum ada arsip data, dan membutuhkan ketelitian yang tepat. Estimasi waktu dalam setiap pembuatan jadwal secara manual paling cepat 12 jam atau setengah hari dan

paling lama sampai 2 hari dan harus membutuhkan konsentrasi yang tinggi. Meski begitu, banyak jurusan yang lebih suka cara manual dalam pembuatan jadwal dikarenakan masih belum begitu yakin dan paham tentang cara kerja aplikasi penjadwalan yang telah ada. Aplikasi penjadwalan yang ada saat ini masih bersifat aplikasi interaktif, sehingga untuk proses penyusunan jadwal masih secara manual, namun arsip data sudah bisa tersimpan, sehingga waktu penyusunan jadwal sekitar 12 jam. Dalam perkembangannya aplikasi ini akan dibuat otomatis ketika merancang jadwal.

Aplikasi penjadwalan otomatis yang sudah ada saat ini berjalan dengan tiga konstrain yaitu bentrok dosen, bentrok ruang dan bentrok sholat duhur atau jum'at. Waktu yang dibutuhkan untuk melakukan komputasi penyusunan jadwal masih lama, sekitar 45 menit. Prakteknya aplikasi penjadwalan otomatis ini masih belum efisien dalam waktu komputasi. Sehingga dibutuhkan aplikasi penjadwalan yang lebih cepat untuk proses penyusunan jadwal.

Dengan dasar untuk mempercepat proses komputasi program penjadwalan matakuliah, peneliti melakukan penelitian dengan judul “**Efisiensi Aplikasi Penjadwalan dengan GPU (*Graphic Processing Unit*) dan *Multithread***” untuk membuat aplikasi penjadwalan lebih cepat dan efisien.

## **1.2. Rumusan Masalah**

- a. Bagaimana tingkat efisiensi komputasi program penjadwalan matakuliah menggunakan GPU ?

- b. Bagaimana tingkat efisiensi komputasi program penjadwalan matakuliah menggunakan *Multithread* ?

### 1.3. Tujuan

- a. Mengetahui tingkat efisiensi komputasi program penjadwalan matakuliah menggunakan GPU.
- b. Mengetahui tingkat efisiensi komputasi program penjadwalan matakuliah menggunakan *Multithread*.

### 1.4. Batasan Masalah

- a. Jadwal perkuliahan jurusan teknik informatika UIN Malang
- b. Menggunakan *software* (perangkat lunak) Netbeans dengan plugin OpenCL.
- c. Menggunakan *hardware* (perangkat keras) yang digunakan dalam penelitian ini adalah Laptop dan komputer dengan spesifikasi :
  1. Komputer dengan spesifikasi :
    - a. *Processor intel pentium dual core*
    - b. RAM 504 MB (*mega byte*)
  2. Laptop dengan spesifikasi :
    - a. Processor core i5
    - b. RAM 4 GB (*giga byte*)
    - c. NVIDIA 380

## BAB II

### STUDI LITERATUR

#### 2.1. *Green Software*

*Green software* merupakan sub bidang ilmu dari *green computing* dan termasuk pengembangan dari bidang ilmu *software engineering* atau rekayasa perangkat lunak. *Green Computing* berarti suatu tindakan yang berhubungan dengan komputasi, algoritma dan komputer yang bersifat ramah lingkungan, mengurangi polusi serta menyelamatkan bumi dari kerusakan (Pratama, 2015). Jadi *Green Software* merupakan tindakan merancang, mendesain, dan mengembangkan perangkat lunak komputer yang mampu berkolaborasi dengan perangkat keras (*hardware*) yang mendukung prinsip dan konsep dari *green computing*.

Murugesan (Murgensen, 2008) mendefinisikan *green software* sebagai *software* yang *environmentfriendly* yang dapat berteman baik dengan lingkungan tanpa merusaknya. Menurut Erdeyli (Erdelyi, 2013) bahwa *green software* adalah aplikasi yang memproduksi sedikit limbah atau sampah selama proses pengembangan dan pengoperasian. Definisi lain yang diberikan oleh Taina (Taina, 2011) tentang *Green Software*. Taina mengungkapkan bahwa *Green Software* harus memenuhi tiga kriteria berikut :

- Proses rekayasa perangkat lunak yang terdiri dari proses pengembangan, pemeliharaan dan penyelesaian sebuah software harus aman bagi lingkungan dan sumberdaya serta mengurangi sampah atau limbah

- Eksekusi *Software* harus aman dari lingkungan
- *Software* harus mendukung pengembangan yang berkelanjutan

## 2.2. Green Software Engineering

*Green software engineering* merupakan salah satu konsep dari *green software* yang terkait dengan Rekayasa Perangkat Lunak (RPL). Bisa disebut juga kegiatan untuk mengembangkan perangkat lunak menjadi green dan sustainability dan dapat menjadikan proses perangkat lunak menjadi berkelanjutan (Nauman, 2011).

Ada beberapa konsep pada *green software engineering* yaitu :

- *The Life Cycle of Software Product*

*The Life Cycle of Software Product* atau Siklus Hidup Pengembangan Perangkat Lunak dalam rekayasa sistem, sistem informasi dan rekayasa perangkat lunak, adalah proses untuk menciptakan atau mengubah sistem, dan model dan metodologi yang digunakan orang untuk mengembangkan sistem ini. Konsep umumnya mengacu pada komputer atau sistem informasi (wikibooks).

- *Sustainable Criteria And Metrics*

Kriteria berkelanjutan dan Metrik untuk produk perangkat lunak. Terutama tentang pengukuran efek umum dari produk perangkat lunak, ada model kualitas dan metrik standar. Melalui Aspek kualitas ini, software bisa direvisi. Memang model dan jumlah karakteristik yang terkait langsung dan tidak langsung efek perlu dikembangkan oleh inisiatif penelitian (Kern & dkk, 2013).

- *Procedure Models*

Model berisi model prosedur, berdasarkan jenis penggunaan yang berbeda: Pengembang, pembeli, administrator, dan pengguna. Model yang diusulkan dapat diimplementasikan untuk mendukung optimalisasi berbagai proses yang berfokus pada *green* dan rekayasa perangkat lunak berkelanjutan. Dengan cara itu, perangkat lunak Rekayasa harus menjadi *green* dan berkelanjutan produksi, dukungan, dan proses aplikasi. Karena modelnya bersifat umum, mereka dapat disesuaikan dengan konteks yang berbeda (Kern & dkk, 2013).

- *Recommendations for Action and Tools*

*Recommendations for Action and Tools* tersedia bagi berbagai pemangku kepentingan. Bagian ini terdiri dari daftar periksa, pedoman, contoh praktik terbaik, perangkat lunak, dan juga alat lainnya (seperti lembar pengumpulan data berbasis kertas). Ini mendukung pemangku kepentingan dengan keahlian profesional yang berbeda tingkat dalam menerapkan teknik *green* atau berkelanjutan pada umumnya, saat mengembangkan, membeli, mengelola, atau menggunakan perangkat lunak produk (Kern & dkk, 2013).

### 2.3. *Maximize Idle* (Memaksimalkan kondisi *idle* suatu program)

*Idle Power* merupakan mode siaga yang biasanya untuk platform *mobile* didefinisikan sebagai daya yang dikonsumsi saat sistem berjalan di ACPI ( *hibernate* atau *sleep* ) dengan aplikasi software dan layanan tetap berjalan tapi tidak aktif mengeksekusi beban kerja. Dalam keadaan ini, minimal harus ada aktivitas *background*. Dengan kata lain, prosesor tetap dalam keadaan siaga. Sehingga perangkat lunak dituntut untuk mengurangi daya siaga dengan

meningkatkan efisiensi *idle* yang akan menyebabkan peningkatan signifikan dalam daya tahan baterai (Steigerwald, 2012). Menurut Peter Larson (Larsson, 2012), untuk mencapai perangkat lunak yang efisien *idle*, dapat dilakukan beberapa hal, yakni :

- *Multithreading*

Eksekusi bisa dipercepat dengan memanfaatkan *multithreading*, sehingga bisa meningkat waktu *idle* yang pada gilirannya menyebabkan penghematan energi. Dibutuhkan proses untuk menyesuaikan *thread*, karena *thread* yang tidak seimbang dapat menyebabkan konsumsi energi meningkat.

- *Reduce Use of High-Resolution Periodic Timers*

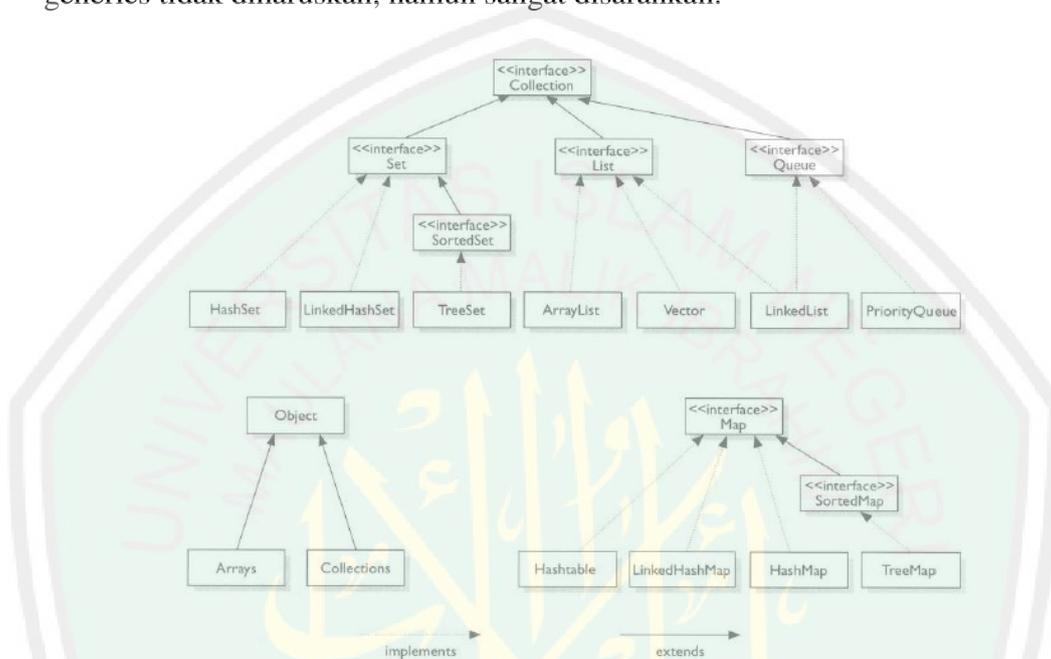
Merupakan cara yang baik untuk mengurangi jejak energi aplikasi adalah membiarkannya menganggur sesering mungkin. Pastikan Aplikasi dioptimalkan untuk menggunakan rate timer terpanjang yang mungkin saat masih memenuhi persyaratan.

#### **2.4. Arraylist**

*ArrayList* merupakan satu kelompok dengan *LinkedList* dalam kelompok *List* pada *Collection*. *Collection* adalah suatu objek yang bisa digunakan untuk menyimpan sekumpulan objek. Objek yang ada dalam *Collection* disebut elemen. *Collection* menyimpan elemen yang bertipe objek, sehingga berbagai tipe objek bisa disimpan dalam *Collection*.

Class mengenai *Collection* tergabung dalam *Java Collection Framework*. Class-class *Collection* diletakkan dalam package *java.util* dan mempunyai dua interface utama yaitu *Collection* dan *Map*. Mulai *java 1.5* (juga dikenal sebagai

J2SE 5), semua class yang termasuk *Java Collection Framework* adalah class generics. Untuk kompatibilitas dengan versi java sebelumnya, penggunaan generics tidak diharuskan, namun sangat disarankan.



**Gambar 2.1** Pembagian data *Collection*

Dapat kita lihat pada Gambar 2.1 *Collection* terbagi menjadi tiga bagian dan salah satunya adalah *List*. *List* digunakan untuk menyimpan sekumpulan objek berdasarkan urutan masuk (*ordered*) dan menerima duplikat. Cara penyimpanannya seperti array, oleh sebab itu memiliki posisi awal dan posisi akhir, menyisipkan objek pada posisi tertentu, mengakses dan menghapus isi list, dimana semua proses ini selalu didasarkan pada urutannya. Class-class yang mengimplementasikan interface *List* adalah *Vector*, *Stack*, *LinkedList* dan *Array List*.

Terdapat interface *Queue* yang cara penyimpanan seperti *List*, interface ini menyimpan objek menggunakan *methode* *FIFO* (*First In First Out*) yaitu objek yang masuk pertama keluar pertama. Class-class yang mengimplementasikan

interface *Queue* adalah *PriorityQueue* dan *LinkedList*. Data yang tersimpan pada objek *PriorityQueue* akan diurutkan, data tersebut harus mengimplementasikan objek *Comparable* atau *Comparator*.

Adapun perbedaan yang paling mendasar antara *ArrayList* dengan *Array* biasa adalah untuk menyimpan data dalam *array* biasa, maka harus mendeklarasikan jumlah elemen maksimal yang bisa menampung. Dengan kata lain jika jumlah datanya fleksibel, maka *array* tidak bisa digunakan. Sedangkan *ArrayList* dapat menampung sejumlah data secara dinamis, sehingga sebarang jumlah akan ditampung oleh *ArrayList* tanpa memperhatikan berapa jumlah maksimal elemen yang dapat ditampung, seperti table 2.1.

**Tabel 2.1 Data Arraylist**

ArrayList
+ add (index:int, element:Object) :void
+ add(o:Object) :void
+ set (index:int, element:Object) :Object
+ get(index:int) :Object
+ remove(index:int) :Object
+ size() :int

## 2.5. Komputasi Paralel

Konsep dasar dari komputasi paralel adalah mendistribusikan beban kerja antara *single* prosesor yang bekerja bersama-sama dalam melakukan komputasi. Sedangkan komputasi *sequensial* hanya dapat melakukan satu komputasi pada satu waktu. Sehingga komputasi paralel dapat melakukan banyak kegiatan pada waktu yang bersama-sama dan dapat menyelesaikan masalah sederhana dalam waktu beberapa menit saja yang seharusnya diselesaikan beberapa jam atau hari dengan menggunakan komputasi *sequensial*.

Sejak tahun 2003 terdapat 2 arsitektur dalam mengembangkan mikroprosesor yaitu, *multicore* dan *manycore* (Hwu & Krik, 2010). Arsitektur *multicore* bermula dari prosesor berinti dua dan terus bertambah hingga saat ini sudah mencapai 8 inti pada satu mikroprosesor. Arsitektur *multicore* banyak digunakan pada CPU (*Central Processing Unit*). Arsitektur ini berfokus untuk berjalan pada banyak inti, dan juga tetap menjaga performa kecepatan eksekusi program yang didesain secara sequensial. Sedangkan arsitektur *manycore* berfokus untuk memaksimalkan kinerja program yang didesain untuk bekerja secara paralel. Arsitektur *manycore* memiliki lebih banyak unit pemroses dari pada arsitektur *multicore* dan arsitektur ini biasanya digunakan pada GPU (*Graphic Processing Unit*). Perbedaan kecepatan antara GPU dan CPU disebabkan karena GPU hanya didesain untuk perhitungan yang intensif dan paralel, tetapi GPU tidak dapat menggantikan CPU sepenuhnya karena GPU hanya didesain untuk melakukan kalkulasi numerik dan pengolah graphic secara paralel.

## 2.6. GPU

GPU digunakan untuk tugas-tugas enkripsi / dekripsi, perhitungan ilmiah dan generasi mata uang *cypto* seperti Bitcoin. Karena kartu grafis yang dibangun untuk paralelisme besar, mereka dapat mengecilkan tingkat perhitungan bahkan CPU yang paling kuat untuk banyak tugas pemrosesan paralel.

GPU awalnya dikembangkan untuk mempercepat *raster* 3D (seperti *raytracing*) telah melampaui kinerja CPU untuk *raytraced* pra-diberikan grafis. Meskipun *raytracing* belum digunakan dalam game telah ada demonstrasi *real-time*. Kemajuan GPGPU berarti bahwa dalam waktu yang tidak terlalu lama kartu

grafis harus mampu memproses geometri intensif yang sama dan pencahayaan sebagai film 3D.

Banyaknya *core* yang dimiliki oleh GPU memungkinkan kecepatan komputasi bisa lebih maksimal dibandingkan dengan CPU. Kecepatan komputasi GPU 30-50 lebih cepat dibandingkan dengan CPU, tapi konsumsi *power* dari GPU juga lebih besar dibandingkan dengan CPU. Meski begitu konsumsi watt per unit Ghz *clock speed* dari GPU dan konsumsi power pada *idle state* GPU lebih kecil dibandingkan dengan CPU. Sehingga penggunaan GPU juga mendukung konsep dari *Green Software* (Krishnan, Dwivedi, & Bhagwat, 2012).

## 2.7. OpenCL

OpenCL merupakan teknologi pemrograman paralel yang bersifat *open source* dan bisa digunakan di segala platform. Bahasa pemrograman yang ada pada OpenCL berbasis C++ dan *code* fungsinya paralelnya disebut kernel. Pemrograman OpenCL memungkinkan program untuk dijalankan pada CPU, GPU, prosesor sel, DSP dan banyak *devices* lainnya (Amalia, 2016) (nvidia).

OpenCL pada awalnya dikembangkan oleh Khornos pada tahun 2008, dan sekarang OpenCL dikembangkan oleh AMD. OpenCL yang merupakan *cross-platform*, mampu berjalan pada hampir semua jenis perangkat, berikut :

1. Intel
2. NVIDIA Corporation
3. Apple, Inc.
4. ARM Limited

5. QUALCOMM
6. Marvell
7. Vivante Corporation
8. MediaTek Inc
9. AMD
10. Texas Instruments
11. Altera Corporation
12. Xilinx, Inc.
13. Imagination Technologies
14. STMicroelectronics International NV
15. IBM corporation
16. Creative Labs
17. Samsung Electronics

### 2.8. Multithread

*Thread* adalah sebuah proses yang berukuran kecil yang dibuat oleh sebuah program untuk dijalankan bersamaan dengan *thread-thread* lainnya. Tujuan *thread* ini adalah agar proses –proses yang dapat dikerjakan bersamaan dapat dijalankan secara bersamaan tanpa memerlukan waktu tunggu untuk proses berikutnya. Ini dapat dicontohkan dari perangkat lunak pengolah kata yang membentuk beberapa *thread* untuk melakukan fungsi tampilan, proses pengetikan, dan proses pemeriksaan jumlah kata, dengan pembagian fungsi tersebut kedalam *thread* maka semua proses tersebut dapat dijalankan secara bersamaan tanpa

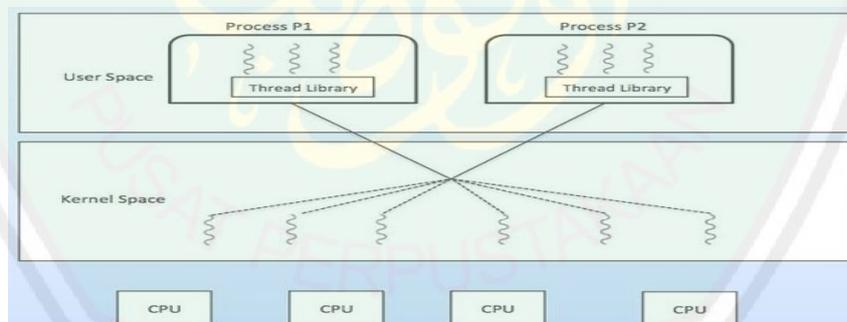
terjadinya delay, sehingga dapat disebut berjalan secara parallel (Kelvin Onggrono, dkk, 2017).

*Multithreading* adalah kumpulan beberapa *thread* yang dijalankan bersamaan pada sebuah processor. Pada *multithreading* ini merupakan proses eksekusi dari kumpulan *thread* dimana kumpulan *thread* tersebut diproses secara berulang – ulang dengan perpindahan dalam waktu nanosecond.

*Multithread* yang berjalan pada sistem operasi memungkinkan untuk menggabungkan *thread* level *user* dengan *thread* level kernel. Oleh karena itu *multithread* memiliki beberapa model (tutorialspoint):

a. *Many to Many Model*

Model *multithread many to many* merupakan model *thread* yang jumlah *thread* level user sama atau lebih sedikit jumlah *thread* level kernel



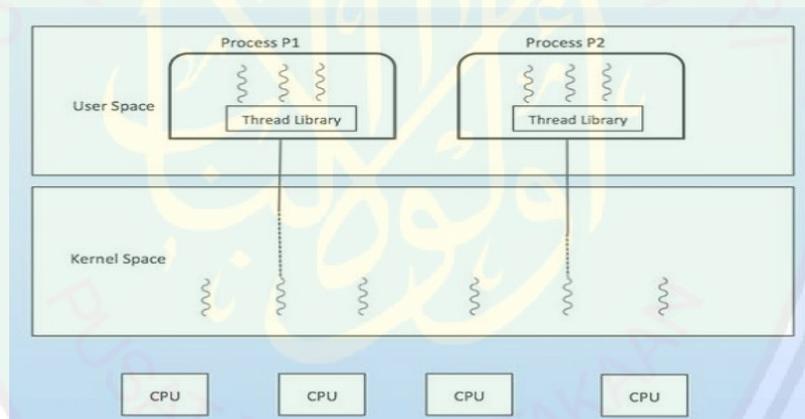
Gambar 2.2. Model *Multithread Many-to-Many*

Gambar 2.2 menunjukkan model *multithreading many-to-many* dimana banyak *thread* level user dengan banyak kernel level kernel sama jumlahnya. Dalam model ini, pengembang dapat membuat sebanyak mungkin *thread* user dan *thread* kernel yang sesuai dapat berjalan secara parallel pada mesin multiprosesor. Model ini memberikan akurasi terbaik pada konkurensi dan

ketika sebuah thread melakukan panggilan sistem *block*, kernel dapat menjadwalkan thread lain untuk eksekusi.

b. *Many to One Model*

Model *many-to-one* memetakan banyaknya jumlah *thread* level user ke dalam satu *thread* tingkat kernel. Manajemen thread dilakukan di ruang pengguna oleh thread library. Saat thread membuat sistem *block*, keseluruhan proses akan diblokir. Hanya satu thread yang bisa mengakses Kernel pada satu waktu, jadi beberapa thread tidak dapat berjalan secara paralel pada multiprocessors. Hal tersebut dijelaskan pada Gambar 2.3



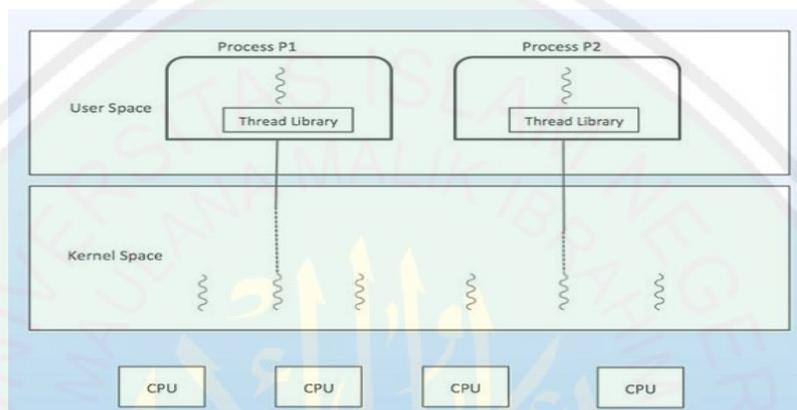
**Gambar 2.3 Model Multithread Many-to-Many**

Jika *library thread* tingkat user diimplementasikan dalam sistem operasi sedemikian rupa sehingga sistem tidak mendukungnya, maka Kernel menggunakan mode hubungan banyak-ke-satu.

c. *One to One Model*

Model *One to One* merupakan model *thread* dengan konsep satu *thread* tingkat user ke satu *thread* tingkat kernel. Model ini memberikan konkurensi

lebih banyak daripada model banyak-ke-satu. Ini juga memungkinkan thread lain dijalankan saat sebuah thread membuat sistem *block*. Hal ini mendukung banyak *thread* untuk dijalankan secara paralel pada mikroprosesor. Hal tersebut dijelaskan pada Gambar 2.4



**Gambar 2.4 Model *Multithread One-to-One***

Kerugian dari model ini adalah membuat benang pengguna memerlukan *thread* Kernel yang sesuai. OS / 2, windows NT dan windows 2000 atau yang menggunakan *one to one relationship* model.

## 2.9. Penjadwalan Matakuliah versi 1.0

Program penjadwalan matakuliah versi 1.0 adalah program penjadwalan perkuliahan yang dibuat dengan menggunakan metode sekuensial untuk melakukan pencarian dalam pembuatan jadwal kuliah. Metode ini bekerja dengan memeriksa data satu per satu secara berurutan. Pemeriksaan data berkaitan dengan implementasi konstrain (batasan) kelas dan dosen untuk menghindari bentrok antar ruang dengan bentrok jadwal dosen yang satu dengan dosen yang lain. Pengaksesan dan pengecekan data semuanya dilakukan di database lokal yakni MySQL. Sesuai dengan penelitian sebelumnya, menyatakan bahwa dengan

menggunakan database MySQL menghasilkan waktu eksekusi yang lebih cepat dari database yang lainnya. Hasil tabel perbandingan waktu eksekusi beberapa database disajikan pada Tabel 2.2

**Tabel 2.2 Perbandingan Waktu Eksekusi Program dari Beberapa Database**

No	DBMS	Waktu Eksekusi (detik)	
		Tabel jadwal kuliah kosong	Tabel jadwal kuliah terisi
1	MS Access	110	2605
2	SQLite	46	1027
3	MySql	51	991

Adapun data-data yang ada pada database tersimpan data jadwal kuliah di tahun-tahun sebelumnya dan menyimpan sebanyak 2040 baris jadwal kuliah dan data plot mengajar yang digunakan oleh program penjadwalan matakuliah versi 1.0 adalah data jurusan Teknik Informatika tahun ajaran 2013/2014 semester ganjil sebanyak 131 baris plot mengajar.

Data-data yang digunakan pada program penjadwalan perkuliahan versi 1.0 adalah data pemasaran matakuliah beserta jumlah sks-nya, data waktu perkuliahan berdasarkan sks, data dosen, data preferensi mengajar, data plot mengajar dan data ruang jurusan Teknik Informatika pada tahun ajaran 2013/2014 semester ganjil.

Data pemasaran matakuliah berisi data-data matakuliah yang dipasarkan beserta data sks dalam semester dan tahun ajaran yang telah ditetapkan. Data pemasaran matakuliah juga memuat semester distribusi untuk setiap matakuliah. Sehingga setiap matakuliah memiliki syarat minimum untuk diambil mahasiswa pada semester yang ditentukan pada semester distribusi tersebut. Data waktu perkuliahan adalah data lama waktu perkuliahan atau jam perkuliahan berdasarkan

dengan bobot sks yang dimiliki oleh suatu matakuliah. Waktu perkuliahan ini dibagi menjadi batas awal waktu perkuliahan dan batas akhir waktu perkuliahan. Data dosen berisi data nama, nip (nomor induk pegawai) dosen dan matakuliah yang diampu.

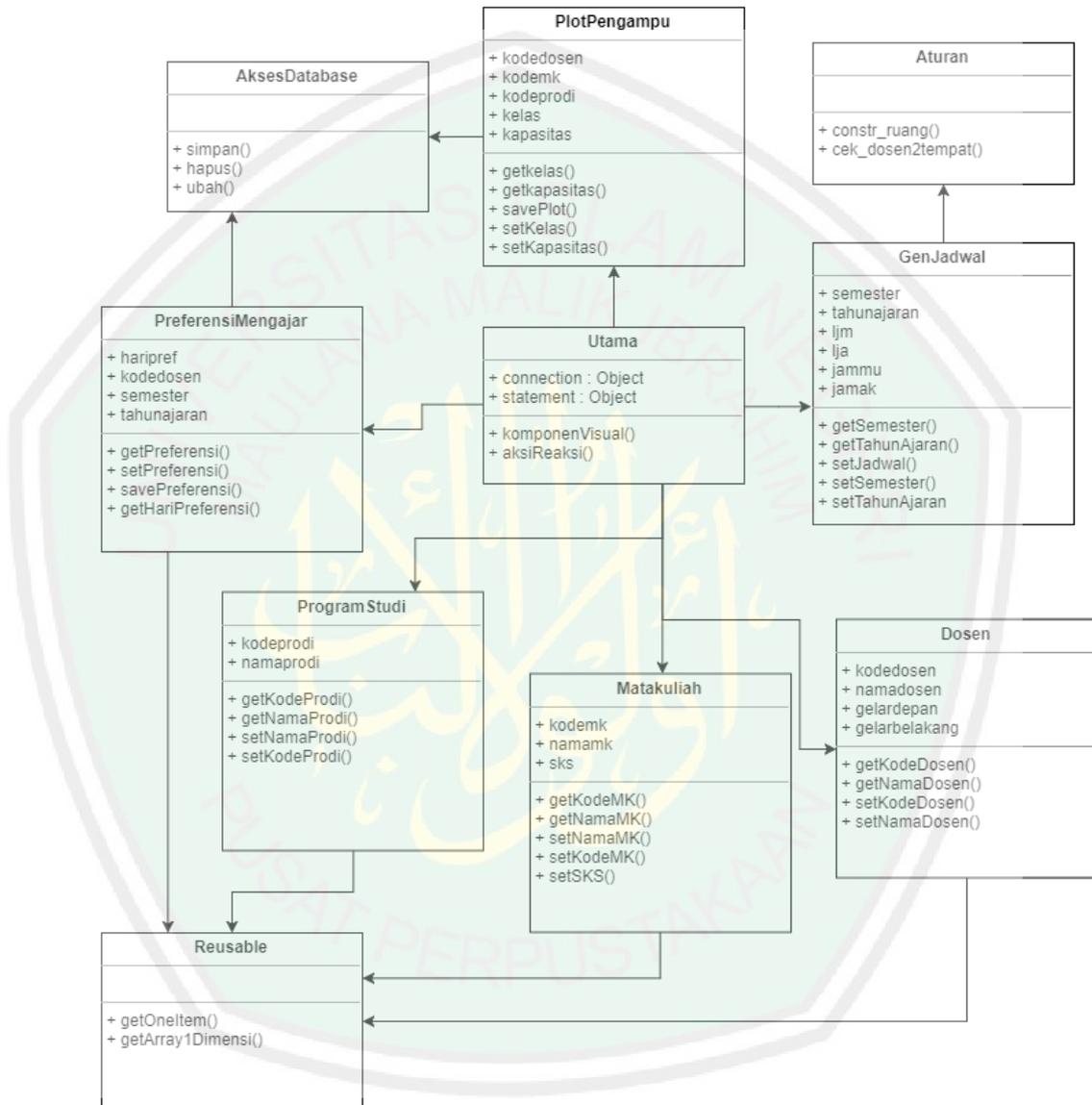
Data preferensi mengajar merupakan data prioritas hari mengajar bagi seorang dosen pengampu matakuliah. Data preferensi mengajar menyimpan hari yang telah dipilih oleh dosen dan kode dosen. Adapun data ruang yang digunakan untuk uji coba dalam penelitian ini menggunakan 5 ruang, 6 ruang dan 7 ruang.

Data plot mengajar merupakan data plot kelas seorang dosen yang mengampu matakuliah tertentu. Dosen tersebut akan disebarakan kedalam beberapa kelas dengan kapasitas kelas yang telah ditentukan. Data plot terdiri dari data semester, tahun ajaran, kode matakuliah, kode dosen, kelas dan kapasitas kelas.

Dari penjelasan tersebut kemudian dibuatlah class untuk mendapatkan data-data perkuliahan dari database. Class-class tersebut digambarkan pada Gambar 2.5.

Pada Gambar 2.5 merepresentasikan class-class dan method yang digunakan pada software penjadwalan matakuliah versi 1.0. Terdapat class `AksesDatabase` untuk menyimpan *source code* yang berhubungan dengan koneksi database. Class `Reusable` digunakan untuk mengambil data dari database kemudian disimpan ke dalam array. Dan class-class lainnya yang mendukung untuk proses penjadwalan, terdapat class `PlotPengampu`, `Aturan`,

PreferensiMengajar, Utama, GenJadwal, ProgramStudi, Matakuliah dan Class dosen.



Gambar 2.5 Class Diagram Software Penjadwalan Versi 1.0

## BAB III

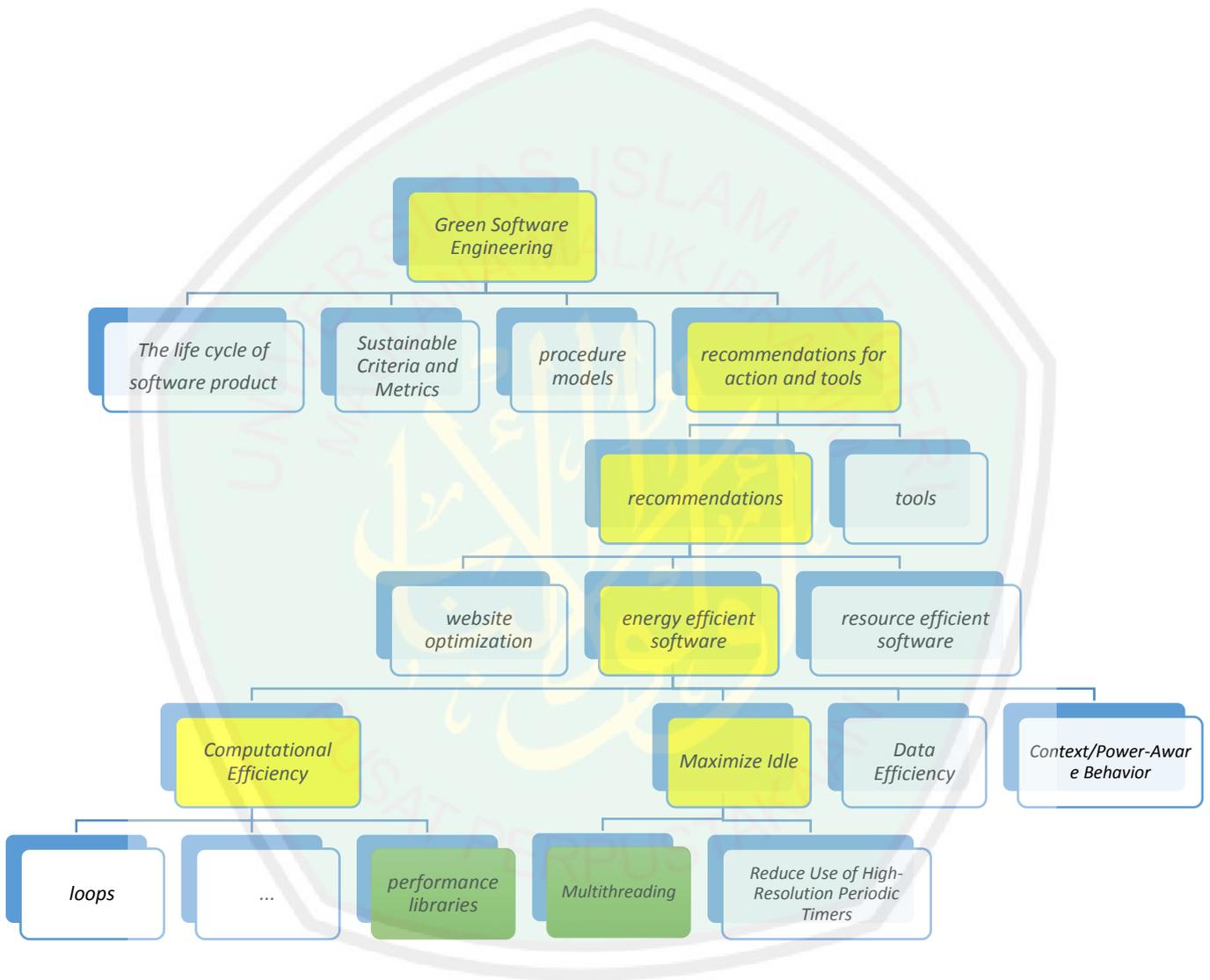
### METODE PENELITIAN

Aplikasi penjadwalan matakuliah versi 1.0 berjalan sangat lama ketika proses *generate* jadwal, maka dibutuhkan proses tambahan untuk mempercepatnya. Menurut (Larsson, 2011) menggunakan *Multithread* adalah salah satu cara untuk efisiensi *software*. Berikut ini penjelasan mengenai langkah-langkah untuk melakukan penelitian secara menyeluruh mulai dari pengolahan data, prosedur penelitian dan pendekatan efisiensi komputasi sebagai rekomendasi dalam perancangan *software*.

#### 3.1. Ruang Lingkup Penelitian

Ruang lingkup pada penelitian ini didasarkan pada pendapat (Larsson, 2011) tentang *energy efficient softwere* yang dapat dilakukan dengan beberapa cara seperti *computational efficient, maximize idle, data efficient, dan power aware behavior*.

Lingkup penelitian ini digambarkan pada Gambar 3.1 dengan blok yang berwarna kuning. Sedangkan kegiatan yang akan dilakukan pada penelitian ini ditandai dengan blok warna hijau.



**Gambar 3.1** Lingkup Penelitian

Gambar 3.1 blok dengan warna kuning dibagian atas sendiri merupakan topik *Green Software Engineering*. Ada beberapa konsep untuk mewujudkan sebuah *software* yang ramah lingkungan. Salah satunya adalah melakukan *Recommendation for Action and Tools* yang berupa pedoman atau tatacara untuk menerapkan konsep *software* yang ramah lingkungan. Cara mengimplementasikannya bisa menggunakan metode yang telah dilakukan oleh peneliti lain. Salah satu penerapan *Recommendation* adalah dengan melakukan *energy efficient software*. Salah satu konsep yang bisa membuat efisiensi energi pada *software* adalah dengan *maximize idle* dan *computational efficiency*. Pada tahap selanjutnya baru diketahui beberapa cara untuk meningkatkan efisiensi dengan menggunakan *multithread* dan *performance libraries*.

#### 3.1.1. Multithread

*Multithread* yang digunakan pada penelitian ini adalah model *many to one* karena setiap data akan dijalankan pada masing-masing *thread* dan hanya satu data yang akan masuk ke dalam kernel. Pada proses *multithread* di penelitian ini juga menambahkan fungsi sinkronisasi data untuk *share* data supaya tidak terjadi bentrok.

#### 3.1.2. Performance Libraries

Efisiensi energi dapat ditingkatkan untuk perhitungan aplikasi intensif . *Performance libraries* seringkali didasarkan pada konsep pengolahan *Single Instruction Multiple Data* (SIMD) yang konsep kerjanya sama dengan komputasi paralel dengan GPU.

### 3.2. Data

Data yang digunakan pada penelitian ini merupakan data Teknik Informatika pada tahun ajaran 2013/2014 semester ganjil dari aplikasi penjadwalan matakuliah versi 1.0 berupa data plot mengajar. Data ini berisi tentang ruang perkuliahan, waktu perkuliahan, hari aktif perkuliahan, matakuliah, kelas, dosen, dan jumlah sks per matakuliah. Beberapa data tersebut diolah secara paralel sesuai prosedur penelitian yang akan dijelaskan pada subbab selanjutnya.

### 3.3. Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non fungsional merupakan bagian penting lainnya agar simulasi perangkat lunak dapat berjalan dengan baik serta untuk mengetahui elemen-elemen yang berhubungan dengan sistem yang sedang berjalan. Tahapan analisis kebutuhan non fungsional terbagi menjadi 2 bagian yaitu analisis kebutuhan perangkat keras, dan analisis kebutuhan perangkat lunak.

#### a. Analisis Kebutuhan Perangkat Keras

##### 1. Komputer dengan spesifikasi :

- *Processor intel pentium dual core*
- RAM 504 MB (*mega byte*)

##### 2. Laptop dengan spesifikasi :

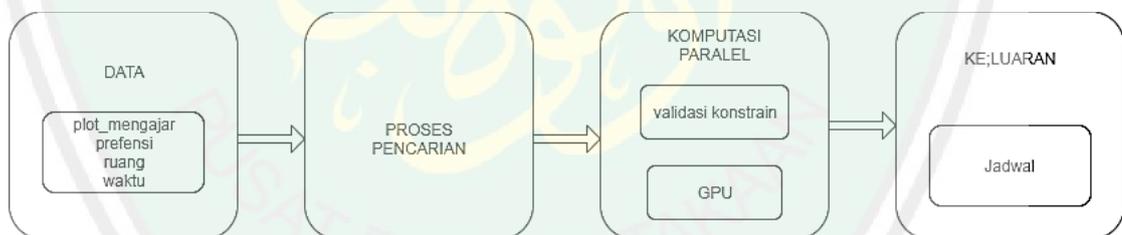
- Processor core i5
- RAM 4 GB (*giga byte*)
- NVIDIA 380

#### b. Analisis Kebutuhan Perangkat Lunak

1. Netbeans versi 6.0 sebagai IDE untuk membangun aplikasi berbasis java dan ditambah dengan plugin OpenCL.
2. MySQL dan Apache sebagai server DBMS (*Database Managent System*) menggunakan XAMPP 1.7.2
3. Windows XP *Proffesional* 2002 sebagai *operating system* komputer dan Windows 7 sebagai OS laptop.

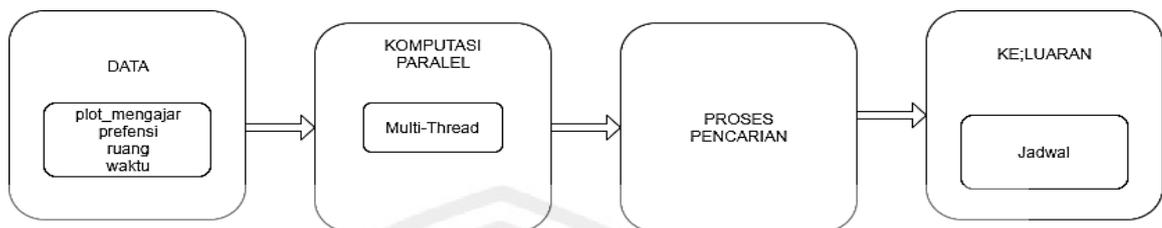
#### 3.4. Perancangan Sistem

Input berupa plot mengajar berisi tentang data ruang, waktu, hari, matakuliah, dosen, kelas, dan sks. Data ini diperoleh dari proses awal aplikasi jadwal dijalankan. Menggunakan GPU dan metode pencarian sederhana untuk memproses data plot mengajar supaya menghasilkan jadwal kuliah.



**Gambar 3.2 Perancangan Sistem GPU**

Metode pencarian ditambah dengan penggunaan GPU dijalankan pada proses validasi konstrain seperti Gambar 3.2. Jadi setiap data metakuliah yang sudah memiliki kandidat ruang, jam, dan hari akan dicek dengan jadwal matakuliah yang sudah terbentuk.



**Gambar 3.3 Perancangan Sistem Multithread**

Proses *multithread* berjalan pada saat awal program setelah semua data pada memori dirubah ke dalam *arraylist*. Setiap satu data plot mengajar akan berjalan pada *multithread* yang dibangun sesuai dengan jumlah plot. Gambar 3.3 menunjukkan metode pencarian berada didalam setiap *thread* yang akan membentuk jadwal.

#### 3.4.1. Blok Masukan

Input pada penelitian ini berupa data plot mengajar, preferensi dosen, matakuliah, jam perkuliahan, dan ruang. Aplikasi versi 1.0 pemakaian data langsung koneksi ke database. Proses koneksi dari program langsung ke database membutuhkan waktu yang relatif lama. Kurang lebih waktu yang dibutuhkan untuk satu kali *generate* jadwal adalah 45 menit.

*Arraylist* merupakan salah satu struktur data untuk menyimpan data yang dilakukan di dalam memori. Java sudah menyediakan *class* sendiri untuk pemakaian *arraylist* sejak versi 1.2. Data dari database akan disimpan pada *arraylist* dengan membuat kelas-kelas pada setiap data yang akan digunakan.

##### 3.4.1.1. Data plot mengajar

Plot mengajar berisi tentang data utama yang akan diolah pada saat *gerenate* jadwal. Kode jurusan, tahun ajaran, dan semester merupakan *field*

data yang sangat menentukan jadwal. Karena untuk membuat jadwal dibutuhkan 3 inputan data tersebut.

Pada sistem GPU proses pengambilan data plot mengajar dilakukan sama seperti versi 1.0, hanya ada sedikit perbedaan saat mengambil data. Pada versi 1.0 pengambilan setiap data memerlukan koneksi ke database, untuk proses GPU koneksi database hanya satu kali ketika data pertama kali dipanggil kemudian data disimpan di dalam *arraylist*.

Proses pengambilan data plot mengajar di *multithread* dilakukan pada struktur data *arraylist*. Pada *multithread* data plot mengajar dibutuhkan untuk proses pembuatan *multithread*. Jumlah *multithread* yang dibuat disesuaikan dengan panjang data dari plot mengajar. Berikut kode objek *arraylist* dalam data plot mengajar.

```
public class listPM {
    public String Semester, TahunAjaran, kodeDosen, jurusan,
        kodeMK, Kelas, Kapasitas, sks, namaDosen,
        namaMatkul;
    public listPM(String Semester, String TahunAjaran, String
        jurusan, String kodeMK, String kelas, String
        kodeDosen, String Kapasitas, String sks, String
        namaDoesn, String namaMatkul) {
        this.Semester = Semester;
        this.TahunAjaran= TahunAjaran;
        this.kodeDosen= kodeDosen;
        this.jurusan = jurusan;
        this.kodeMK = kodeMK;
        this.Kelas = kelas;
        this.Kapasitas = Kapasitas;
        this.sks=sks;
        this.namaDosen=namaDoesn;
        this.namaMatkul=namaMatkul; } }
```

Kode class `listPM` berisi data dari *database* plot mengajar, dosen dan matakuliah yang di dalamnya terdapat data semester, tahun ajaran, kode dosen, jurusan, kodeMK, kelas, kapasitas, sks, nama dosen, dan nama matakuliah. Class `listPM` akan digunakan sebagai class objek untuk menyimpan data *arraylist*. Data dari *arraylist* `listPM` merupakan komponen terpenting dalam proses penjadwalan karena data tersebut yang akan dijadikan jadwal.

```

ArrayList<listPM> lst_plot = new ArrayList<listPM>();
public void setPlotMengajar() {
    ResultSet rs = null;
    try {
        String sql = "SELECT Semester, TahunAjaran, Jurusan,
                    KodeMK, Kelas, KodeDosen, KapasitasKelas, Mtk_SKS,
                    Mtk_NAMA, DOS_GELARDEPAN, DOS_NAMA,
                    DOS_GELARBELAKANG FROM plotmengajar,
                    matakuliah, dosen WHERE KodeMk=Mtk_KODE and
                    DOS_KODE=KodeDosen";
        rs = db.ambilData(sql);
        while (rs.next()) {
            String semester = rs.getString(1);
            String tahunAjaran = rs.getString(2);
            String jurusan = rs.getString(3);
            String kodeMK = rs.getString(4);
            String kelas = rs.getString(5);
            String kodeDosen = rs.getString(6);
            String KpasitasKelas = rs.getString(7);
            String sks = rs.getString(8);
            String namaMatkul=rs.getString(9);
            String namaDosen=rs.getString(10)+"
                    "+rs.getString(11)+"", "+rs.getString(12);
            lst_plot.add(new listPM(semester, tahunAjaran,
                    jurusan, kodeMK, kelas, kodeDosen,
                    KpasitasKelas, sks, namaDosen,
                    namaMatkul));
        }
    } catch (Exception ex) {
        System.out.println("Terjadi kesalahan ambil data
        : " + ex);
    }
}

```

*Method* `setPlotMengajar` merupakan *method* yang berada di kelas memori dan berguna untuk mengisi *arraylist* `lst_plot` dengan data dari

database. *Method* `setPlotMengajar` dipanggil pada saat program *generate* dijalankan. Fungsi dari *arraylist* `lst_plot` untuk menyimpan data matakuliah yang akan di *generate* menjadi jadwal.

#### 3.4.1.2. Data Preferensi, Ruang, dan Jam

Data preferensi berisi tentang kode dosen serta hari dimana dosen bisa mengajar. Dosen bisa memasukkan hari preferensi mengajar sesuai dengan keinginan. Data ini dibutuhkan untuk pencarian hari yang sesuai pada saat proses pencarian slot hari. Kelas `listPreferensi` berguna untuk mengganti proses pengambilan data preferensi hari dosen yang langsung dari *database* dan diganti ke data objek memori *arraylist*.

Data ruang merupakan kumpulan data tentang jumlah ruang yang disediakan untuk satu semester perkuliahan. Data ini menentukan tempat dimana perkuliahan akan dilakukan. Proses pencarian jadwal juga menggunakan data ruang sebagai node yang akan dicari. Kelas `listRuang` berguna untuk mengganti proses pengambilan data ruang yang langsung dari *database* dan diganti ke data objek memori *arraylist*.

Data jam merupakan akumulasi waktu yang dibutuhkan untuk kegiatan perkuliahan. Penentuan waktu tergantung dengan banyaknya sks pada matakuliah. Waktu perkuliahan yang dilaksanakan tidak boleh bentrok dengan sholat dhuhur dan sholat jum'at. Kelas `listJamKuliah` berguna untuk mengganti proses pengambilan data jamkuliah yang langsung dari *database* dan diganti ke data objek memori *arraylist*.

Proses pemindahan data dengan cara seperti sub bab 3.3.1.1 yang menjelaskan data plot mengajar. Pertama membentuk data objek didalam kelas java kemudian baru dilakukan pengisian data *arraylist* yang diambil dari database.

#### 3.4.2. Pencarian Jadwal

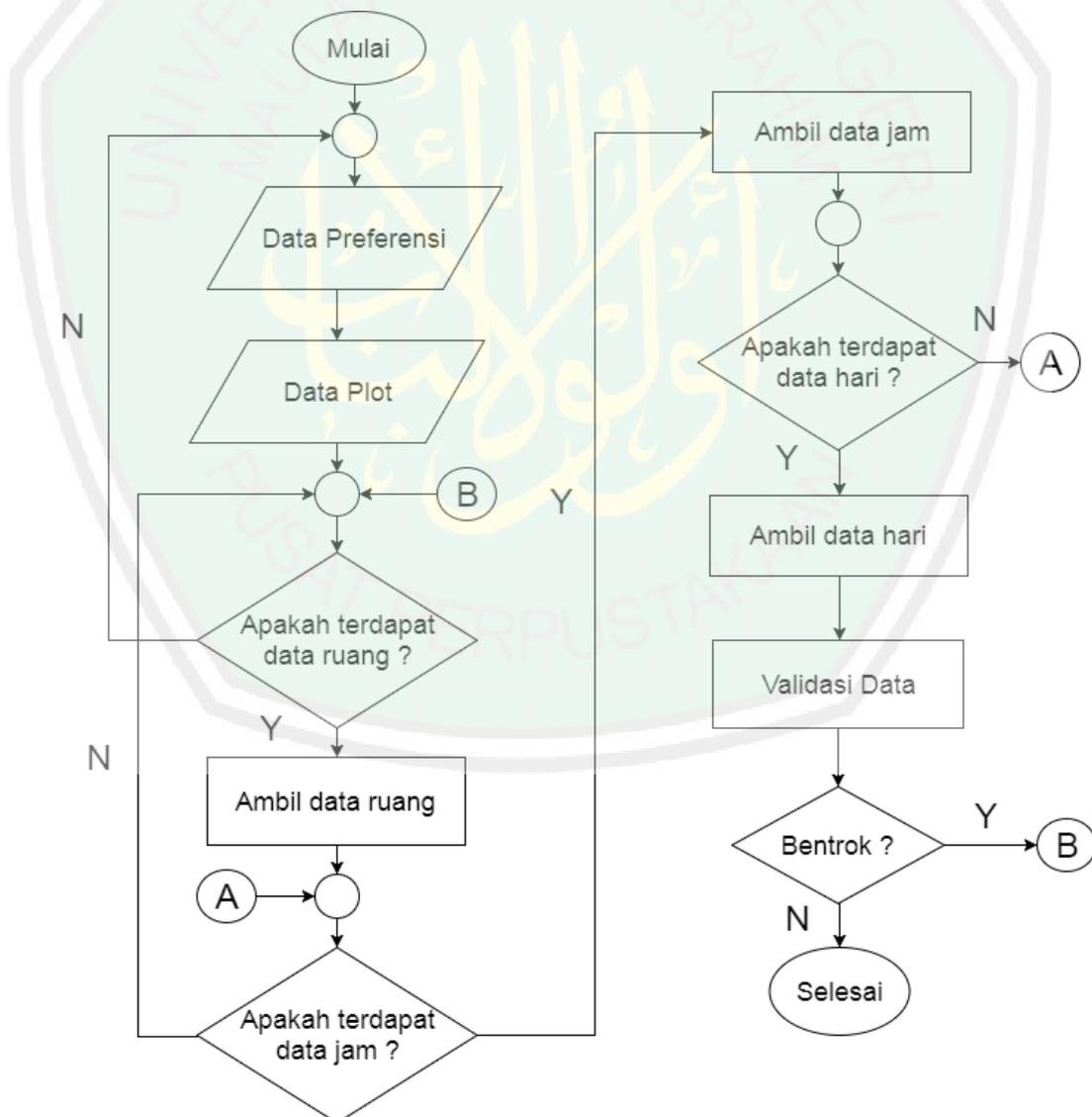
Metode yang digunakan merupakan pencarian sederhana dengan cara memilih node terdekat dan langsung diisi. Pada metode pencarian data matakuliah, dosen, kelas, dan SKS akan diolah untuk membuat kandidat ruang, jam, dan hari yang kosong. Selanjutnya dilakukan validasi konstrain untuk mengetahui bentok data dengan membandingkan beberapa jadwal matakuliah yang sudah tersusun. Ketika terjadi bentrok, maka program akan mengirimkan hasil ke proses pencarian untuk melakukan proses pencarian ulang. Hal ini dilakukan sampai data jadwal matakuliah tidak bentrok dengan slot yang terpilih.



**Gambar 3.4 Model Pencarian**

Gambar 3.4 menerangkan bahwa setiap node ruang memiliki node hari, node hari memiliki node jumlah sks, dan node sks memiliki node jam atau slot di dalam array. Pada aplikasi penjadwalan, ada beberapa konstrain yang harus terpenuhi seperti :

- dosen tidak boleh bentrok pada jam, ruang, dan hari yang sama
- pelaksanaan perkuliahan tidak boleh bentrok dengan waktu sholat dhuhur dan sholat jumat.

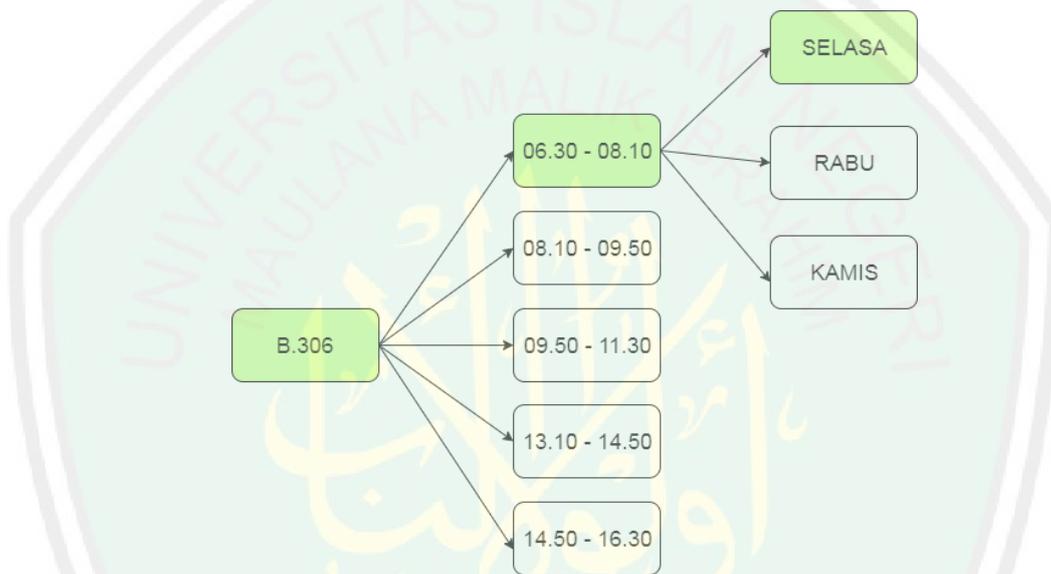


Gambar 3.5 Flowchart Proses Pencarian

Gambar 3.5 merupakan penjelasan jalannya program pencarian jadwal secara linier. Dimulai dari data plot sebagai data yang akan diolah atau dicari slot penjadwalannya. Kemudian baru masuk ke proses pencarian ruang, jam dan hari sebagai slot jadwal. Berikut contoh implementasi model pencarian :

- a. Data plot mengajar berisi, semester = 1, tahun ajaran = 1314, kode jurusan = 55, kode matakuliah = 0765309, kelas = A, kode dosen = 65002, dan kapasitas kelas = 40.
- b. Kemudian mencari ruang yang tersedia pada kode jurusan 55, tahun ajaran = 1314 dan semester = 1, didapat 7 ruang yang tersedia. Selanjutnya mencari sks matakuliah untuk menentukan slot jam yang sesuai. Sks matakuliah dengan kode = 0765309 adalah 2 sks, sehingga slot jam perkuliahan disesuaikan. Data yang terakhir di cari adalah data preferensi hari dosen mengajar. Data ini diambil dengan kriteria kode dosen = 65002, tahun ajaran = 1314, dan semester = 1. Data hasil preferensi dosen adalah selasa, rabu, dan kamis.
- c. Proses selanjutnya melakukan pengecekan ruang yang tersedia mulai dari B.306, B.307, B.314, B.315, B.316, B.317, sampai B.318
- d. Setiap slot ruang melakukan *nested loop* terhadap data waktu. Setelah itu masuk ke perulangan jam perkuliahan, dimulai dari jam 06.30 sampai 16.30.
- e. Proses selanjutnya pencarian slot hari yang sesuai dengan preferensi dosen. Untuk contoh ini hari yang digunakan adalah selasa, rabu dan kamis. Proses ini dilakukan didalam *nested loop* waktu

- f. Ketika ketiga slot tersebut sesuai maka akan dicek apakah masih kosong atau tidak. Proses pengecekan ini masuk kedalam validasi konstrain yang dijelaskan bab sub bab 3.4.2. Ketika data tidak terjadi bentrok maka data disimpan langsung ke data jadwal. Contoh penerapan slot pada contoh implementasi ini terdapat pada Gambar 3.6



**Gambar 3.6 Contoh Implementasi Pencarian Slot Jadwal**

- g. Pada Gambar 3.6 didapat data slot jadwal untuk plot mengajar yang tersedia pada langkah a yaitu ruang B.306, waktu perkuliahan 06.30-08.10 dan harinya adalah selasa.
- h. Proses a sampai g dilakukan kembali sampai semua data plot mengajar selesai.

### 3.4.3. Validasi Konstrain

Validasi konstrain merupakan proses untuk pengecekan jadwal yang disimpan ke dalam data jadwal. Ada dua konstrain yang dilakukan validasi yaitu

bentrok dosen dan ruangan. Data yang diproses merupakan data dari hasil pencarian slot ruang, hari dan jam menggunakan pencarian. Pada validasi konstrain juga terdapat aturan untuk tidak melanggar waktu sholat.

Kriteria untuk menentukan bentrok atau tidaknya jadwal dilakukan pada kelas tersendiri yaitu kelas Aturan.java. Pada aplikasi versi 1.0 kelas Aturan.java memiliki 2 *method*, yaitu *method* validasi bentrok dosen dan *method* validasi ruang. Kedua *method* tersebut memiliki persamaan, yaitu memproses data hari perkuliahan, jam mulai matakuliah, jam akhir matakuliah, dan tahun ajaran. Ada 3 kriteria untuk menentukan bentrok atau tidaknya jadwal dengan data.

jam mulai jadwal  $\geq$  jam mulai data dan jam mulai jadwal  $<$  jam akhir data.

jam mulai jadwal  $<$  jam akhir data dan jam akhir jadwal  $>$  jam mulai data.

jam mulai data  $\geq$  jam mulai jadwal dan jam akhir data  $\leq$  jam akhir jadwal.

```

if ((hari.equals(h) && jammu.getTime() >= ljm && jammu.getTime() <
    lja && tahunajaran.equals(t) && semester.equals(s)) &&
    (ruang.equals(r) || kodedosen.equals(kd))) {
    test = true;
    break;
} else if ((hari.equals(h) && jammu.getTime() < lja &&
    jamak.getTime() > ljm && tahunajaran.equals(t) &&
    semester.equals(s)) && (ruang.equals(r) ||
    kodedosen.equals(kd))) {
    test = true;
    break;
} else if ((hari.equals(h) && ljm >= jammu.getTime() && lja <=
    jamak.getTime() && tahunajaran.equals(t) &&
    semester.equals(s)) && (ruang.equals(r) ||
    kodedosen.equals(kd))) {
    test = true;
    break;
} else {
}
}

```

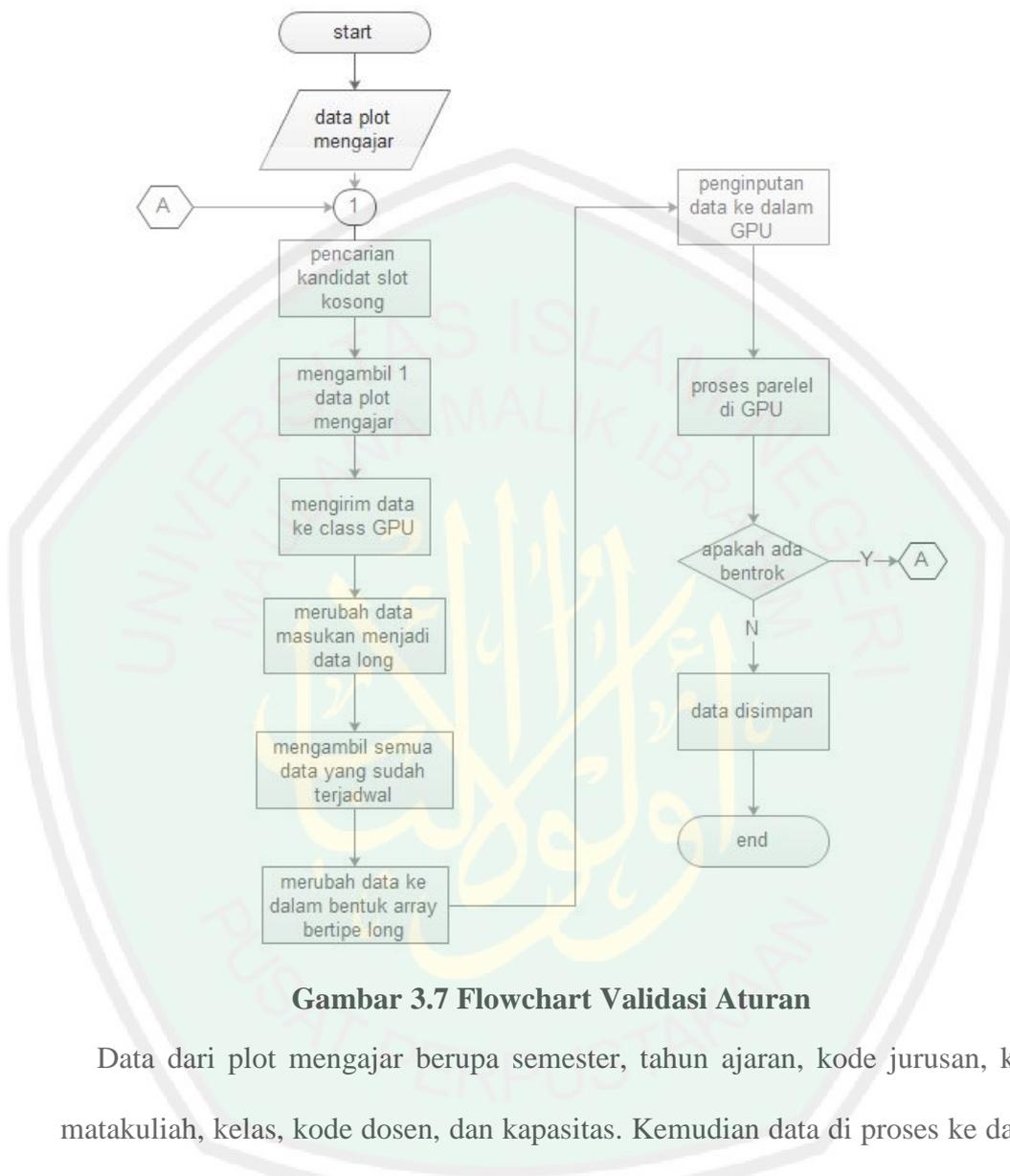
*Method* dosen dan ruang yang masih terpisah menjadi 2 , digabung menjadi satu untuk mengurangi proses *looping* validasi. Penggabungan proses

validasi dilakukan dengan cara memasukkan kode dosen ke dalam *method* validasi ruang. Penambahan tersebut terletak pada setiap kriteria validasi. Proses validasi ruang dan validasi konstrain menghasilkan nilai bentrok bila *variable test* bernilai *true*, sehingga proses pencarian akan dilakukan lagi sampai nilai test pada kelas *Aturan.java* bernilai *false*.

#### 3.4.4. GPU

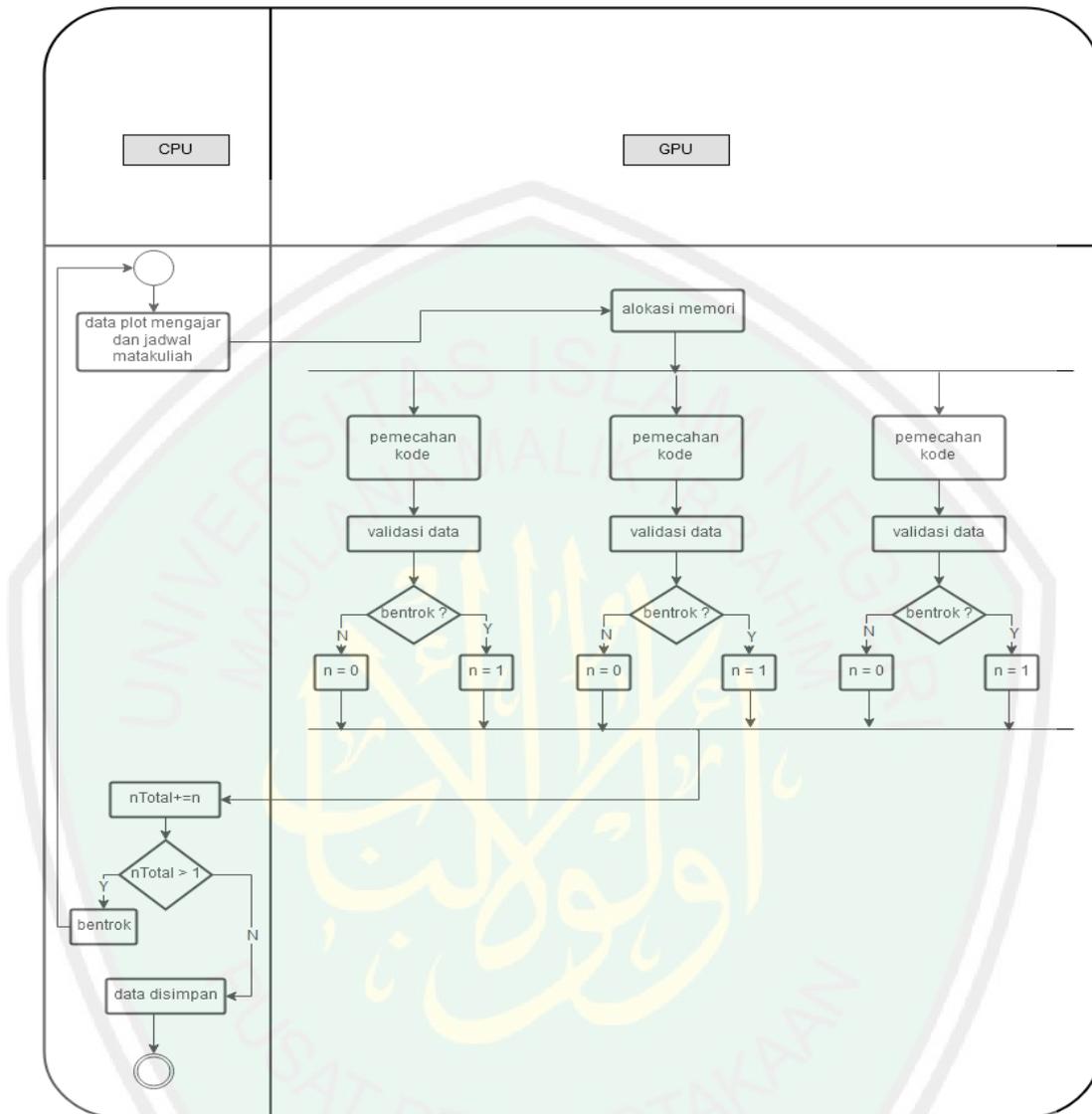
Komputasi paralel dengan menggunakan GPU dilakukan pada proses validasi konstrain. Karena proses validasi membutuhkan waktu yang lama dibandingkan dengan proses komputasi yang lainnya. Proses validasi terus berlangsung sampai menemukan kandidat ruang, jam, dan hari yang tidak berbenturan dengan konstrain yang sudah ditetapkan.

Pada proses validasi konstrain, data semester, tahun ajaran, kelas, hari, ruang, kapasitas, jam mulai, jam akhir, kode dosen dan kode matakuliah dijadikan satu data menjadi data bertipe *long*, dan disimpan dalam array untuk proses paralel. Data pembanding berupa data matakuliah yang sudah terjadwal dijadikan data bertipe *long*, karena GPU hanya bisa memproses data yang bertipe *primitive*.



**Gambar 3.7 Flowchart Validasi Aturan**

Data dari plot mengajar berupa semester, tahun ajaran, kode jurusan, kode matakuliah, kelas, kode dosen, dan kapasitas. Kemudian data di proses ke dalam pencarian sehingga menghasilkan slot ruang, jam, dan hari. Selanjutnya data akan dirubah ke dalam data array *long* untuk dikirim ke GPU. Gambar 3.7 menerangkan bahwa semua data akan dirubah dalam bentuk array *long* dan diproses menggunakan GPU.



**Gambar 3.8 Activity Diagram GPU**

Proses validasi paralel dilakukan saat data masuk ke dalam memori kelas GPU. Data plot mengajar dan jadwal matakuliah akan dikirim ke GPU dalam bentuk array dengan tipe data *long*. Pada Gambar 3.8, proses pemecahan kode menjelaskan tentang pengambilan data *long*. Kemudian data *long* tersebut digunakan pada proses validasi.

Proses pembangkitan jadwal dilakukan pada class GenJadwal. Data dari plot mengajar, jadwal kuliah, ruang, preferensi, dan jam kuliah dirubah kedalam

bentuk *arraylist*. Tahap pencarian slot matakuliah yang kosong menggunakan metode. Setiap node yang masih kosong akan diisi sesuai dengan kriteria dan tidak boleh bertrok dengan konstrain yang sudah ditetapkan.

Pencarian slot ruang, jam, dan hari untuk setiap plot mengajar akan dilakukan pada perulangan pencarian. Selanjutnya data slot akan dimasukkan ke proses validasi yang dilakukan secara paralel. Data yang dimasukkan dirubah ke dalam satu data bertipe *long*, berikut contoh perubahan data :

- kode dosen : 65002
- tahun ajaran : 1314
- hari : selasa
- jam mulai : 06.30
- jam akhir : 09.00
- ruang : B.316
- semester : 1
- data *long* 1 dengan struktur : hari, jam mulai, jam akhir, tahun ajaran, semester.
- data *long* 1 = 20630090013141
- data *long* 2 dengan struktur : kode dosen,ruang.
- Data *long* 2 = 650021

Data *long* class *genJadwal.java* akan diolah pada *class* GPU untuk proses paralel. Data tersebut akan di pecah kembali menjadi data pembanding yang dilakukan di GPU. Berikut cara pengambilan data :

- hari = long1 / 1000000000000000
- jam mulai = (long1 % 1000000000000000) / 1000000000

- $\text{jam akhir} = (\text{long1} \% 1000000000) / 100000$
- $\text{tahun ajaran} = (\text{long1} \% 100000) / 10$
- $\text{semester} = \text{long1} \% 10$
- $\text{kode dosen} = \text{long2} / 10$
- $\text{ruang} = \text{long2} \% 10$

Berikut contoh implementasi komputasi penjadwalan dengan menggunakan GPU :

1. Data sampel plot mengajar, preferensi, matakuliah, jam, dan jadwal kuliah berbentuk arraylist dengan struktur sama seperti di tabel database.
2. Proses selanjutnya, dilakukan proses pencarian slot ruang, jam dan hari yang sesuai dengan preferensi hari dosen.
3. Melakukan proses validasi dengan merubah data plot menjadi data yang dibutuhkan dalam proses paralel GPU. Berikut kodenya :

```
String jamMul = ""+jammulai;
String jamAkh = ""+jamakhir;
String in     = pref.get(k) +pj(jamMul)+ pj(jamAkh)+
tahunajaran +
           semester;
String in1    = kodedosen+ruangKelas(ruang.get(i));
long data[]  = new long [mem.lst_jadwal_kuliah.size()];
long masuk[] = {Long.parseLong(in)};
long masuk1[] = {Long.parseLong(in1)};
JoclTemplate11 jocl = new JoclTemplate11(mem,
masuk[0],masuk1[0]
           ,data);

if(jocl.aturan!=true){
           //kode program penyimpanan
}else{
}
}
```

Pada baris kode `JoclTemplate11 jocl = new JoclTemplate11(mem,masuk[0],masuk1[0],data);` berfungsi

untuk memanggil dan menjalankan kelas `JoclTemplate1.java`. Data `jocl.aturan` yang telah terisi nilai di kelas `JoclTemplate1` digunakan pada tahap pengecekan selanjutnya. Ketika `jocl.aturan` bernilai *false* maka komputasi akan dilanjutkan dengan menyimpan data hasil pencarian ke dalam database jadwal dan sebaliknya ketika `joc.aturan` bernilai *true* maka proses pencarian slot dilakukan kembali.

4. Data tersebut dikirim ke kelas `JoclTemplate1.java` untuk proses pembangkitan GPU. Berikut kodenya :

```

CLProgram program =
context.createProgram(JoclTemplate1.class.getResourceAsStream("tesCL.cl"));

program.build();

CLBuffer<LongBuffer> clBufferA
= context.createLongBuffer(globalWorkSize, READ_ONLY);
CLBuffer<LongBuffer> clBufferB
= context.createLongBuffer(globalWorkSize, READ_ONLY);
CLBuffer<LongBuffer> clBufferAa
= context.createLongBuffer(globalWorkSize, READ_ONLY);
CLBuffer<LongBuffer> clBufferBb
= context.createLongBuffer(globalWorkSize, READ_ONLY);
CLBuffer<LongBuffer> clBufferC
= context.createLongBuffer(globalWorkSize, WRITE_ONLY);

fillBuffer(clBufferB.getBuffer(), 67890);
isiA(clBufferA.getBuffer(), a);
isiB(clBufferB.getBuffer(), data);
isiAa(clBufferA.getBuffer(), aa);
isiBb(clBufferB.getBuffer(), data1);

CLKernel kernel = program.createCLKernel("sampleKernel");
kernel.putArgs(clBufferA, clBufferB, clBufferAa, clBufferBb,
               clBufferC).putArg(elementCount);
CLCommandQueue queue =
context.getMaxFlopsDevice().createCommandQueue();
queue.putWriteBuffer(clBufferA, false)
      .putWriteBuffer(clBufferB, false)
      .putWriteBuffer(clBufferAa, false)
      .putWriteBuffer(clBufferBb, false)
      .put1DRangeKernel(kernel, 0, globalWorkSize, localWorkSize)
      .putReadBuffer(clBufferC, true);

context.release();

```

Kode untuk mengaktifkan fungsi kernel yang telah dibuat terdapat pada baris program `CLProgram program = context.createProgram (JoclTemplate11.class.getResourceAsStream ("tesCL.cl"));` yang berfungsi untuk memanggil file `tesCL.cl`. Pada kode tersebut terdapat perintah untuk mengisi data uji yang akan dikirim ke karnel `clBufferA` dan `clBufferAa` untuk data array jadwal kuliah, `clBufferB` dan `clBufferBb` untuk data uji slot kuliah , dan `clBufferC` untuk data hasil komputasi yang disimpan di array hasil.

5. Data plot mengajar yang telah dirubah dalam bentuk array *long*, dikirim ke kernel GPU, dan diproses validasi. Berikut kode karnel untuk memproses data plot :

```
kernel void sampleKernel(global long *a,global long *b,global
                        long *aa,global long *bb,global long
                        *hasil, int size){

int id = get_global_id(0);

long hari = a[0]/1000000000000000L;
long jm= (a[0]%1000000000000000L)/1000000000L;
long ja= (a[0]%1000000000L)/100000L;
long kodos = aa[0]/10L;
long ruang=(aa[0]%10L);
long tahan = (a[0]%100000L)/10L;
long sem = (a[0]%10L);

long haril = b[id]/1000000000000000L;
long jml= (b[id]%1000000000000000L)/1000000000L;
long jal= (b[id]%1000000000L)/100000L;
long kodosl = bb[id]/10L;
long ruangl=(bb[id]%10L);
long tahanl = (b[id]%100000L)/10L;
long seml = (b[id]%10L);

if( (hari==haril && tahan==tahanl && sem==seml && jm>=jml &&
    jm<jal) && (kodos==kodosl || ruang==ruangl)){
    hasil[id] = 1;
}else if ((hari==haril && tahan==tahanl && sem==seml && jm<jal
    && ja>jml) && (kodos==kodosl || ruang==ruangl)){
    hasil[id] = 1;
```

```

}else if ((hari==haril && tahan==tahan1 && sem==sem1 &&
        jml>=jm && jal<=ja) && (kodos==kodos1 ||
        ruang==ruang1)){
        hasil[id] = 1;
}else{
        hasil[id] = 0;
}
}

```

void sampleKernel merupakan method yang akan dieksekusi oleh OpenCL.

Pada code tersebut terdapat proses penguraian data dari data *long* yang panjang dibagi menjadi beberapa bagian untuk proses validasi dan hasilnya akan disimpan pada array *long* hasil.

- Hasil dari array *long* kemudian ditotal untuk mengetahui matakuliah tersebut bentrok atau tidak. Berikut kode program :

```

long h1=0;
for(int i=0;i<data.length;i++){
    h1+=clBufferC.getBuffer().get();
}
if(h1>0){
    aturan = true;
}else{
    aturan = false;
}

```

fungsi *loop* pada kode untuk menghitung jumlah nilai hasil, kemudian ketika total hasil lebih dari sama dengan 1 maka jadwal yang diuji mengalami bentrok, sehingga dilakukan lagi proses pencarian slot dan sebaliknya.

- Semua data yang ada pada plot mengajar akan diproses sama seperti langkah 1 sampai 6. Jika semua plot sudah dilakukan pencarian dan validasi maka jadwal akan tampil pada *interface* yang sudah disediakan.

### 3.4.5. Multithread

Proses komputasi paralel dengan *multithread*, dijalankan pada *class* GenJadwal.java sebanyak panjang data plot mengajar. Satu data dari plot mengajar akan masuk ke dalam satu *thread* sehingga proses paralelnya berjalan dengan bersama-sama. Proses *multithread* meliputi seluruh proses komputasi yang ada pada aplikasi termasuk validasi konstrain, pencarian, dan penyimpanan hasil komputasi. Sehingga dibutuhkan sinkronisasi antar *thread* supaya data dapat diolah secara efektif.

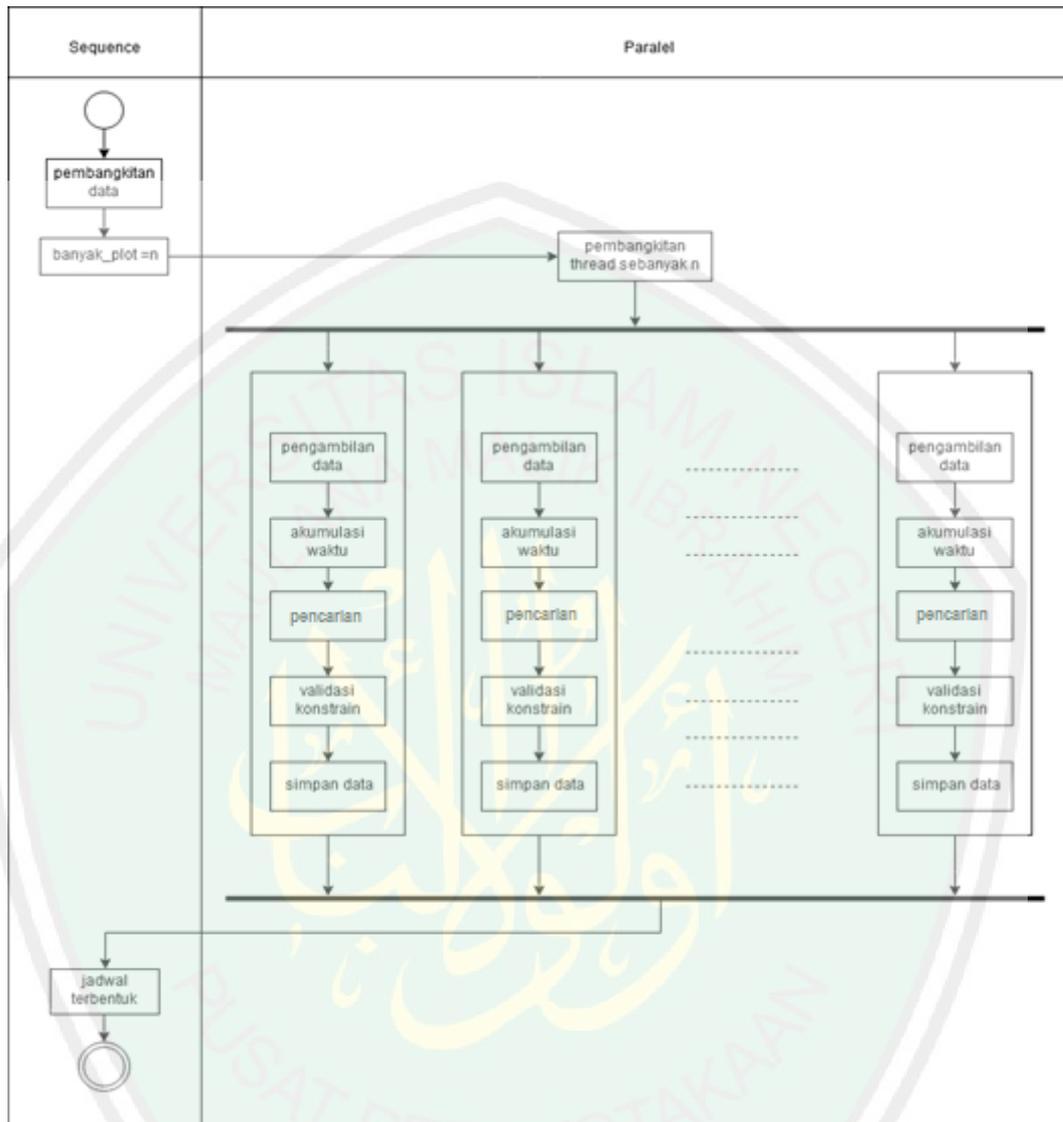


Gambar 3.9 Flowchart Multithread

Gambar 3.9 pada proses pertama, data dari database dirubah dalam bentuk *arraylist*. *Arraylist* plot mengajar digunakan untuk proses pembentukan *thread*. Banyaknya *thread* tergantung panjang data *arraylist* plot mengajar, sehingga looping saat membangkitkan *thread* sama panjang dengan data plot mengajar. Data plot mengajar akan masuk satu per satu ke dalam *thread* dan selanjutnya dilakukan sinkronisasi data terhadap kelas *thread* yang sudah terjadwal. *Output* proses dari *multithread* jadwal, matakuliah tersusun secara *random* setiap generate data. Hal ini tergantung dengan data plot jadwal yang terjadwal di awal proses *thread*.

Proses komputasi diawali dengan pengambilan data dari database dan dirubah kedalam bentuk *arraylist*. Data matakuliah, plot mengajar, dosen, ruang, dan waktu disimpan ke memori pada saat awal program dijalankan. Data *arraylist* tersebut akan diolah dan kemudian diproses ke dalam pencarian jadwal yang berjalan pada masing-masing *thread* yang telah dibuat.

Proses *multithread* membutuhkan kelas *synchronization* untuk mendapatkan data yang sama antar kelas *thread*. Kemudian data diolah secara paralel agar bisa digunakan secara bersama-sama. Sehingga proses *synchronization* data dilakukan setelah proses penyimpanan jadwal. Jadwal yang sudah selesai diproses, akan ditampilkan pada *form* yang terdapat di GUI java. Proses menampilkan jadwal dilakukan secara paralel dengan menggunakan data *arraylist*, setelah semua data plot telah selesai dijadwalkan. Proses *multithread* pada penelitian ini menggunakan pemodelan *multithread one-to-one*. Jadi untuk satu *thread* level user akan dikerjakan oleh satu *thread* level kernel



**Gambar 3.10 Activity Diagram Multithread**

Proses komputasi yang terdapat di *thread*, sama dengan proses pencarian jadwal pada aplikasi versi 1.0. Perbedaannya terdapat pada data, untuk aplikasi *multithread* tidak perlu looping untuk mendapatkan data plot mengajar dan di tambah dengan sinkronisasi data ketika proses komputasi selesai. Gambar 3.10 menjelaskan proses sinkronisasi ini bertujuan untuk menghindari tabrakan data pada proses pembentukan kandidat ruang, jam dan hari.

Hasil jadwal yang terbentuk dari proses *multithread* lebih bervariasi karena untuk menyelesaikan satu data membutuhkan waktu yang berbeda. Sehingga data yang masuk ke dalam jadwal bisa *random* dan berganti-ganti. Berikut merupakan contoh implementasi dengan *multithread* :

1. Data dari database berupa preferensi, jam kuliah, ruang, plot mengajar dan jadwal kuliah disimpan kedalam *arraylist*. Data *arraylist* plot mengajar tersebut berbentuk seperti tabel pada database, seperti contoh pada Tabel 3.1. Preferensi hari mengajar dosen digambarkan pada tabel 3.2

**Tabel 3.1 Contoh Data Plot Mengajar**

No	Kode Mk	Kode Dosen	Semester	Tahun Ajaran	Kelas
1	0765314	65016	1	1314	A
2	0765314	65016	1	1314	B
3	0765309	65002	1	1314	A
4	0765309	65002	1	1314	B

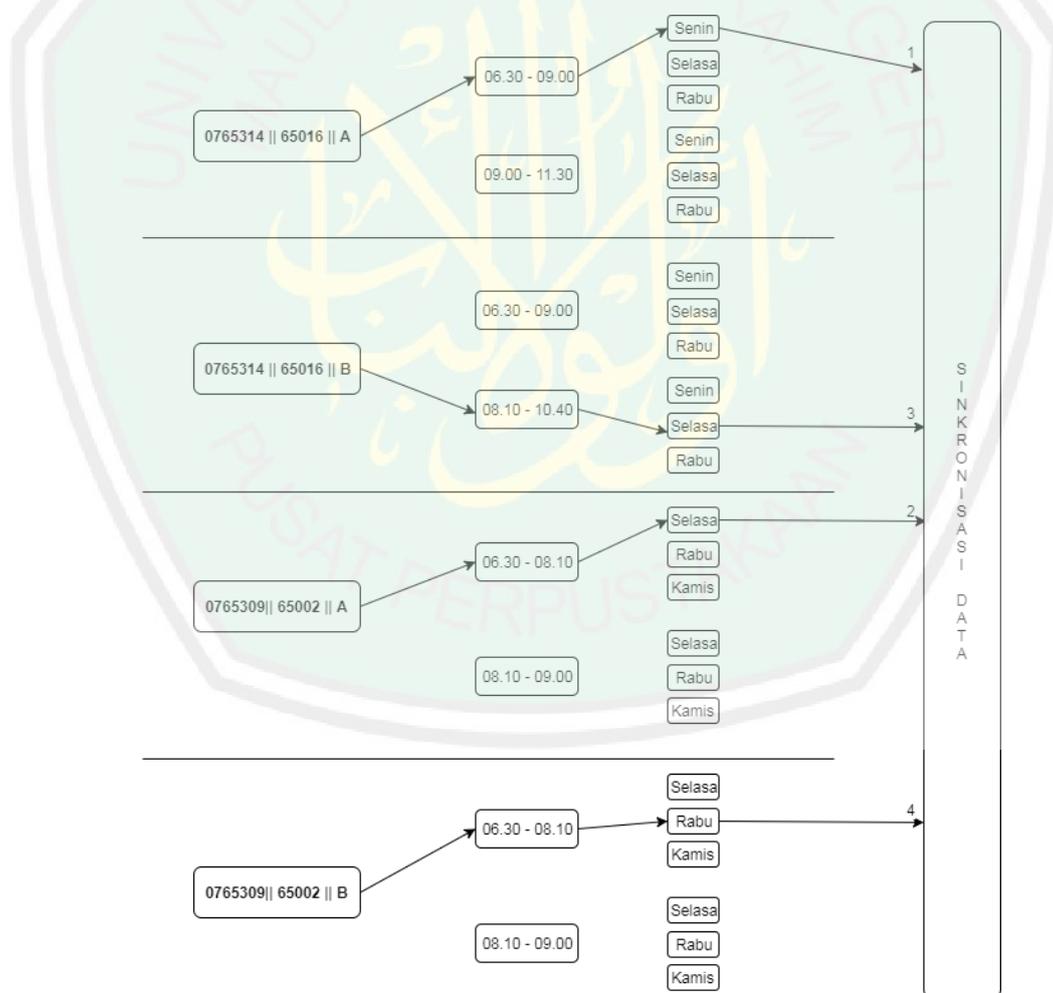
**Tabel 3.2 Contoh Data Preferensi Mengajar**

No	Kode Dosen	Preferensi hari
1	65002	2,3,4
2	65016	1,2,3

2. Jumlah plot mengajar digunakan sebagai jumlah banyaknya *thread* yang akan di buat. Contohnya jumlah plot mengajar pada Table 3.1 ada 4 data, maka jumlah *thread* yang dibuat sebanyak 4. Berikut kode pembuatan *thread* :

```
ExecutorService executor =
    Executors.newFixedThreadPool(mem.lst_plot.size());
```

3. Barulah setiap data plot mengajar akan masuk kedalam masing-masing *thread* yang dibuat, serta berjalan secara bersama-sama. Proses paralel tersebut dapat diilustrasikan dengan adanya blok-blok proses yang berjalan secara bersama-sama.
4. Proses pencarian jadwal dilakukan di dalam *thread* setiap plot. Pencarian tersebut prosesnya sama seperti sub bab 3.4.2. Contoh proses pencarian secara *random*, seperti Gambar 3.11.



**Gambar 3.11 Contoh Proses Pencarian *Multithread***

5. Setelah satu *thread* selesai melakukan proses pencarian maka setiap *thread* akan melakukan sinkronisasi data. Berikut kode sinkronisasinya :

```
public void run() {
    synchronized(GenThread.class) {
        setJadwal();
    }
}
```

Proses sinkronisasi *thread* dilakukan ketika satu kelas *thread* selesai melakukan proses pencarian jadwal. Sinkronisasi dilakukan antar kelas *thread* yang berjalan.

6. Data yang pertama selesai proses *thread* akan dimasukkan kedalam jadwal kuliah. Hal itulah yang menyebabkan hasil jadwal menjadi random. Hasil dari data uji seperti Gambar 3.12

0765314    65016    A	B.306	06.30 - 09.00	Senin
0765314    65016    B	B.306	08.10 - 10.40	Selasa
0765309   65002    A	B.306	06.30 - 08.10	Selasa
0765309   65002    B	B.306	06.30 - 08.10	Rabu

**Gambar 3.12 Hasil Data Uji**

7. Ketika semua data plot sudah selesai melakukan pencarian jadwal dan telah selesai sinkronisasi data, maka hasilnya akan ditampilkan pada *interface* yang sudah disediakan.

### 3.4.6. Blok Keluaran

Output pada proses ini adalah jadwal perkuliahan yang sudah sesuai dengan beberapa konstrain. Total jadwal yang harus tersusun sebanyak 131 matakuliah dengan data, 7 ruang perkuliahan, hari aktif senin sampai jumat, dan waktu perkuliahan mulai jam 06.30 sampai 16.30.

Jadwal yang sudah terbentuk secara otomatis akan tersimpan pada database dan pada proses validasi, data akan diambil kembali kemudian dibandingkan dengan data yang tersimpan. Berikut merupakan contoh gambar output jadwal yang sudah jadi pada versi 1.0 :

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Selasa	09.50 - 11.30	0765216	RANGKAIAN DIGITAL	2	A	65011	TOTOK CHAMIDI, M.Kom	B.314
Selasa	12.20 - 14.50	0765210	STRUKTUR DATA	3	B	65008	ZAINAL ABIDIN, S.Kom	B.314
Selasa	14.50 - 16.30	0765216	RANGKAIAN DIGITAL	2	C	65011	TOTOK CHAMIDI, M.Kom	B.314
Selasa	06.30 - 09.00	0765218	JARINGAN KOMPUTER	3	A	65020	Yumifa MT	B.315
Selasa	09.00 - 11.30	0765218	JARINGAN KOMPUTER	3	D	65020	Yumifa MT	B.315
Selasa	12.20 - 14.50	0765301	KECERDASAN BUATAN	3	C	65102	Hani, M.T	B.315
Selasa	14.50 - 16.30	0765309	REKAYASA PERANGKAT LUNAK	2	A	65002	FATCHURROCHMAN, M.Kom	B.315
Selasa	06.30 - 09.00	0765303	PENGOLAHAN CITRA	3	C	65015	Dr Cahyo Crystdian	B.316
Selasa	09.00 - 11.30	0765305	PEMROGRAMAN MULTIMEDIA	3	C	65019	Fressy MT	B.316
Selasa	12.20 - 14.00	0765309	REKAYASA PERANGKAT LUNAK	2	E	65002	FATCHURROCHMAN, M.Kom	B.316
Selasa	14.00 - 16.30	0765313	KOMPUTASI BERGERAK	3	A	65101	Syauqi, M.Kom	B.316
Selasa	06.30 - 09.00	0765313	KOMPUTASI BERGERAK	3	B	65101	Syauqi, M.Kom	B.317
Selasa	09.00 - 11.30	0765314	SISTEM INFORMASI GEOGRAFIS	3	E	65018	Fahrul Kurniawan M.MT	B.317
Selasa	12.20 - 14.00	0765402	ISU PROFESIONAL & PROYEK G...	2	C	65021	Ali Wafa M.Kom	B.317
Selasa	14.00 - 15.40	0765404	MANAJ. PROYEK PERANGKAT LU...	2	D	65010	MUHAMMAD AINUL YAQIN, M.Kom	B.317
Rabu	06.30 - 09.00	0765103	BAHASA INGGRIS I	3	C	65111	Robert Aprillianto M.Pd	B.306
Rabu	09.00 - 10.40	0765109	STUDI AL-HADITS	2	A	65199	Sugeng Ali Mansyur M.Pd	B.306
Rabu	10.40 - 14.30	0865122	TARIYAHULULU AL BAR	1	A	65007	ROBOINDA MELANI, S.Kom	B.306

**Gambar 3.13 Contoh Output Jadwal Versi 1.0**

Gambar 3.13 menjelaskan bahwa output yang disimpan pada database akan langsung dicetak ketika proses komputasi generate jadwal selesai. Data hari, jam, kode MK, nama MK, SKS, kelas, kode dosen nama dosen dan ruang merupakan field yang ditampilkan pada output jadwal yang sudah terbentuk. Data tersebut diambil dari database jadwalKuliah.

### 3.5. Rancangan Uji Coba

Proses ini dilakukan secara manual dengan membandingkan setiap proses perubahan waktu. Mulai dari aplikasi tingkat awal atau versi 1.0 sampai dengan aplikasi dengan beberapa tambahan seperti penggunaan struktur data, GPU dan *multithread*.

**Tabel 3.3 Catatan Waktu Proses Komputasi**

Versi aplikasi	Waktu
Versi 1.0.0 (belum ada tambahan)	A
Versi n (dengan fitur tambahan)	B

Data dari pencatatan waktu komputasi di Table 3.3, menggambarkan berapa banyak perubahan kecepatan komputasi pada aplikasi yang telah ditambahkan GPU, *multithread* dan struktur data.

Perhitungan *speedup* juga dilakukan untuk menghitung perbandingan proses sekuensial dengan proses paralel pada *multithread* dengan rumus : (Almutairi & Alruwaili, 2012)

$$Speedup = \frac{\text{execution time of sequential algorithm}}{\text{execution time of parallel algorithm}}$$

Selain menghitung efisiensi waktu, pada penelitian ini juga menghitung banyaknya jadwal yang bisa tersusun dengan beberapa kondisi, seperti jumlah ruang yang dikurangi.

Uji coba pada penelitian ini dilakukan dengan beberapa tahap, seperti :

- 1) Tahap uji coba aplikasi penjadwalan versi 1.0 pada komputer dan laptop.
- 2) Tahap uji coba aplikasi penjadwalan dengan GPU pada komputer dan laptop.

- 3) Tahap uji coba aplikasi penjadwalan dengan *Multithread* pada komputer dan laptop.

Untuk proses perhitungan waktu komputasi tahap 1, 2 dan 3 semua dilakukan dengan kriteria yang sama seperti, jumlah data *input*, data jadwal, dan spesifikasi *hardware* untuk ujicoba. Proses perhitungan waktu memiliki beberapa proses :

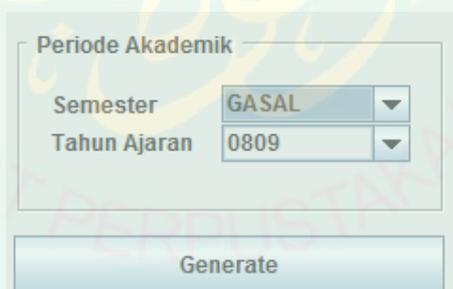
- Input data preferensi mengajar dengan memilih nama dosen, hari yang akan dipilih, semester dan tahun ajaran
- Input data plotting mengajar dengan menginput semester, tahun ajaran, nama dosen, nama matakuliah, kelas dan kapasitas kelas
- Data yang digunakan menggunakan database dengan kondisi kosong dengan kondisi database telah terisi dengan jadwal pada tahun-tahun sebelumnya.
- Menjalankan program dengan jumlah ruang yang berbeda-beda, yakni lima, enam dan tujuh jumlah ruang
- Menjalankan program penjadwalan perkuliahan
- Mencatat waktu eksekusi program setelah program berhenti

Pada proses perhitungan *speedup* hanya menggunakan tahap 1 dan 3 untuk menghitung perbandingan waktu. Proses *speedup* dilakukan dengan menambah beberapa kriteria data. Berikut proses uji coba *speedup* :

- Input data preferensi mengajar dengan memilih nama dosen, hari yang akan dipilih, semester dan tahun ajaran
- Input data plotting mengajar dengan menginput semester, tahun ajaran, nama dosen, nama matakuliah, kelas dan kapasitas kelas
- Menambah data jadwal kuliah mulai dari 0, 2140, 4280, dan 8560.
- Menjalankan program penjadwalan perkuliahan
- Mencatat waktu eksekusi program setelah program berhenti

### 3.6. Desain Interface

Desain interface pada aplikasi versi 1.0 dengan aplikasi yang akan dibangun tidak berbeda. Penggunaan panel *interface* untuk proses penginputan data plot mengajar, data preferensi mengajar, *generate* jadwal dan periode akademik. Berikut merupakan gambar desain interface :



Periode Akademik	
Semester	GASAL
Tahun Ajaran	0809
Generate	

**Gambar 3.14 Panel Periode Akademik**

Gambar 3.14 merupakan *interface* dari panel periode akademik yang berfungsi untuk memproses *generate* jadwal. Ada dua input yang harus dipilih yaitu semester dan tahun ajaran yang digunakan sebagai penentu kategori jadwal.

Plotting Pengajar

Jurusan: PENDIDIKAN AGAMA ISLAM

Nama Matakuliah: [Empty]

Kelas / Kapasitas: A 10

Dosen Pengampu: [Empty]

Save

**Gambar 3.15 Panel Plotting Pengajar**

Gambar 3.15 merupakan *interface* dari panel plotting pengajar yang berfungsi untuk mengisi data plot mengajar. Ada 5 input yang harus dipilih ketika ingin menyimpan data plot mengajar. Jurusan, nama matakuliah, kelas, kapasitas, dan dosen pengampu. Proses *generate* jadwal membutuhkan satu pilihan inputan yang ada di panel plotting pengajar yaitu jurusan.

Preferensi Mengajar

Nama Dosen: A. GHANAIM FASYA,

Preferensi:

- Senin
- Selasa
- Rabu
- Kamis
- Jumat
- Sabtu

Save

**Gambar 3.16 Panel Prefensi Mengajar**

Gambar 3.16 merupakan *interface* dari panel preferensi mengajar yang berfungsi untuk mengisi data preferensi. Ada 1 input yang harus diisi yaitu nama dosen dan ditambah hari yang sesuai dengan pilihan dosen.



## BAB IV

### UJI COBA DAN HASIL

Pada bagian ini dijelaskan mengenai implementasi dari setiap langkah-langkah dalam ujicoba dan memaparkan hasil percobaan yang telah dilakukan. Hasil dari uji coba penelitian yang telah dilakukan merupakan output aplikasi penjadwalan versi 1.0 yang telah di efisiensi waktunya menggunakan GPU ataupun *multithread*. Proses pembuatan jadwal menggunakan metode pencarian dengan penambahan validasi konstrain. Melalui penelitian ini diharapkan aplikasi penjadwalan yang awalnya berjalan 45 menit bisa lebih cepat, sehingga lebih efisien waktu dalam pembuatan jadwal perkuliahan.

#### 4.1. Deskripsi Program

Program aplikasi penjadwalan versi 1.0 masih menggunakan koneksi database yang dalam pembangkitan database memerlukan waktu lama, yaitu sekitar 45 menit. Pada penelitian ini aplikasi versi 1.0 akan dilakukan efisiensi waktu dalam proses komputasi dengan tambahan penggunaan struktur data *arraylist*, GPU dan *multithread*. Penggunaan *arraylist* bertujuan untuk mengganti koneksi langsung dari database menjadi ke memori.

Data yang digunakan merupakan data perkuliahan jurusan teknik informatika yang berisi tentang data dosen, matakuliah, waktu perkuliahan, hari perkuliahan, dan ruang perkuliahan. Jadwal perkuliahan akan terbentuk ketika semua data tersebut ditambah dengan data plot mengajar saling terhubung. Data

uji jadwal pada penelitian ini sekitar 2140 baris data yang sudah tersimpan dan 131 baris data plot mengajar.

Penelitian ini menggunakan data perkuliahan untuk jurusan Teknik Informatika tahun ajaran 2013/2014 semester ganjil. Jurusan selain Teknik Informatika memiliki kriteria tersendiri untuk penyusunan jadwal sehingga masih belum bisa diujicobakan pada aplikasi ini.

Proses pada saat mencari slot kosong, setiap plot mengajar masih menggunakan komputasi *sequence*. Program mengambil setiap data yang dibutuhkan, kemudian secara bergantian masuk ke dalam setiap proses. Lamanya komputasi dikarenakan proses penjadwalan berjalan secara *sequence* dan ditambah dengan koneksi langsung ke database.

## 4.2. Hasil Implementasi Sistem

Hasil implementasi sistem dilakukan untuk mengintegrasikan metode yang digunakan dalam penelitian ini ke dalam langkah – langkah yang tersusun rapi sehingga diperoleh hasil yang dapat diukur tingkat akurasi. Langkah – langkah ini direncanakan dan kemudian dijalankan, sehingga dapat diperoleh hasil yang akurat untuk mengukur tingkat keberhasilan penelitian. Berikut beberapa tahapan dalam proses implementasi sistem pada penelitian dapat diuraikan sebagai berikut:

### 4.2.1. Hasil Implementasi Kelas Aturan

Kelas aturan pada aplikasi versi 1.0 memiliki 2 *method*, yaitu *method* *contstrain1* dan *method* *konstrain2* yang berguna sebagai pegecekan konstrain ruang dan dosen. Setelah dilakukuan uji coba, proses validasi konstrain

dilakukan pada 1 *method* dengan menggabungkan ruang dan dosen dalam satu kali validasi. Berikut perubahan kode dari versi 1.0 ke aplikasi :

Versi 1.0	Hasil
<pre> if(hari.equals(h) &amp;&amp;   jammu.getTime()&gt;=ljm &amp;&amp;   jammu.getTime()&lt;lja &amp;&amp;   ruang.equals(r) &amp;&amp;   tahunajaran.equals(t) &amp;&amp;   semester.equals(s)) {     test=true;     break; }else if(hari.equals(h) &amp;&amp;   jammu.getTime()&lt;lja &amp;&amp;   jamak.getTime()&gt;ljm &amp;&amp;   ruang.equals(r) &amp;&amp;   tahunajaran.equals(t) &amp;&amp;   semester.equals(s)) {     test=true;     break; }else if(hari.equals(h) &amp;&amp;   ljm&gt;=jamdmu.getTime() &amp;&amp;   lja&lt;=jamak.getTime() &amp;&amp;   ruang.equals(r) &amp;&amp;   tahunajaran.equals(t) &amp;&amp;   semester.equals(s)) {     test=true;     break; } </pre>	<pre> if ((hari.equals(h) &amp;&amp;   jammu.getTime() &gt;= ljm &amp;&amp;   jammu.getTime() &lt; lja &amp;&amp;   tahunajaran.equals(t) &amp;&amp;   semester.equals(s)) &amp;&amp;   (ruang.equals(r)      kodedosen.equals(k))) {     test = true;     break; }else if((hari.equals(h) &amp;&amp;   jammu.getTime() &lt; lja   &amp;&amp;   jamak.getTime() &gt; ljm   &amp;&amp;   tahunajaran.equals(t)   &amp;&amp; semester.equals(s))   &amp;&amp; (ruang.equals(r)      kodedosen.equals(k))) {     test = true;     break; } else if((hari.equals(h)&amp;&amp;ljm &gt;=   jammu.getTime() &amp;&amp;   lja&lt;=jamak.getTime()&amp;&amp;   tahunajaran.equals(t)   &amp;&amp;   semester.equals(s))   &amp;&amp; (ruang.equals(r)        kodedosen.equals(k))) {     test = true;     break; } </pre>

Kolom versi 1.0 merupakan *method* untuk validasi konstrain ruang, sedangkan pada kolom hasil, validasi ruang dan dosen dijadikan satu. Hasil ini dilakukan untuk mempersingkat dalam validasi konstrain, selain itu pada aplikasi GPU data yang dikirim ke karnel lebih mudah.

#### 4.2.2. Hasil Implementasi *Arraylist*

Penggunaan *arraylist* berfungsi untuk menyimpan data sementara pada memori dan proses eksekusi komputasi pada aplikasi. Langkah pertama untuk implementasi *arraylist* adalah pemindahan data satu persatu dari *database* ke *arraylist*. Data dari *database* yang dirubah ke dalam memori meliputi ruang, preferensi, jam kuliah, plot mengajar, dan jadwal kuliah berikut merupakan contoh hasil implementasi *arraylist* jadwal:

Versi 1.0	Hasil
<pre>String sql2="SELECT * from   JadwalKuliah where KodeMK ='"+kodematakuliah+"' and   JurKode='"+ kodejurusan+"'   and Semester='"+semester+"'   and TahunAjaran='"+   tahunajaran +" 'and Kelas =   '"+kelas+"'"; ResultSet rs2=statement2.   executeQuery(sql2 if(rs2.next() {  cekkodemk=rs2.getString(1);  cekjurkode=rs2.getString(2); cektahunajaran=rs2.   getString(3); ceksemester=rs2.   getString(4); cekkelas=rs2.getString(5); }</pre>	<pre>for(int jk=0; jk&lt; mem.   lst_jadwal_kuliah.size()   ;jk++){ if(mem.lst_jadwal_kuliah.   get(jk).kodeMK.contains   (kodematakuliah) &amp;&amp;   mem.lst_jadwal_kuliah.   get(jk).jurKode.contains   (kodejurusan) &amp;&amp;   mem.lst_jadwal_kuliah.   get(jk).semester.   contains(smt) &amp;&amp;   mem.lst_jadwal_kuliah.   get(jk).tahunAjaran.   contains(thn) &amp;&amp;   mem.lst_jadwal_kuliah.   get(jk).kelas.contains   (kelas)){   cekkodek=mem.     lst_jadwal_kuliah.get     (jk).kodeMK;   cekjurkode=mem.     lst_jadwal_kuliah.get     (jk).jurKode;   cektahunajaran=mem.     lst_jadwal_kuliah.get     (jk).tahunAjaran;   ceksemester=mem.     lst_jadwal_kuliah.get     (jk).semester;   cekkelas=mem.     lst_jadwal_kuliah.get     (jk).kelas; } }</pre>
<pre>sql="insert into JadwalKuliah</pre>	<pre>mem.lst_jadwal_kuliah.add (new listJadwalKuliah (kodematakuliah,</pre>

<pre>values ('"+kodematakuliah+"',         '"+kodejurusan+"', '"+ tahunajaran+"', '"+semester+ "', '"+kelas+"', '"+kodedosen+ "', '"+haripref[k]+"', '"+ jammulai+"', '"+jamakhir+"',         '"+ruangkelas[i]+"', '"+         kapasitaskelas+"');"; statement.executeUpdate(sql) ;</pre>	<pre>kodejurusan, tahunajaran, semester, kelas, kodedosen, pref.get(k), jammulai, jamakhir, ruang.get(i), kapasitaskelas));</pre>
<pre>String sql="select Hari, JamMulai, JamAkhir, Ruang, TahunAjaran, Semester from JadwalKuliah"; ResultSet rs=statement. executeQuery(sql); while(rs.next()){     h=rs.getString(1);     jm=rs.getString(2);     ja=rs.getString(3);     r=rs.getString(4);     t=rs.getString(5);     s=rs.getString(6);     ljm=sdf.parse(jm).         getTime();     lja=sdf.parse(ja).         getTime(); }</pre>	<pre>for(int i = 0; i &lt; mem.lst_jadwal_kuliah. size(); i++) {     h=mem.lst_jadwal_kuliah.         get(i).hari;     jm=mem.lst_jadwal_kuliah.         get(i).jamMulai;     ja=mem.lst_jadwal_kuliah.         get(i).jamAkhir;     r=mem.lst_jadwal_kuliah.         get(i).ruang;     t=mem.lst_jadwal_kuliah.         get(i).tahunAjaran;     s=mem.lst_jadwal_kuliah.         get(i).semester;     ljm=sdf.parse(jm).         getTime();     lja=sdf.parse(ja).         getTime(); }</pre>

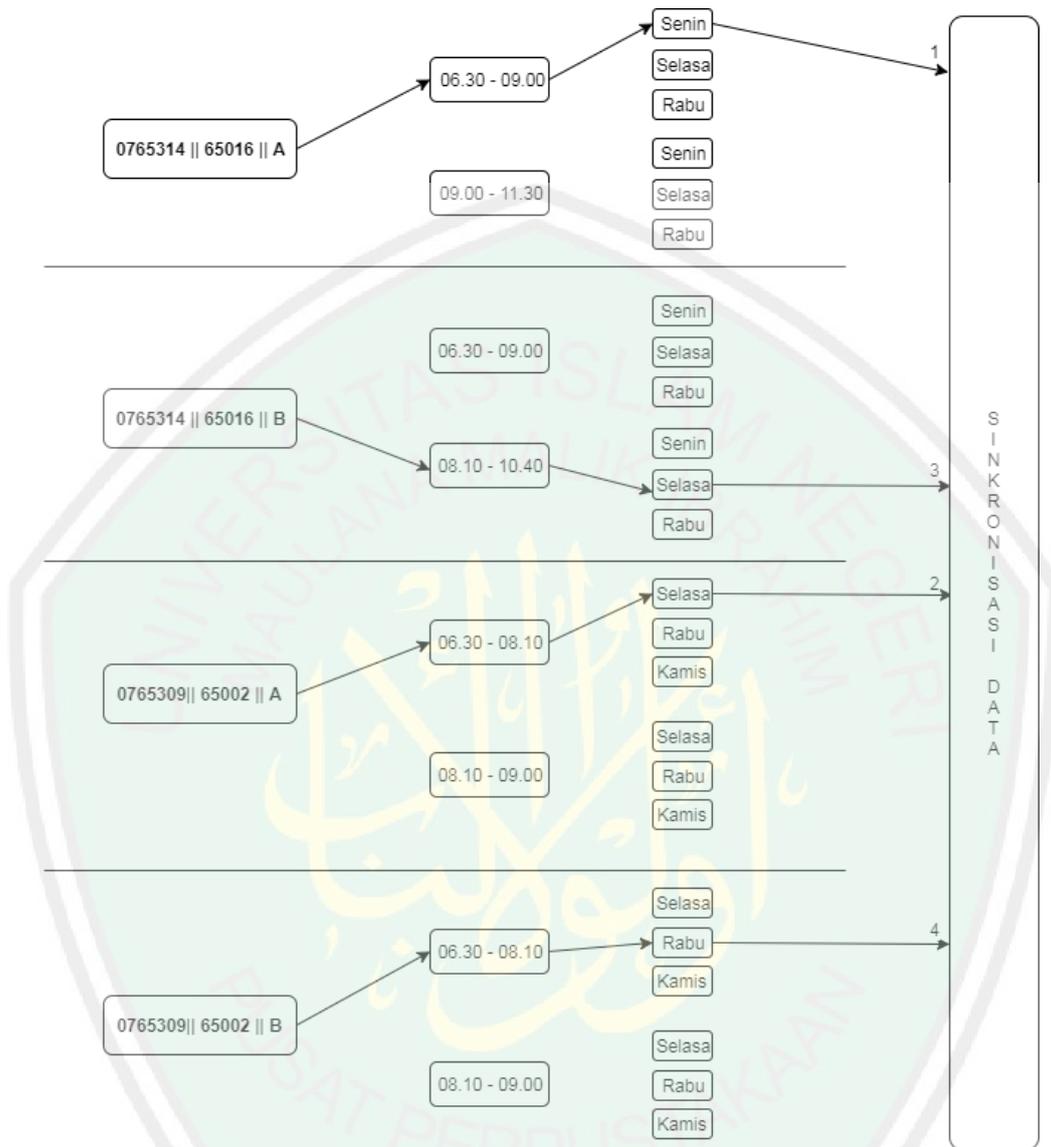
Kode diatas merupakan contoh penerapan *arraylist* yang merubah fungsi pemanggilan data dari database. Kolom 1 menunjukan bahwa perbedaan pengambilan data jadwal kuliah versi 1.0 dengan hasil. Kode tersebut digunakan sebagai data pembanding jadwal matakuliah di *genJadwal*. Kolom 2 menampilkan perbedaan penyimpanan data jadwal yang akan digunakan untuk proses selanjutnya atau untuk menyimpan data jadwal yang sudah terjadwal. Kode pada kolom 3 merupakan fungsi yang terdapat di kelas *Aturan* dan digunakan sebagai proses validasi konstrain jadwal yang sudah terbentuk.

#### 4.2.3. Hasil Implementasi *Multithread*

Uji coba dengan *multithread* menggunakan 4 data dari plot mengajar yang diambil dari database *plotmengajar* seperti Tabel 3.1. Dari uji coba tersebut didapat bahwa urutan jadwal hasil *generate* jadwal tergantung data yang selesai terlebih dahulu dikerjakan oleh *multithread*. Ketika satu data *thread* sudah selesai melakukan pencarian, maka akan melakukan sinkronisasi data.

Penjelasan tentang hasil jalannya ujicoba digambarkan oleh Gambar 4.1 yang menerangkan bahwa setiap data plot mengajar dikerjakan oleh satu *thread* kemudian baru dilakukan sinkronisasi. Sehingga data yang tersimpan di *arraylist* bisa digunakan oleh *thread* lain yang sedang berjalan. Sinkronisasi data ini bertujuan untuk mengirim data jadwal kuliah yang sudah tersimpan dan selesai pada proses pencarian slot.

Proses penjadwalan ini sudah memenuhi beberapa konstrain seperti samanya hari preferensi mengajar dosen dengan jadwal yang terbentuk. Dosen tidak ada bentrok jam dan ruang ketika mengajar sehingga jadwal sesuai. Data ini menunjukkan bahwa hasil dari aplikasi dengan *multithread* sama dengan hasil dari versi 1.0 dalam upaya penerapan konstrain yang sudah ada dan jumlah jadwal yang terbentuk.



**Gambar 4.1 Data Ujicoba Multithread**

Dari Gambar 4.1 data plot pertama yang selesai melakukan pencarian adalah plot data 0765314||65016||A dan mengisi slot ruang B.306, slot jam 06.30 sampai 09.00 dan slot hari Senin. Data tersebut langsung diisi dengan plot tersebut karena masih kosong dan tidak terjadi bentrok.

Data plot selanjutnya yang selesai melakukan pencarian adalah plot data 0765314||65002||A dan mengisi slot ruang B.306, slot jam 06.30 sampai 08.10

dan slot hari Selasa. Data tersebut langsung diisi dengan plot tersebut karena masih kosong dan tidak terjadi bentrok.

Data plot selanjutnya yang selesai melakukan pencarian adalah plot data 0765309||65016||B dan mengisi slot ruang B.306, slot jam 08.10 sampai 10.40 dan slot hari Selasa. Menurut referensi mengajar dari dosen 65016 sudah sesuai yaitu hari Selasa karena hari Senin sudah terpakai. Untuk slot jam data plot 0765309||65016||B mengisi slot jam 08.10 karena untuk jam 06.30 diisi dengan 0765314||65002||A yang sudah selesai terlebih dahulu. Proses tersebut dilakukan karena telah melakukan sinkronisasi data.

Data plot terakhir yang selesai melakukan pencarian adalah plot data 0765314||65002||B dan mengisi slot ruang B.306, slot jam 06.30 sampai 08.10 dan slot hari Rabu. Data tersebut langsung diisi dengan plot tersebut karena masih kosong dan tidak terjadi bentrok.

Dari uji coba diatas diketahui bahwa urutan jadwal proses *multithread* dipengaruhi oleh data yang sudah selesai terlebih dahulu. Masing-masing kelas *thread* melakukan proses sinkronisasi data yang berguna untuk mengetahui slot-slot jadwal yang sudah terisi supaya tidak terjadi bentrok data. Berikut Gambar 4.2 merupakan hasil dari ujicoba 4 plot mengajar pada aplikasi versi 1.1 dengan *multithread*

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Senin	06.30 - 09.00	0785314	SISTEM INFORMASI GEOGRAFIS	3	A	85018	Fahrul Kurniawan M.MT	B.306
Selasa	06.30 - 08.10	0785309	REKAYASA PERANGKAT LUNAK	2	A	85002	FATCHURROCHMAN, M.Kom	B.306
Selasa	08.10 - 10.40	0785314	SISTEM INFORMASI GEOGRAFIS	3	B	85018	Fahrul Kurniawan M.MT	B.306
Rabu	06.30 - 08.10	0785309	REKAYASA PERANGKAT LUNAK	2	B	85002	FATCHURROCHMAN, M.Kom	B.306

**Gambar 4.2 Interface Hasil Ujicoba Multithread**

### 4.3. Pengujian Sistem Tahap Waktu Komputasi

Setelah melakukan integrasi metode komputasi paralel ke dalam langkah-langkah yang terencana dan tersusun rapi, sehingga dapat diukur tingkat akurasi efisiensi komputasi. Langkah-langkah metode yang sudah direncanakan kemudian dilakukan agar dapat menghitung tingkat keberhasilan penelitian.

Pada subbab ini menampilkan hasil pengujian dari penelitian yang telah dilakukan. Pengujian yang dilakukan menggunakan komputasi paralel dengan GPU dan komputasi paralel dengan *multithread*. Kemudian dihitung waktu yang dibutuhkan untuk proses komputasi.

#### 4.3.1. Uji Coba dengan GPU

Uji coba aplikasi dengan GPU menggunakan data 131 plot mengajar, 2140 data jadwal dengan kriteria tahun ajaran 2013/2014, semester ganjil, dan jurusan teknik informatika. Spesifikasi komputer yang digunakan untuk uji coba aplikasi

adalah laptop lenovo G480, memory 4 giga byte, dan Intel ® HD Graphics 4000.

Berikut output dari uji coba :

1. Uji coba dengan data ruang sebanyak 7, data plot mengajar 131, data jadwal 2140 dengan laptop lenovo G480.

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Senin	06.30 - 09.00	0765103	BAHASA INGGRIS I	3	A	65111	Robert Aprillianto M.Pd	B.306
Senin	09.00 - 09.50	0765105	ILMU SOSIAL DASAR	1	A	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	09.50 - 10.40	0765105	ILMU SOSIAL DASAR	1	C	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	10.40 - 11.30	0765106	ILMU BUDAYA DASAR	1	A	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	12.20 - 13.10	0765106	ILMU BUDAYA DASAR	1	C	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	13.10 - 14.50	0765108	STUDI AL-QUR AN	2	C	65017	Imamudin M.A	B.306
Senin	14.50 - 16.30	0765110	STUDI FIQH	2	D	65151	Muhammad Karim M.Pd	B.306
Senin	06.30 - 08.10	0765113	TEOLOGI ISLAM	2	A	60002	AHMAD BARIZI, M.A	B.307
Senin	08.10 - 09.50	0765113	TEOLOGI ISLAM	2	C	60002	AHMAD BARIZI, M.A	B.307
Senin	09.50 - 11.30	0765207	METEDOLOGI PENELITIAN	2	A	65005	SYAHIDUZ ZAMAN M.Kom	B.307
Senin	12.20 - 14.00	0765208	PENGANTAR TEKNOLOGI INFOR...	2	A	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.307
Senin	14.00 - 15.40	0765208	PENGANTAR TEKNOLOGI INFOR...	2	B	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.307
Senin	06.30 - 09.00	0765214	DESAIN BASIS DATA	3	A	65130	Ainatul M M.Kom	B.314
Senin	09.00 - 11.30	0765214	DESAIN BASIS DATA	3	C	65130	Ainatul M M.Kom	B.314
Senin	12.20 - 14.50	0765214	DESAIN BASIS DATA	3	D	65130	Ainatul M M.Kom	B.314

**Gambar 4.3 Hasil Ujicoba GPU dengan 7 Ruangan**

Gambar 4.3. merupakan output dari uji coba dengan menghasilkan program matakuliah yang terjadwal 131 dan waktu sekitar 3566 detik.

2. Uji coba dengan data ruang sebanyak 6, data plot mengajar 131, data jadwal 2140 dengan laptop lenovo G480.

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Senin	06.30 - 09.00	0765103	BAHASA INGGRIS I	3	A	65111	Robert Aprillianto M.Pd	B.306
Senin	09.00 - 09.50	0765105	ILMU SOSIAL DASAR	1	A	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	09.50 - 10.40	0765105	ILMU SOSIAL DASAR	1	C	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	10.40 - 11.30	0765106	ILMU BUDAYA DASAR	1	A	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	12.20 - 13.10	0765106	ILMU BUDAYA DASAR	1	C	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	13.10 - 14.50	0765108	STUDI AL-QUR AN	2	C	65017	Imamudin M.A	B.306
Senin	14.50 - 16.30	0765110	STUDI FIQH	2	D	65151	Muhammad Karim M.Pd	B.306
Senin	06.30 - 08.10	0765113	TEOLOGI ISLAM	2	A	60002	AHMAD BARIZI, M.A	B.307
Senin	08.10 - 09.50	0765113	TEOLOGI ISLAM	2	C	60002	AHMAD BARIZI, M.A	B.307
Senin	09.50 - 11.30	0765207	METEDOLOGI PENELITIAN	2	A	65005	SYAHIDUZ ZAMAN M.Kom	B.307
Senin	12.20 - 14.00	0765208	PENGANTAR TEKNOLOGI INFOR...	2	A	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.307
Senin	14.00 - 15.40	0765208	PENGANTAR TEKNOLOGI INFOR...	2	B	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.307
Senin	06.30 - 09.00	0765214	DESAIN BASIS DATA	3	A	65130	Ainatul M M.Kom	B.314
Senin	09.00 - 11.30	0765214	DESAIN BASIS DATA	3	C	65130	Ainatul M M.Kom	B.314

**Gambar 4.4 Hasil Ujicoba GPU dengan 6 Ruangan**

Gambar 4.4 merupakan output dari uji coba dengan menghasilkan program matakuliah yang terjadwal 131 dan waktu sekitar 3560 detik.

- Uji coba dengan data ruang sebanyak 5, data plot mengajar 131, data jadwal 2140 dengan laptop lenovo G480

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Senin	06.30 - 09.00	0765103	BAHASA INGGRIS I	3	A	65111	Robert Aprillianto M.Pd	B.306
Senin	09.00 - 09.50	0765105	ILMU SOSIAL DASAR	1	A	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	09.50 - 10.40	0765105	ILMU SOSIAL DASAR	1	C	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	10.40 - 11.30	0765106	ILMU BUDAYA DASAR	1	A	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	12.20 - 13.10	0765106	ILMU BUDAYA DASAR	1	C	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	13.10 - 14.50	0765108	STUDI AL-QUR AN	2	C	65017	Imamudin M.A	B.306
Senin	14.50 - 16.30	0765110	STUDI FIQH	2	D	65151	Muhammad Karim M.Pd	B.306
Senin	06.30 - 08.10	0765113	TEOLOGI ISLAM	2	A	60002	AHMAD BARIZI, M.A	B.307
Senin	08.10 - 09.50	0765113	TEOLOGI ISLAM	2	C	60002	AHMAD BARIZI, M.A	B.307
Senin	09.50 - 11.30	0765207	METEDOLOGI PENELITIAN	2	A	65005	SYAHIDUZ ZAMAN M.Kom	B.307
Senin	12.20 - 14.00	0765208	PENGANTAR TEKNOLOGI INFOR...	2	A	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.307
Senin	14.00 - 15.40	0765208	PENGANTAR TEKNOLOGI INFOR...	2	B	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.307
Senin	06.30 - 09.00	0765214	DESAIN BASIS DATA	3	A	65130	Ainatul M M.Kom	B.314
Senin	09.00 - 11.30	0765214	DESAIN BASIS DATA	3	C	65130	Ainatul M M.Kom	B.314

**Gambar 4.5 Hasil Ujicoba GPU dengan 5 Ruang**

Gambar 4.5 merupakan output dari uji coba dengan menghasilkan program matakuliah yang terjadwal 111 dan waktu sekitar 3554 detik.

#### 4.3.2. Uji Coba dengan *Multithread*

Uji coba dengan *multithread* menggunakan data 131 plot mengajar dan 2140 jadwal dengan kriteria sama dengan uji coba pada GPU. Susunan jadwal yang terbentuk berbeda dengan uji coba aplikasi versi 1.0 dan GPU, hasil dari uji coba *multithread* berbeda-beda pada setiap *generate* jadwal. Waktu yang dibutuhkan untuk komputasi pada *multithread* juga berbeda dari GPU dan aplikasi versi 1.0, lebih cepat daripada 2 ujicoba sebelumnya. Berikut output dari uji coba dengan *multithread* :

- Uji coba dengan data ruang sebanyak 7, data plot mengajar 131, data jadwal 2140 dengan laptop lenovo G480

**Tabel 4.1 Ujicoba Multithread dengan 7 Ruangan**

No Percobaan	Waktu (s)	Jadwal
1	3.264	131
2	2.96	131
3	3.168	131
4	3.254	131
5	3.191	131

Kesimpulan dari Tabel 4.1 yang melakukan 5 kali uji coba dengan 7 ruangan jumlah jadwal yang berhasil terjadwal sebanyak 131 dari 131 plot mengajar dan dengan rata – rata waktu 3.1674 detik.

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Senin	06.30 - 09.00	0765103	BAHASA INGGRIS I	3	A	65111	Robert Aprillianto M.Pd	B.306
Senin	09.00 - 10.40	0765113	TEOLOGI ISLAM	2	D	60002	AHMAD BARIZI, M.A	B.306
Senin	10.40 - 11.30	0765106	ILMU BUDAYA DASAR	1	B	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	12.20 - 14.00	0765113	TEOLOGI ISLAM	2	A	60002	AHMAD BARIZI, M.A	B.306
Senin	14.00 - 15.40	0765110	STUDI FIOH	2	D	65151	Muhammad Karim M.Pd	B.306
Senin	15.40 - 16.30	0765105	ILMU SOSIAL DASAR	1	B	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	06.30 - 08.10	0765110	STUDI FIOH	2	C	65151	Muhammad Karim M.Pd	B.307
Senin	08.10 - 09.50	0765108	STUDI AL-QUR AN	2	A	65017	Imamudin M.A	B.307
Senin	09.50 - 11.30	0765108	STUDI AL-QUR AN	2	C	65017	Imamudin M.A	B.307
Senin	12.20 - 13.10	0765106	ILMU BUDAYA DASAR	1	C	13014	NI MATUZ ZUHROH, M.Si	B.307
Senin	13.10 - 14.00	0765105	ILMU SOSIAL DASAR	1	C	13014	NI MATUZ ZUHROH, M.Si	B.307
Senin	14.00 - 15.40	0765208	PENGANTAR TEKNOLOGI INFOR...	2	A	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.307
Senin	06.30 - 09.00	0765214	DESAIN BASIS DATA	3	A	65130	Ainatul M M Kom	B.314
Senin	09.00 - 10.40	0765215	SITEM OPERASI	2	D	65009	null SATRIA MANDALA, MCS	B.314
Senin	12.20 - 14.50	0765303	PENGOLAHAN CITRA	3	C	65015	Dr Cahyo Crysdian	B.314
Senin	14.50 - 16.30	0765310	INTERAKSI MANUSIA & KOMPUT...	2	A	65005	SYAHIDUZ ZAMAN M.Kom	B.314
Senin	06.30 - 09.00	0765303	IPENGOLAHAN CITRA	3	B	65015	Dr Cahyo Crysdian	B.315

**Gambar 4.6 Contoh Output Jadwal Multithread dengan 7 Ruangan**

Gambar 4.6 merupakan output sample dari uji coba *multithread* dengan 7 ruang. Hasil jadwal dari 5 kali percobaan untuk urutan jadwal berbeda-beda setiap kali *generate*.

2. Uji coba dengan data ruang sebanyak 6, data plot mengajar 131, data jadwal 2140 dengan laptop lenovo G480

**Tabel 4.2 Ujicoba Multithread dengan 6 Ruangan**

No Percobaan	Waktu (s)	Jadwal
1	2.591	130
2	2.796	129
3	2.724	129
4	2.694	129
5	2.741	129

Kesimpulan dari Tabel 4.2 yang melakukan 5 kali uji coba dengan 6 ruangan jumlah jadwal yang berhasil terjadwal rata-rata sebanyak 129 dari 131 plot mengajar dan dengan rata – rata waktu 2.7092 detik.

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Senin	06.30 - 09.00	0765103	BAHASA INGGRIS I	3	A	65111	Robert Aprillianto M.Pd	B.306
Senin	09.00 - 10.40	0765402	ISU PROFESIONAL & PROYEK G...	2	B	65021	Ali Wafa M.Kom	B.306
Senin	10.40 - 11.30	0765105	ILMU SOSIAL DASAR	1	A	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	12.20 - 14.50	0765314	SISTEM INFORMASI GEOGRAFIS	3	A	65018	Fahrul Kurniawan M.MT	B.306
Senin	14.50 - 16.30	0765215	SITEM OPERASI	2	C	65009	null SATRIA MANDALA, MCS	B.306
Senin	06.30 - 09.00	0765305	PEMROGRAMAN MULTIMEDIA	3	D	65019	Fressy MT	B.307
Senin	09.00 - 11.30	0765314	SISTEM INFORMASI GEOGRAFIS	3	B	65018	Fahrul Kurniawan M.MT	B.307
Senin	12.20 - 14.50	0765214	DESAIN BASIS DATA	3	C	65130	Ainatul M M.Kom	B.307
Senin	14.50 - 16.30	0765208	PENGANTAR TEKNOLOGI INFOR...	2	B	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.307
Senin	06.30 - 08.10	0765208	PENGANTAR TEKNOLOGI INFOR...	2	C	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.314
Senin	08.10 - 09.00	0765106	ILMU BUDAYA DASAR	1	A	13014	NI MATUZ ZUHROH, M.Si	B.314
Senin	09.00 - 10.40	0765108	STUDI AL-QUR AN	2	A	65017	Imamudin M.A	B.314
Senin	12.20 - 14.00	0765109	STUDI AL-HADITS	2	A	65199	Sugeng Ali Mansyur M.Pd	B.314
Senin	14.00 - 15.40	0765110	STUDI FIQH	2	A	65151	Muhammad Karim M.Pd	B.314
Senin	15.40 - 16.30	0765106	ILMU BUDAYA DASAR	1	B	13014	NI MATUZ ZUHROH, M.Si	B.314
Senin	06.30 - 08.10	0765113	TEOLOGI ISLAM	2	B	60002	AHMAD BARIZI, M.A	B.315
Senin	09.00 - 09.50	0765105	ILMU SOSIAL DASAR	1	D	13014	NI MATUZ ZUHROH, M.Si	B.315

#### Gambar 4.7 Contoh Output Jadwal Multithread dengan 6 Ruangan

Gambar 4.7 merupakan output sample dari uji coba *multithread* dengan 6 ruang. Hasil jadwal dari 5 kali percobaan untuk urutan jadwal berbeda-beda setiap kali *generate*.

3. Uji coba dengan data ruang sebanyak 5, data plot mengajar 131, data jadwal 2140 dengan laptop lenovo G480

Tabel 4.3 Ujicoba Multithread dengan 5 Ruangan

No Percobaan	Waktu (s)	Jadwal
1	2.188	112
2	2.241	112
3	2.457	110
4	2.315	113
5	2.146	112

Kesimpulan dari Tabel 4.3 yang melakukan 5 kali uji coba dengan 5 ruangan, jumlah jadwal yang berhasil terjadwal rata-rata sebanyak 111 dari 131 plot mengajar dan dengan rata – rata waktu 2.2694 detik.

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Senin	06.30 - 09.00	0765103	BAHASA INGGRIS I	3	A	65111	Robert Aprillianto M.Pd	B.306
Senin	10.40 - 11.30	0765106	ILMU BUDAYA DASAR	1	B	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	14.50 - 16.30	0765402	ISU PROFESIONAL & PROYEK G...	2	B	65021	Ali Wafa M.Kom	B.306
Senin	06.30 - 09.00	0765314	SISTEM INFORMASI GEOGRAFIS	3	A	65018	Fahrul Kurniawan M.MT	B.307
Senin	09.00 - 10.40	0765113	TEOLOGI ISLAM	2	B	60002	AHMAD BARIZI, M.A	B.307
Senin	12.20 - 13.10	0765106	ILMU BUDAYA DASAR	1	D	13014	NI MATUZ ZUHROH, M.Si	B.307
Senin	13.10 - 14.50	0765108	STUDIAL-QUR AN	2	D	65017	Imamudin M.A	B.307
Senin	14.50 - 16.30	0765113	TEOLOGI ISLAM	2	A	60002	AHMAD BARIZI, M.A	B.307
Senin	06.30 - 08.10	0765113	TEOLOGI ISLAM	2	C	60002	AHMAD BARIZI, M.A	B.314
Senin	08.10 - 09.50	0765208	PENGANTAR TEKNOLOGI INFOR...	2	A	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.314
Senin	09.50 - 11.30	0765208	PENGANTAR TEKNOLOGI INFOR...	2	B	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.314
Senin	12.20 - 14.50	0765214	DESAIN BASIS DATA	3	A	65130	Ainatul M M.Kom	B.314
Senin	14.50 - 16.30	0765215	SITEM OPERASI	2	E	65009	null SATRIA MANDALA, MCS	B.314
Senin	06.30 - 08.10	0765310	INTERAKSI MANUSIA & KOMPUT...	2	B	65005	SYAHIDUZ ZAMAN M.Kom	B.315
Senin	09.00 - 11.30	0765314	SISTEM INFORMASI GEOGRAFIS	3	E	65018	Fahrul Kurniawan M.MT	B.315
Senin	12.20 - 14.00	0765402	ISU PROFESIONAL & PROYEK G...	2	E	65021	Ali Wafa M.Kom	B.315
Senin	14.00 - 15.40	0765207	METEDOLOGI PENELITIAN	2	A	65005	SYAHIDUZ ZAMAN M.Kom	B.315

**Gambar 4.8 Contoh Output Jadwal Multithread dengan 5 Ruangan**

Gambar 4.8 merupakan output sample dari uji coba *multithread* dengan 5 ruang. Hasil jadwal dari 5 kali percobaan untuk urutan jadwal berbeda-beda setiap kali *generate*.

#### 4.4. Pengujian Sistem Tahap *Speedup*

Pada subbab ini menampilkan hasil pengujian dari penelitian yang telah dilakukan. Pengujian yang dilakukan menggunakan aplikasi penjadwalan versi 1.0 dengan komputasi paralel dengan *multithread*. Kemudian dihitung *speedup* pada keadaan tertentu.

##### 4.4.1. Perbedaan jumlah *thread*

Ujicoba dengan perbedaan jumlah *thread* dilakukan sebanyak 5 kali, dengan kriteria jadwal kuliah 2140, plot mengajar 131 dan ujicoba dilakukan pada perangkat laptop.

**Tabel 4.4 Hasil Ujicoba dengan Perbedaan Jumlah Thread**

Jumlah <i>thread</i>	Waktu eksekusi (detik)
5	3.3088
100	3.2382
131	3.4094
150	3.2354
250	3.1572

Untuk setiap satu data uji pada Tabel 4.4, waktu eksekusi yang didapat merupakan rata-rata dari 5 kali ujicoba. Dari ujicoba pada Tabel 4.4 diketahui bahwa jumlah *thread* berpengaruh pada waktu eksekusi aplikasi. Meskipun perbedaan waktunya tidak terlalu besar.

#### 4.4.2. Perbedaan jumlah data jadwal matakuliah

Ujicoba dengan perbedaan jumlah data jadwal kuliah dilakukan sebanyak 4 kali, dengan jumlah data jadwal kuliah berubah mulai dari 0, 2140, 4280, dan 8560 *record* data. plot mengajar 131 dan jumlah *multithread* sebanyak 131. ujicoba dilakukan pada perangkat laptop. Versi 1.0 merupakan proses komputasi sekuensial yang dilakukan dengan menggunakan akses database Mysql. Sedangkan aplikasi versi 1.0 *arraylist* merupakan pengembangan dari aplikasi versi 1.0 yang proses pengambilan datanya dari *arraylist* dan proses komputasinya berjalan secara sekuensial.

**Tabel 4.5 Hasil Ujicoba Multithread dengan Perbedaan Komputasi**

Jumlah data matakuliah	Waktu eksekusi (detik)		
	Versi 1.0	<i>Arraylist</i>	<i>Multithread</i>
0	24.6824	13.1678	3.3088
2140	273.7298	11.1186	3.4094
4280	615.1576	11.4718	3.4374
8560	1145.072	92.037	3.2662

Untuk setiap satu data uji pada Tabel 4.5, waktu eksekusi yang didapat merupakan rata-rata dari 5 kali ujicoba. Data jumlah matakuliah berpengaruh cukup banyak terhadap waktu eksekusi pada aplikasi versi 1.0 yang masih menggunakan database Mysql. Setiap ada penambahan data maka waktu komputasinya juga bertambah. Berbeda dengan aplikasi versi 1.0 dengan *multithread* pada 3 kali ujicoba diawal, estimasi waktu yang dibutuhkan untuk komputasi sekitar 11 detik. Namun pada saat jumlah data matakuliah sebesar 8560 waktu yang dibutuhkan meningkat sekitar 90 detik. Aplikasi versi 1.1 dengan *multithread* waktu yang dibutuhkan untuk proses komputasi sekitar 3 detik untuk seluruh data.

#### 4.5. Pembahasan Sistem

Setelah melakukan uji coba dalam penelitian ini ke dalam langkah-langkah yang terencana dan tersusun rapi sehingga dapat diukur waktu efisiensi komputasi. Langkah-langkah *methode* yang sudah direncanakan dan dilakukan kemudian dapat menghitung tingkat keberhasilan penelitian.

Pada subbab ini menampilkan hasil pengujian dari penelitian yang telah dilakukan. Pengujian yang dilakukan menggunakan GPU dan *multithread* pada 2 komputer yang berbeda serta dibandingkan dengan aplikasi versi 1.0. Sehingga diperoleh waktu efisiensi yang lebih akurat.

##### 4.5.1. Hasil dengan Perbedaan Perangkat

Hasil dengan menggunakan perangkat yang berbeda. Pertama menggunakan komputer dengan spesifikasi intel pentium dual core, RAM 512, dan kedua

menggunakan laptop lenovo G480 core i5, dan RAM 4 GB. Proses ujicoba dilakukan dengan 3 menggunakan aplikasi versi 1.0, aplikasi menggunakan GPU dan aplikasi menggunakan *multithread*.

**Tabel 4.6 Data Ujicoba dengan Laptop**

Uji coba	Jumlah data jadwal		Hasil		Konstrain			
	0 <i>record</i> data	2140 <i>record</i> data	Data terjadwal	Data tidak terjadwal	preferensi	ruang	dosen	salat
Versi 1.0	15 detik	786 detik	131	0	0	0	0	0
GPU	3564 detik	3566 detik	131	0	0	0	0	0
<i>Multithread</i>	3 detik	3 detik	131	0	0	0	0	0

**Tabel 4.7 Data Ujicoba dengan dekstop**

Uji coba	Jumlah data jadwal		Hasil		Konstrain			
	0 <i>record</i> data	2140 <i>record</i> data	Data terjadwal	Data tidak terjadwal	preferensi	ruang	dosen	salat
Versi 1.0	19 detik	991 detik	131	0	0	0	0	0
<i>Multithread</i>	15 detik	15 detik	131	0	0	0	0	0

Dari data Tabel 4.6 menunjukkan hasil dari uji coba yang dilakukan pada perangkat laptop dan diperoleh hasil dengan beberapa kesimpulan. Pada aplikasi versi 1.0 perbedaan kecepatan antara data jadwal yang *record* terisi 2140 dengan data jadwal yang *record* data 0 sangat besar, hal inilah yang mempengaruhi aplikasi belum bisa dikatakan efisien. Aplikasi menggunakan GPU menghasilkan waktu yang relatif lama dibandingkan dengan aplikasi versi 1.0. Karena hal tersebut maka belum bisa dikatakan efisien pada penggunaan aplikasi GPU. Aplikasi yang dibangun dengan *multithread* menghasilkan waktu yang

cepat dengan rata-rata 3 detik. Output yang dikeluarkan oleh *multithread* juga *random*, dalam artian setiap kali generate jadwal, output susunan jadwal yang dikeluarkan berbeda-beda. Data hasil jadwal yang terbentuk antara aplikasi versi 1.0, GPU dan *multithread* sama yaitu 131, dan data yang tidak terjadwal sebanyak 0 *record*. Semua konstrain yang ada pada aplikasi versi 1.0 berjumlah 0 data, yang berarti tidak ada konstrain yang dilanggar. Sama halnya dengan aplikasi GPU dan aplikasi *multithread* yang hasil bentrok konstrain berjumlah 0 data. Sehingga aplikasi dengan menggunakan *multithread* lebih efisien dalam hal kecepatan dibandingkan dengan aplikasi versi 1.0 dan GPU dikarenakan waktu yang lebih cepat.

Data dari Tabel 4.7 merupakan hasil uji coba dengan perangkat komputer, dan hanya 2 aplikasi yang dapat diuji, karena pada komputer tidak tersedia *graphic card* sehingga aplikasi dengan GPU tidak bisa dijalankan. Hasil aplikasi versi 1.0 menghasilkan waktu yang berbeda tergantung dengan jumlah *record* data pada jadwal, hal ini berbeda dengan uji coba menggunakan laptop lenovo G480. Aplikasi menggunakan *multithread* lebih stabil dan cepat dibandingkan dengan aplikasi 1.0. Perbedaan waktu eksekusi antara aplikasi 1.0 dengan *multithread* dengan *record* jadwal kosong sekitar 4 detik, yaitu 19 detik pada aplikasi versi 1.0 dan 15 detik untuk aplikasi *multithread*. Hasil jadwal yang terbentuk antara versi 1.0 dan *multithread* berjumlah sama yaitu 131 dan untuk konstrain yang dilanggar juga bernilai sama yaitu 0 record data.

#### 4.5.2. Pembahasan Hasil Jadwal yang Terbentuk

Hasil urutan jadwal dengan komputasi versi 1.0 dan *multithread* terdapat beberapa perbedaan. Aplikasi versi 1.0 memiliki urutan yang konsisten, dalam artian untuk setiap proses *generate* menghasilkan susunan jadwal yang sama. Sedangkan untuk proses *generate* aplikasi dengan *multithread* susunan jadwal yang terbentuk secara *random*, jadi setiap *generate* jadwal hasilnya berbeda-beda.

Berikut merupakan hasil perbandingan kesesuaian data jadwal versi 1.0 dengan *multithread* dilihat dari konstrain preferensi mengajar setiap dosen :

**Tabel 4.8** Perbandingan Konstrain Preferensi Dosen

Nama Dosen	Preferensi Mengajar	Versi 1.0	Versi <i>Multithread</i>	Keterangan
Fahrul Kurniawan, M.MT	1,2,3	1,1,2,2	1,1,2,3	Berbeda
Ali Wafa, M.Kom	1,2,3	1,1,2,3,3	1,1,2,2,3	Sama
Ni'matus Zuhroh, M.Si	1,2	1,1,1,1,1,2,2,2	1,1,1,1,2,2,2,2	Sama
Robert Aprilianto, M.Pd	1,2,3,4,5	1,3,4,5	1,2,3,4	Berbeda
Imamudin, M.Ag	1,5	1,1,5,5	1,1,1,5	Sama
Ahmad Barizi, M.Ag	1,3,5	1,3,3,3	1,1,5,5	Berbeda
Ainatul M., M.Kom	1,2,3,4,5	1,1,2,5	1,1,3,5	Berbeda
Sugeng Ali M, M.Pd	1,2,3,4,5	1,5,5,5	2,3,3,4	Berbeda
M. Karim, M.Pd	1,2,3,4,5	1,3,3,5	1,1,1,4	Berbeda
M. Amin H., M.T	1,2,3	1,2,3	1,1,2	Berbeda
Syahiduz Zaman, M.Kom	1,2,3,4,5	1,1,1,4	1,1,3,5	Berbeda
Satria Mandala, M.Cs	1,5	1,1,1,5,5	1,1,5,5,5	Sama
Dr. Cahyo Cyrsdian	1,2,3	1,1,2,3	1,1,2,3	Sama
Fressy M.T	1,2,3	1,1,2,3	1,1,2,3	Sama
Linda M.T	2,3,4,5	2,3,4,5	2,4,4,5	Berbeda
Mukhlis Fakhruddin M.Si	2,3,5	2,2,3,5	2,3,5,5	Sama
Ririen K., M.Kom	2,3	2,2,3,3	2,2,3,3	Sama
Totok Chamidy, M.Kom	2,3,4	2,2,3,4	2,3,3,3	Berbeda
Syauqi, M.Kom	2,3,4	2,4,4,4	2,3,3,4	Berbeda
Fatchurrohman, M.Kom	2,3,4	2,3,4,4	2,3,3,4	Sama
Irwan, M.Kom	2,3,4	2,3,4,4	2,2,3,4	Sama
Roro Indah M., M.Kom	2,3,4	2,2,3,3,3,3,4	2,2,2,3,3,4,4	Sama
Yunifa, M.T	2,3,4	2,2,3,4	2,3,4,4	Sama
Hani, M.T	2,3,4	2,3,3,4	2,3,4,4	Sama
M. Ainul Yaqin, M.Kom	2,3,4	2,2,3,4	2,3,3,4	Sama
Zainal Abidin, M.Kom	2,3,4	2,3,4,4	2,2,4,4	Berbeda
Bayu Adhi N., M.Kom	3,5	3,5	5,5	Berbeda
Suhartono	4,5	4,5,5	4,4,5	Sama
Dra. Siti Anijat	4,5	4,4,5,5	4,5,5,5	Sama
Ir. Jasmani	5	5	5	Sama
Dr Ali Mahmudi	5	5	5	Sama

Ket : 1 = senin, 2 = selasa, 3 = rabu, 4 = kamis, 5 = jumat

Dari Tabel 4.8 menunjukkan bahwa adanya perbedaan hari pada jadwal yang tersusun antara versi 1.0 dan *multithread* dengan rincian data yang berbeda sebanyak 13 data, sedangkan data yang sama sebanyak 18 data. Data pada kolom preferensi mengajar merupakan konstrain hari dosen tersebut mengajar. Nilai 1 berarti hari Senin, dan berlaku seterusnya sampai nilai 5 yaitu hari Jum'at. Pada kasus seperti nama dosen Fahrul Kurniawan, M.MT memiliki preferensi hari 1,2,3. Jadwal yang terbentuk di versi 1.0 menunjukkan hari 1,1,2,2 sedangkan pada aplikasi *multithread* hari yang terbentuk 1,1,2,3. Kedua data tersebut tidak melanggar konstrain preferensi dosen tetapi belum bisa digunakan sebagai dasar untuk efisiensi karena terdapat perbedaan hasil. Dimana pada aplikasi versi 1.0 hari yang digunakan hanya 1 dan 2 sedangkan aplikasi *multithread* 1,2, dan 3. Perbedaan ini dikarenakan data versi 1.0 dikerjakan secara sekuensial sehingga slot langsung mengisi hari yang kosong. Sedangkan aplikasi *multithread* data yang selesai terlebih dahulu akan dimasukkan ke slot jadwal, sehingga untuk data hari pada contoh versi 1.0 adalah 1,1,2,2 berbeda dengan *multithread* 1,1,2,3 yang disebabkan karena untuk kelas terakhir pada hari ke 2 sudah digunakan oleh slot yang lain sehingga harus pindah ke hari selanjutnya.

**Tabel 4.9** Perbandingan Konstrain Ruang

Hari	Ruang B.306		Ruang B.307		Ruang B.314		Ruang B.315		Ruang B.316		Ruang B.317		Ruang B.318	
	1.0	1.1	1.0	1.1	1.0	1.1	1.0	1.1	1.0	1.1	1.0	1.1	1.0	1.1
Senin	7	5	4	5	4	4	5	4	5	5	3	6	1	1
Selasa	5	5	5	5	5	4	5	4	4	5	3	6	2	0
Rabu	6	6	4	4	4	5	5	4	4	4	3	4	2	0
Kamis	6	5	4	4	4	4	5	4	4	4	3	3	0	0
Jumat	4	4	4	3	4	4	3	4	3	3	1	3	0	0

Ket: 1.0 = aplikasi versi 1.0, 1.1 = aplikasi *multithread*

Data dari Tabel 4.9 merupakan hasil perbandingan banyaknya ruang yang digunakan pada aplikasi versi 1.0 dan aplikasi *multithread*. Kolom yang bernilai 1.0 merupakan ruang yang digunakan pada aplikasi versi 1.0 sedangkan kolom 1.1 merupakan data yang didapat dari aplikasi *multithread*. Jumlah ruangan yang digunakan pada aplikasi ini sebanyak 7 ruang yaitu B.306, B.307, B.314, B.315, B.316, B.317, dan B.318. Sebagai contoh, jumlah total ruang B.306 selama hari Senin sampai Jumat yang digunakan pada aplikasi 1.0 sebanyak 28 sedangkan pada aplikasi *multithread* sebanyak 25. Hal ini termasuk perbedaan hasil antara aplikasi versi 1.0 dengan aplikasi *multithread* pada konstrain ruang. Sehingga belum bisa digunakan sebagai dasar efisiensi karena perbedaan hasil.

Hari	Jam	Kode MK	Nama MK	SKS
Senin	06.30 - 09.00	0765314	SISTEM INFORMASI GEOGRAFIS	3
Senin	09.00 - 10.40	0765402	ISU PROFESIONAL & PROYEK G...	2
Senin	10.40 - 11.30	0765106	ILMU BUDAYA DASAR	1
Senin	12.20 - 14.00	0765402	ISU PROFESIONAL & PROYEK G...	2
Senin	14.00 - 14.50	0765106	ILMU BUDAYA DASAR	1
Senin	14.50 - 15.40	0765106	ILMU BUDAYA DASAR	1
Senin	15.40 - 16.30	0765105	ILMU SOSIAL DASAR	1
Senin	06.30 - 09.00	0765103	BAHASA INGGRIS I	3
Senin	09.00 - 10.40	0765108	STUDI AL-QUR AN	2
Senin	12.20 - 14.00	0765108	STUDI AL-QUR AN	2
Senin	14.00 - 15.40	0765113	TEOLOGI ISLAM	2
Senin	06.30 - 09.00	0765214	DESAIN BASIS DATA	3
Senin	09.00 - 11.30	0765214	DESAIN BASIS DATA	3
Senin	12.20 - 14.00	0765109	STUDI AL-HADITS	2
Senin	14.00 - 15.40	0765110	STUDI FIQH	2
Senin	06.30 - 08.10	0765110	STUDI FIQH	2
Senin	08.10 - 09.00	0765105	ILMU SOSIAL DASAR	1

**Gambar 4.9** Hasil Jam Perkuliahan versi 1.0

Hari	Jam	Kode MK	Nama MK	SKS
Senin	06.30 - 09.00	0765103	BAHASA INGGRIS I	3
Senin	09.00 - 10.40	0765402	ISU PROFESIONAL & PROYEK G...	2
Senin	10.40 - 11.30	0765106	ILMU BUDAYA DASAR	1
Senin	12.20 - 14.00	0765402	ISU PROFESIONAL & PROYEK G...	2
Senin	14.00 - 16.30	0765314	SISTEM INFORMASI GEOGRAFIS	3
Senin	06.30 - 09.00	0765305	PEMROGRAMAN MULTIMEDIA	3
Senin	09.00 - 11.30	0765314	SISTEM INFORMASI GEOGRAFIS	3
Senin	12.20 - 14.00	0765310	INTERAKSI MANUSIA & KOMPUT...	2
Senin	14.00 - 15.40	0765310	INTERAKSI MANUSIA & KOMPUT...	2
Senin	15.40 - 16.30	0765105	ILMU SOSIAL DASAR	1
Senin	06.30 - 09.00	0765314	SISTEM INFORMASI GEOGRAFIS	3
Senin	09.00 - 11.30	0765305	PEMROGRAMAN MULTIMEDIA	3
Senin	12.20 - 14.50	0765303	PENGOLAHAN CITRA	3
Senin	14.50 - 16.30	0765215	SITEM OPERASI	2
Senin	06.30 - 09.00	0765303	PENGOLAHAN CITRA	3
Senin	09.00 - 11.30	0765214	DESAIN BASIS DATA	3
Senin	12.20 - 14.50	0765214	DESAIN BASIS DATA	3

**Gambar 4.10** Hasil Jam Perkuliahan *Multithread*

Gambar 4.9 merupakan hasil keterangan jam perkuliahan pada aplikasi versi 1.0 dan Gambar 4.10 merupakan hasil dari aplikasi *multithread*. dari kedua data tersebut memiliki hasil yang sama pada konstrain waktu salat, yaitu untuk jam salat duhur. Perkuliahan akan berhenti pada jam 11.30 dan akan dimulai kembali jam 12.20. Sama halnya dengan konstrain salat Jumat, perkuliahan akan berhenti pada jam 10.40 dan akan dimulai kembali pada jam 13.00. Sehingga konstrain jam salat hasilnya sama antara aplikasi versi 1.0 dengan aplikasi *multithread*.

**Tabel 4.10** Hasil Perbandingan Konstrain Bentrok Dosen

Aplikasi	Data <i>input</i> plot	Data Hasil jadwal		Konstrain dosen
		Terjadwal	Tidak terjadwal	
Versi 1.0	131 <i>record</i>	131	0	0
<i>multithread</i>	131 <i>record</i>	131	0	0

Pada Tabel 4.10 ini menunjukkan bahwa konstrain bentrok dosen bernilai 0 untuk aplikasi versi 1.0 dan aplikasi *multithread*. Hal ini diperoleh karena jumlah data plot yang terjadwal sama dengan data *input* plot yaitu sebesar 131 *record* data. Hasil yang sama pada konstrain bentrok dosen bisa dikaitkan dengan kecepatan efisiensi.

#### 4.5.3. Hasil dengan Perbedaan Jumlah Ruang

Hasil uji coba yang dilakukan dengan perbedaan ruangan kelas pada setiap jadwal yang di *generate*. Ada 3 pembandingan data ruangan yaitu dengan 7 ruangan kelas, 6 ruangan kelas dan 5 ruangan kelas. Uji coba ini dilakukan pada

aplikasi GPU dan aplikasi *multithread* dengan menggunakan perangkat laptop lenovo G480.

### 1. Aplikasi GPU

Pada subbab 4.3.1 uji coba dilakukan dengan 3 tes ruang yang berbeda, dan menghasilkan grafik sebagai berikut :



**Gambar 4.11 Data Jadwal Matakuliah GPU dan Versi 1.0**

Gambar 4.9 menjelaskan bahwa jumlah matakuliah yang terjadwal pada aplikasi GPU sama dengan jumlah matakuliah yang terjadwal pada aplikasi versi 1.0. Susunan urutan matakuliah yang terjadwal pada kedua uji coba juga sama. Matakuliah yang terjadwal pada ujicoba 7 ruang sebanyak 131 jadwal, 6 ruang 131 jadwal dan 5 ruang 111 jadwal.

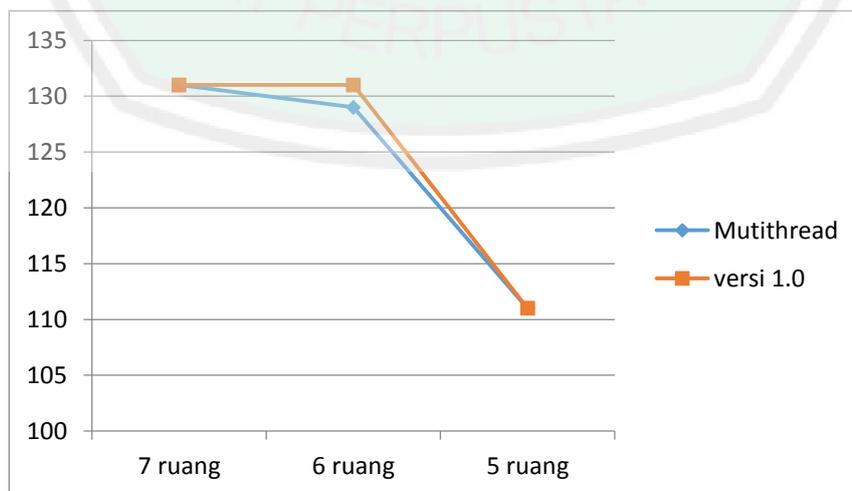


**Gambar 4.12 Waktu Komputasi GPU dan Versi 1.0**

Gambar 4.10 menjelaskan perbedaan waktu eksekusi pada aplikasi GPU dengan aplikasi versi 1.0. Aplikasi dengan GPU waktu rata-rata setiap generate jadwal 3560 detik, sedangkan aplikasi versi 1.0 membutuhkan waktu eksekusi rata-rata 780 detik.

## 2. Aplikasi Multithread

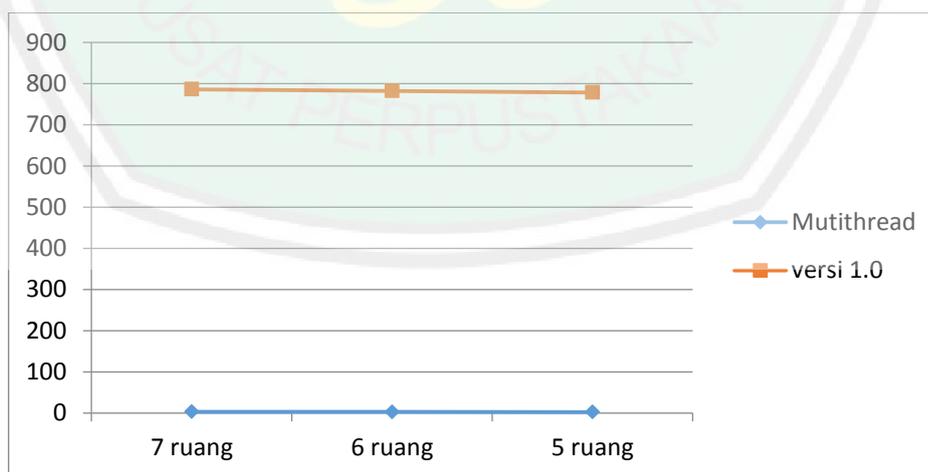
Pada subbab 4.3.2 uji coba dilakukan dengan 3 tes ruang yang berbeda, dan menghasilkan grafik sebagai berikut :



**Gambar 4.13 Data Jadwal Matakuliah Multithread dan Versi 1.0**

Gambar 4.11 menjelaskan bahwa jumlah matakuliah yang terjadwal pada *multithread* rata-ratanya menurun ketika jumlah ruang berkurang, dikarenakan jadwal yang tersusun bersifat random jadi ruangan yang terpakai tidak bisa maksimal. Susunan urutan matakuliah yang terjadwal pada kedua uji coba tersebut berbeda. Aplikasi versi 1.0 menghasilkan data jadwal sama setiap generate. Hasil generate aplikasi *multithread* lebih acak.

Uji coba dengan 7 ruang pada versi 1.0 menjadwalkan data 131, sama halnya dengan *multithread* mampu menjadwalkan data sebanyak 131. Uji coba dengan 6 ruang berbeda hasil antara versi 1.0 dan *multithread*, pada versi 1.0 matakuliah yang terjadwal 131 data sedangkan *multithread* dari 5 kali percobaan rata-rata 129 data yang terjadwal. Uji coba dengan 5 ruang rata-rata data yang terjadwal sama sebanyak 111 data, namun data dari *multithread* lebih bervariasi dari 5 kali uji coba dengan *range* data sebanyak 110 sampai 113.



**Gambar 4.14 Waktu Komputasi Multithread dan Versi 1.0**

Gambar 4.10 menjelaskan perbedaan waktu eksekusi pada aplikasi *multithread* dengan aplikasi versi 1.0. Aplikasi dengan *multithread* waktu

rata-rata setiap generate jadwal 3 detik, sedangkan aplikasi versi 1.0 membutuhkan waktu eksekusi rata-rata 780 detik. Dari hasil tersebut aplikasi *multithread* lebih efisien daripada aplikasi GPU.

#### 4.5.4. Hasil *Speedup* Aplikasi paralel dengan sekuensial

Hasil perhitungan *speedup* didapat dengan cara membagi hasil proses waktu komputasi sekuensial dengan hasil waktu komputasi dengan paralel. Pada ujicoba kali ini Tabel 4.8 menerangkan perbedaan *speedup* antara versi 1.0 dengan *multithread*. Sedangkan Tabel 4.9 menerangkan perbedaan *speedup* antara aplikasi versi 1.0 *arraylist* dengan *multithread*.

**Tabel 4.11 Hasil Speedup Aplikasi Versi 1.0 dengan Multithread**

Jumlah data matakuliah	Waktu eksekusi (detik)		Speedup
	Versi 1.0	Multithread	
0	24.6824	3.3088	7.459623
2140	273.7298	3.4094	80.2868
4280	615.1576	3.4374	178.9601
8560	1145.072	3.2662	350.5823

**Tabel 4.12 Hasil Speedup Aplikasi Versi 1.0 Arraylist dengan Multithread**

Jumlah data matakuliah	Waktu eksekusi (detik)		Speedup
	Versi arraylist	Multithread	
0	13.1678	3.3088	3.97963
2140	11.1186	3.4094	3.26116
4280	11.4718	3.4374	3.337348
8560	92.037	3.2662	28.17862

Dari Tabel 4.8 diperoleh hasil *speedup* untuk multithread dengan versi 1.0 sangat besar setiap kali penambahan data jadwal. Waktu komputasi pada versi 1.0 juga sangat lama dibandingkan dengan *multithread*.

Dari Tabel 4.9 diperoleh hasil *speedup* untuk multithread dengan versi 1.0 *arraylist* tidak terlalu besar setiap kali penambahan data jadwal. Waktu komputasi pada versi 1.0 *arraylist* 3 kali lebih lama dibandingkan dengan *multithread*.

Dari kedua data diatas diperoleh kesimpulan bahwa dengan *multithread* bisa meningkatkan kecepatan komputasi sebuah aplikasi. *Speedup* hasil ujicoba pada aplikasi ini menunjukkan bahwa komputasi paralel lebih cepat ketimbang dengan komputasi sekuensial.

#### 4.6.Kajian Al-Qur'an dan Hadis

Penjadwalan merupakan pembagian waktu terhadap rancangan urutan pekerjaan yang akan dilakukan. Penjadwalan sangat diperlukan untuk mempermudah pekerjaan karena dapat membagi pekerjaan sesuai dengan waktu yang ditentukan sehingga dapat mencapai hasil yang optimal.

Allah S.W.T memberikan kita kehidupan dengan adanya batas waktu agar kita dapat menggunakan waktu di dunia dengan sebaik-baiknya. Sehingga dengan adanya penyusunan jadwal dapat dikatakan sebagai pengatur waktu kegiatan kita agar kita tidak menyia-nyiakan waktu dengan kegiatan yang tidak berguna. Seperti firman Allah S.W.T berikut :

وَلِكُلِّ أُمَّةٍ أَجَلٌ صَلَّى فَإِذَا جَاءَ أَجْلُهُمْ لَا يَسْتَأْذِنُونَ سَاعَةً صَلَّى

وَلَا يَسْتَفْتِحُونَ (٣٤)

Artinya : *Tiap-tiap umat mempunyai batas waktu; maka apabila telah datang waktunya mereka tidak dapat mengundurkannya barang sesaat pun dan tidak dapat (pula) memajukannya (QS. Al-A'raf :34).*

Pada ayat diatas dijelaskan bahwasanya setiap manusia yang hidup di muka bumi ini memiliki batas waktu. Adapun waktu yang diberikan oleh Allah S.W.T tidak dapat dimajukan ataupun dimundurkan oleh makhlukNya. Inilah salah satu contoh nyata dalam kehidupan dari penerapan Q.S Al-A'raf : 34, yakni dengan pembuatan jadwal untuk mengatur kegiatan kita.

Konsep dari komputasi paralel adalah mengerjakan program secara bersama. Sehingga ketika satu program selesai maka langsung disusul dengan program yang lainnya. Firman Allah SWT di Q.S Al-Sharh ayat 7 yaitu :

فَإِذَا فَرَغْتَ فَانصَبْ ﴿٧﴾

Artinya : *Maka apabila kamu telah selesai (dari satu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain.*

Pada Q.S Al-Sharh ayat 7 diterangkan bahwa, ketika manusia telah selesai dengan satu urusan maka dianjurkan untuk segera melakukan urusan yang lain. Perintah ini mengajarkan bahwa waktu untuk satu urusan harus digunakan secara tepat supaya bisa bermanfaat untuk urusan selanjutnya. Dalam praktek komputasi paralel ketika satu proses selesai maka langsung disusul oleh proses lainnya karena berjalan secara bersama dalam beberapa *core*.

Begitu pentingnya waktu dalam agama islam, nabi Muhammad SAW mengingatkan umatnya bahwa untuk selalu mempersiapkan lima hal sebelum

datang yang lima. Hadist tersebut sering disebut dengan hadist *ightanim khomsan qobla khomsin*. Diriwayatkan oleh Baihaki dari Ibnu Abbas yang berisi :

إِغْتَنِمَ خَمْسًا قَبْلَ خَمْسٍ : شَبَابَكَ قَبْلَ هَرَمِكَ وَ صِحَّتَكَ قَبْلَ سَقَمِكَ وَ غِنَاكَ قَبْلَ فَقْرِكَ وَ فَرَاعَكَ قَبْلَ شَغْلِكَ وَ حَيَاتَكَ قَبْلَ مَوْتِكَ

Artinya : *Manfaatkanlah yang lima sebelum datang lima perkara yang lain : mudamu sebelum tuamu, sehatmu sebelum sakitmu, kayamu sebelum miskinmu, waktu senggangmu sebelum kesibukanmu dan hidupmu sebelum matimu* (H.R Baihaki dari Ibnu Abbas, Shu'ab al-iman 9575).

Efisiensi berkaitan erat dengan mengoptimalkan waktu yang tersedia. Jika sebelumnya pada program penjadwalan versi 1.0 memerlukan waktu yang lama, maka diperlukan efisiensi untuk meminimalisir waktu dari program penjadwalan versi 1.0. Dengan meminimalisir waktu untuk melakukan penjadwalan, kita dapat menggunakan sisa waktu untuk kebaikan yang lainnya. Karena dengan membuang-buang waktu dengan kegiatan tidak berguna, kita akan menjadi orang yang merugi. Seperti dalam firman Allah S.W.T :

وَالْعَصْرِ (١) إِنَّ الْإِنْسَانَ لَفِي خُسْرٍ (٢) إِلَّا الَّذِينَ آمَنُوا وَعَمِلُوا الصَّالِحَاتِ وَتَوَّصُوا بِالصَّبْرِ (٣)

Artinya :

*“Demi masa. Sesungguhnya manusia itu benar-benar dalam kerugian, kecuali orang-orang yang beriman dan mengerjakan amal saleh dan nasehat menasehati supaya mentaati kebenaran dan nasehat menasehati supaya menepati kesabaran.”* (QS. Al ‘Ashr: 1-3).

Isi dari surat Al-'Ashr supaya kita tidak membuang-buang waktu dan tidak menjadi orang yang merugi. Ditambah dengan pendapat Imam Syafi'i dalam kitab Al Jawaabul Kaafi karya Ibnu Qoyim bahwa waktu seperti pedang, jika kita tidak pandai menggunakan pedang maka pedang tersebut akan menebas kita sendiri. Demikian juga waktu yang telah diberikan oleh Allah SWT, jika kita tidak mampu memanfaatkan waktu maka kita akan menjadi merugi. Sehingga adanya efisiensi waktu pada program penjadwalan ini dapat dikatakan mengimplementasikan Al Qur'an surat Al 'Ashr.



## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan hasil dan pembahasan yang diperoleh, dapat disimpulkan bahwa dengan mengimplementasikan GPU, proses komputasi aplikasi versi 1.0 menjadi lebih lama sekitar 3500 detik, sedangkan untuk versi 1.0 proses komputasi berjalan dengan waktu sekitar 700 detik. Sehingga aplikasi dengan GPU belum bisa dikatakan efisien karena waktu untuk komputasi lebih lama dibandingkan versi 1.0. Proses komputasi dengan *multithread* pada program aplikasi penjadwalan dapat meningkatkan kecepatan waktu komputasi sekitar 3 detik untuk satu kali proses. Ada beberapa konstrain pada aplikasi *multithread* yang hasilnya berbeda dengan versi 1.0 seperti, konstrain preferensi dosen dan ruang, sehingga untuk dua konstrain tersebut belum bisa dikatakan efisien. Untuk konstrain salat dan bentrok dosen hasilnya sama dengan versi 1.0 sehingga efisiensi kecepatan antara versi 1.0 dan *multithread* bisa dilakukan. Karena efisiensi waktu merupakan salah satu ciri dari *green software*, dapat dikatakan bahwa implementasi *multithread* pada program aplikasi penjadwalan merupakan langkah pendekatan untuk menuju *green software*.

Implementasi penggunaan komputasi paralel lebih cepat dibandingkan dengan komputasi sekuensial. Hal ini dapat dibuktikan dengan perbedaan kecepatan proses sekuensial dengan *arraylist* sekitar 11 detik dan proses paralel

dengan *multithread* 3 detik. Sehingga komputasi paralel dengan *multithread* 3 kali lebih cepat dibandingkan aplikasi sekuensial dengan *arraylist*.

## 5.2. Saran

Berdasarkan penelitian yang telah dilakukan, pada bagian aplikasi penjadwalan, masih terdapat banyak hal yang perlu dikembangkan. Berikut merupakan saran untuk penelitian selanjutnya :

- Menambahkan pembagian *workitem* yang sesuai pada masing-masing *core* GPU untuk mempercepat proses komputasi GPU.
- Menambahkan form untuk memasukkan data jadwal yang terbaru.

## DAFTAR PUSTAKA

- Agarwal. (2012). *Sustainable Approaches and Good Practices in Green Software Engineering*. IJRRCs.
- Almutairi, A. H., & Alruwaili, A. H. (2012). *Improving of Quicksort Algorithm Performance by Sequential Thread or Parallel Algorithms*. Global Journal of Computer Science and Technology Hardware & Computation .
- Amalia, K. R. (2016). *Implementasi Metode Alpha Trimmed Mean Filter Menggunakan OpenCL untuk Menghapus Noise pada Citra Berwarna*. UIN Maulana Malik Ibrahim.
- Erdelyi, K. (2013). *Special factors of development of green software supporting eco sustainability*. IEEE .
- Hwu, & Krik. (2010). *Programming Massively Parallel Processors A Hands-on Approach*.
- Kelvin Onggrono, dkk. (2017). *ANALISIS PENGGUNAAN PARALLEL PROCESSING MULTITHREADING*. InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan) Vol 2, No 1.
- Kern, E., & dkk. (2013). *Green Software and Green Software Engineering Definitions, Measurements, and Quality Aspects*. ICT4S 2013: Proceedings of the First International Conference on Information and Communication Technologies for Sustainability.
- Krishnan, Dwivedi, & Bhagwat. (2012). *Green Computing using Graphical Processing Units*. Bangalore.

- Larsson, P. (2011). *Energy-Efficient Software Guidelines*. Intel Software Solutions Group, Tech.
- Mohammad Zarkasi, d. (2013). *Implementasi Komputasi Paralel Untuk Enkripsi Citra Berbasis AES Menggunakan JPPF*. JURNAL TEKNIK POMITS Vol. 2, No. 1,.
- Murgensen, S. (2008). *Harnessing Green IT: Principles and Practices*. IEEE Computer Society.
- nvidia. (n.d.). Dikunjungi 20 Januari 2017, dari /www.nvidia.com:  
<http://www.nvidia.com/object/what-is-gpu-computing.html>
- Pratama, I. P. (2015). *Green Computing*. Bandung: Informatika .
- Steigerwald, B. ., (2012). *Developing Green Software*. SOFTWARE & SERVICES GROUP, INTEL CORPORATION, FOLSOM, CA, USA.
- Taina, J. (2011). *Good, Bad, and Beautiful Software - In Search of Green Software Quality Factors*. Green ICT.
- tutorialspoint. (n.d.). Dikunjungi 20 Desember 2017, dari  
[https://www.tutorialspoint.com/operating\\_system/os\\_multi\\_threading.htm](https://www.tutorialspoint.com/operating_system/os_multi_threading.htm)
- wikibooks. (n.d.). Dikunjungi 20 Desember 2017, dari  
[https://en.wikibooks.org/wiki/Introduction\\_to\\_Software\\_Engineering/Process/Life\\_Cycle](https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Process/Life_Cycle)
- Yuniastari, N. L. (2015). *Pengukuran Tingkat Efektivitas Dan Efisiensi Sistem Eresearch STIKOM Bali*. Konferensi Nasional Sistem & Informatika 2015.

## LAMPIRAN

### Hasil Jadwal Kuliah versi 1.0 dan GPU

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Senin	06.30 - 09.00	0765314	SISTEM INFORMASI GEOGRAFIS	3	D	65018	Fahrul Kurniawan M.MT	B.306
Senin	09.00 - 10.40	0765402	ISU PROFESIONAL & PROYEK G...	2	A	65021	Ali Wafa M.Kom	B.306
Senin	10.40 - 11.30	0765106	ILMU BUDAYA DASAR	1	A	13014	NI MATUZ ZUHROH M.Si	B.306
Senin	12.20 - 14.00	0765402	ISU PROFESIONAL & PROYEK G...	2	C	65021	Ali Wafa M.Kom	B.306
Senin	14.00 - 14.50	0765106	ILMU BUDAYA DASAR	1	B	13014	NI MATUZ ZUHROH M.Si	B.306
Senin	14.50 - 15.40	0765106	ILMU BUDAYA DASAR	1	D	13014	NI MATUZ ZUHROH M.Si	B.306
Senin	15.40 - 16.30	0765105	ILMU SOSIAL DASAR	1	A	13014	NI MATUZ ZUHROH M.Si	B.306
Senin	06.30 - 09.00	0765103	BAHASA INGGRIS I	3	A	65111	Robert Aprillianto M.Pd	B.307
Senin	09.00 - 10.40	0765108	STUDI AL-QUR AN	2	A	65017	Imamudin M.A	B.307
Senin	12.20 - 14.00	0765108	STUDI AL-QUR AN	2	C	65017	Imamudin M.A	B.307
Senin	14.00 - 15.40	0765113	TEOLOGI ISLAM	2	C	60002	AHMAD BARIZI, M.A	B.307
Senin	06.30 - 09.00	0765214	DESAIN BASIS DATA	3	A	65130	Ainatul M M.Kom	B.314
Senin	09.00 - 11.30	0765214	DESAIN BASIS DATA	3	C	65130	Ainatul M M.Kom	B.314
Senin	12.20 - 14.00	0765109	STUDI AL-HADITS	2	C	65199	Sugeng Ali Mansyur M.Pd	B.314
Senin	14.00 - 15.40	0765110	STUDI FIQH	2	A	65151	Muhammad Karim M.Pd	B.314
Senin	06.30 - 08.10	0765110	STUDI FIQH	2	C	65151	Muhammad Karim M.Pd	B.315
Senin	08.10 - 09.00	0765105	ILMU SOSIAL DASAR	1	D	13014	NI MATUZ ZUHROH M.Si	B.315
Senin	09.00 - 10.40	0765208	PENGANTAR TEKNOLOGI INFOR...	2	A	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.315
Senin	12.20 - 14.00	0765310	INTERAKSI MANUSIA & KOMPUT...	2	A	65005	SYAHIDUZ ZAMAN M.Kom	B.315
Senin	14.00 - 15.40	0765310	INTERAKSI MANUSIA & KOMPUT...	2	B	65005	SYAHIDUZ ZAMAN M.Kom	B.315
Senin	06.30 - 08.10	0765310	INTERAKSI MANUSIA & KOMPUT...	2	D	65005	SYAHIDUZ ZAMAN M.Kom	B.316
Senin	08.10 - 09.50	0765215	SITEM OPERASI	2	B	65009	null SATRIA MANDALA, MCS	B.316
Senin	09.50 - 11.30	0765215	SITEM OPERASI	2	C	65009	null SATRIA MANDALA, MCS	B.316
Senin	12.20 - 14.00	0765215	SITEM OPERASI	2	D	65009	null SATRIA MANDALA, MCS	B.316
Senin	14.00 - 16.30	0765303	PENGOLAHAN CITRA	3	A	65015	Dr Cahyo Crysdian	B.316
Senin	06.30 - 09.00	0765303	PENGOLAHAN CITRA	3	B	65015	Dr Cahyo Crysdian	B.317
Senin	09.00 - 11.30	0765305	PEMROGRAMAN MULTIMEDIA	3	A	65019	Fressy MT	B.317
Senin	12.20 - 14.50	0765305	PEMROGRAMAN MULTIMEDIA	3	B	65019	Fressy MT	B.317
Senin	09.00 - 11.30	0765314	SISTEM INFORMASI GEOGRAFIS	3	A	65018	Fahrul Kurniawan M.MT	B.318
Selasa	06.30 - 09.00	0765314	SISTEM INFORMASI GEOGRAFIS	3	E	65018	Fahrul Kurniawan M.MT	B.306
Selasa	09.00 - 11.30	0765315	SISTEM INFORMASI CERDAS	3	D	65016	Linda M.T	B.306
Selasa	12.20 - 14.00	0765402	ISU PROFESIONAL & PROYEK G...	2	D	65021	Ali Wafa M.Kom	B.306
Selasa	14.00 - 14.50	0765106	ILMU BUDAYA DASAR	1	C	13014	NI MATUZ ZUHROH M.Si	B.306
Selasa	14.50 - 16.30	0765112	SEJARAH PERADABAN ISLAM	2	B	33333	Mukhlis Fakhruddin M.Si	B.306

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Selasa	06.30 - 08.10	0765112	SEJARAH PERADABAN ISLAM	2	D	33333	Mukhlis Fakhruddin M.Si	B.307
Selasa	08.10 - 10.40	0765203	ALJABAR LINIER DAN MATRIKS	3	A	65003	RIRIEN KUSUMAWATI, S.Si, M.Kom	B.307
Selasa	10.40 - 11.30	0765105	ILMU SOSIAL DASAR	1	B	13014	NI MATUZ ZUHROH, M.Si	B.307
Selasa	12.20 - 14.50	0765203	ALJABAR LINIER DAN MATRIKS	3	B	65003	RIRIEN KUSUMAWATI, S.Si, M.Kom	B.307
Selasa	14.50 - 16.30	0765216	RANGKAIAN DIGITAL	2	A	65011	TOTOK CHAMIDI, M.Kom	B.307
Selasa	06.30 - 09.00	0765210	STRUKTUR DATA	3	C	65101	Syauqi, M.Kom	B.314
Selasa	09.00 - 11.30	0765214	DESAIN BASIS DATA	3	D	65130	Ainatul M.M.Kom	B.314
Selasa	12.20 - 14.00	0765216	RANGKAIAN DIGITAL	2	D	65011	TOTOK CHAMIDI, M.Kom	B.314
Selasa	14.00 - 15.40	0765309	REKAYASA PERANGKAT LUNAK	2	B	65002	FATCHURROCHMAN, M.Kom	B.314
Selasa	15.40 - 16.30	0765105	ILMU SOSIAL DASAR	1	C	13014	NI MATUZ ZUHROH, M.Si	B.314
Selasa	06.30 - 08.10	0765309	REKAYASA PERANGKAT LUNAK	2	E	65002	FATCHURROCHMAN, M.Kom	B.315
Selasa	08.10 - 09.50	0765206	STATISTIKA	2	B	65113	Irwan M.Kom	B.315
Selasa	09.50 - 11.30	0765208	PENGANTAR TEKNOLOGI INFOR...	2	B	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.315
Selasa	12.20 - 14.00	0765209	LOGIKA DAN ALGORITMA	2	A	65007	RORO INDA MELANI, S.Kom	B.315
Selasa	14.00 - 15.40	0765209	LOGIKA DAN ALGORITMA	2	C	65007	RORO INDA MELANI, S.Kom	B.315
Selasa	06.30 - 09.00	0765218	JARINGAN KOMPUTER	3	A	65020	Yunifa MT	B.316
Selasa	09.00 - 11.30	0765218	JARINGAN KOMPUTER	3	D	65020	Yunifa MT	B.316
Selasa	12.20 - 14.50	0765301	KECERDASAN BUATAN	3	C	65102	Hani, M.T	B.316
Selasa	14.50 - 16.30	0765404	MANAJ. PROYEK PERANGKAT LU...	2	B	65010	MUHAMMAD AINUL YACIN, M.Kom	B.316
Selasa	06.30 - 09.00	0765303	PENGOLAHAN CITRA	3	C	65015	Dr Cahyo Crysdiar	B.317
Selasa	09.00 - 10.40	0765404	MANAJ. PROYEK PERANGKAT LU...	2	E	65010	MUHAMMAD AINUL YACIN, M.Kom	B.317
Selasa	12.20 - 14.50	0765305	PEMROGRAMAN MULTIMEDIA	3	C	65019	Fressy MT	B.317
Selasa	06.30 - 09.00	0765313	KOMPUTASI BERGERAK	3	C	65008	ZAINAL ABIDIN, S.Kom	B.318
Selasa	09.00 - 11.30	0765314	SISTEM INFORMASI GEOGRAFIS	3	B	65018	Fahrul Kurniawan M.T	B.318
Selasa	06.30 - 09.00	0765315	SISTEM INFORMASI CERDAS	3	A	65016	Linda M.T	B.306
Rabu	09.00 - 10.40	0765402	ISU PROFESIONAL & PROYEK G...	2	B	65021	Ali Wafa M.Kom	B.306
Rabu	10.40 - 11.30	0865122	TARBIYAH ULUL ALBAB	1	A	65007	RORO INDA MELANI, S.Kom	B.306
Rabu	12.20 - 14.00	0765402	ISU PROFESIONAL & PROYEK G...	2	E	65021	Ali Wafa M.Kom	B.306
Rabu	14.00 - 15.40	0765112	SEJARAH PERADABAN ISLAM	2	A	33333	Mukhlis Fakhruddin M.Si	B.306
Rabu	15.40 - 16.30	0865122	TARBIYAH ULUL ALBAB	1	C	65007	RORO INDA MELANI, S.Kom	B.306
Rabu	06.30 - 09.00	0765103	BAHASA INGGRIS I	3	B	65111	Robert Aprillianto M.Pd	B.307
Rabu	09.00 - 10.40	0765113	TEOLOGI ISLAM	2	A	60002	AHMAD BARIZI, M.A	B.307
Rabu	12.20 - 14.00	0765113	TEOLOGI ISLAM	2	B	60002	AHMAD BARIZI, M.A	B.307
Rabu	14.00 - 15.40	0765113	TEOLOGI ISLAM	2	D	60002	AHMAD BARIZI, M.A	B.307

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Rabu	06.30 - 09.00	0765203	ALJABAR LINIER DAN MATRIKS	3	C	65003	RIRIEN KUSUMAWATI, S.Si, M.Kom	B.314
Rabu	09.00 - 11.30	0765203	ALJABAR LINIER DAN MATRIKS	3	D	65003	RIRIEN KUSUMAWATI, S.Si, M.Kom	B.314
Rabu	12.20 - 14.00	0765216	RANGKAIAN DIGITAL	2	E	65011	TOTOK CHAMIDI, M.Kom	B.314
Rabu	14.00 - 15.40	0765309	REKAYASA PERANGKAT LUNAK	2	C	65002	FATCHURROCHMAN, M.Kom	B.314
Rabu	06.30 - 08.10	0765110	STUDI FIQH	2	D	65151	Muhammad Karim, M.Pd	B.315
Rabu	08.10 - 09.50	0765206	STATISTIKA	2	D	65113	Irwan M.Kom	B.315
Rabu	09.50 - 11.30	0765208	PENGANTAR TEKNOLOGI INFOR...	2	C	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.315
Rabu	12.20 - 14.00	0765209	LOGIKA DAN ALGORITMA	2	B	65007	RORO INDA MELANI, S.Kom	B.315
Rabu	14.00 - 15.40	0765209	LOGIKA DAN ALGORITMA	2	D	65007	RORO INDA MELANI, S.Kom	B.315
Rabu	06.30 - 09.00	0765218	JARINGAN KOMPUTER	3	B	65020	Yunifa MT	B.316
Rabu	09.00 - 11.30	0765301	KECERDASAN BUATAN	3	A	65102	Hani, M.T	B.316
Rabu	12.20 - 14.50	0765301	KECERDASAN BUATAN	3	D	65102	Hani, M.T	B.316
Rabu	14.50 - 16.30	0765404	MANAJ. PROYEK PERANGKAT LU...	2	C	65010	MUHAMMAD AINUL YAQIN, M.Kom	B.316
Rabu	06.30 - 09.00	0765303	PENGOLAHAN CITRA	3	D	65015	Dr Cahyo Crysdiar	B.317
Rabu	09.00 - 11.30	0765306	POMROGRAMAN WEB	3	B	65121	null Bayu Adhi Nugroho M.Kom	B.317
Rabu	12.20 - 14.50	0765305	PEMROGRAMAN MULTIMEDIA	3	D	65019	Fressy MT	B.317
Rabu	06.30 - 09.00	0765313	KOMPUTASI BERGERAK	3	D	65008	ZAINAL ABIDIN, S.Kom	B.318
Rabu	09.00 - 11.30	0765314	SISTEM INFORMASI GEOGRAFIS	3	C	65018	Fahrul Kurniawan M.MT	B.318
Kamis	06.30 - 09.00	0765315	SISTEM INFORMASI CERDAS	3	B	65016	Linda M.T	B.306
Kamis	09.00 - 10.40	0765207	METEDOLOGI PENELITIAN	2	A	65005	SYAHIDUZ ZAMAN, M.Kom	B.306
Kamis	10.40 - 11.30	0865122	TARBIYAH ULUL ALBAB	1	B	65007	RORO INDA MELANI, S.Kom	B.306
Kamis	12.20 - 14.00	0865102	BAHASA INDONESIA	2	B	11036	Dra Siti Annijat Maimunah M.Pd	B.306
Kamis	14.00 - 15.40	0865102	BAHASA INDONESIA	2	D	11036	Dra Siti Annijat Maimunah M.Pd	B.306
Kamis	15.40 - 16.30	0865122	TARBIYAH ULUL ALBAB	1	D	65007	RORO INDA MELANI, S.Kom	B.306
Kamis	06.30 - 09.00	0765103	BAHASA INGGRIS I	3	C	65111	Robert Aprillianto M.Pd	B.307
Kamis	09.00 - 11.30	0765210	STRUKTUR DATA	3	A	65008	ZAINAL ABIDIN, S.Kom	B.307
Kamis	12.20 - 14.50	0765210	STRUKTUR DATA	3	B	65008	ZAINAL ABIDIN, S.Kom	B.307
Kamis	14.50 - 16.30	0765216	RANGKAIAN DIGITAL	2	B	65011	TOTOK CHAMIDI, M.Kom	B.307
Kamis	06.30 - 09.00	0765210	STRUKTUR DATA	3	D	65101	Syauci, M.Kom	B.314
Kamis	09.00 - 10.40	0765216	RANGKAIAN DIGITAL	2	C	65011	TOTOK CHAMIDI, M.Kom	B.314
Kamis	12.20 - 14.00	0765309	REKAYASA PERANGKAT LUNAK	2	A	65002	FATCHURROCHMAN, M.Kom	B.314
Kamis	14.00 - 15.40	0765309	REKAYASA PERANGKAT LUNAK	2	D	65002	FATCHURROCHMAN, M.Kom	B.314
Kamis	06.30 - 08.10	0765206	STATISTIKA	2	A	65113	Irwan M.Kom	B.315
Kamis	08.10 - 09.50	0765206	STATISTIKA	2	E	65113	Irwan M.Kom	B.315

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Kamis	09.50 - 11.30	0765404	MANAJ. PROYEK PERANGKAT LU...	2	A	65010	MUHAMMAD AINUL YAQIN, M.Kom	B.315
Kamis	12.20 - 14.50	0765201	MATEMATIKA DASAR	3	B	65001	SUHARTONO, M.Kom	B.315
Kamis	14.50 - 16.30	0765310	INTERAKSI MANUSIA & KOMPUT...	2	C	65005	SYAHIDUZ ZAMAN M.Kom	B.315
Kamis	06.30 - 09.00	0765218	JARINGAN KOMPUTER	3	C	65020	Yunifa MT	B.316
Kamis	09.00 - 11.30	0765301	KECERDASAN BUATAN	3	B	65102	Hani, M.T	B.316
Kamis	12.20 - 14.50	0765306	POMROGRAMAN WEB	3	C	65004	MUHAMMAD FAISAL M.T	B.316
Kamis	14.50 - 16.30	0765404	MANAJ. PROYEK PERANGKAT LU...	2	D	65010	MUHAMMAD AINUL YAQIN, M.Kom	B.316
Kamis	06.30 - 09.00	0765306	POMROGRAMAN WEB	3	D	65004	MUHAMMAD FAISAL M.T	B.317
Kamis	09.00 - 11.30	0765313	KOMPUTASI BERGERAK	3	A	65101	Syaugi, M.Kom	B.317
Kamis	12.20 - 14.50	0765313	KOMPUTASI BERGERAK	3	B	65101	Syaugi, M.Kom	B.317
Jumat	06.30 - 09.00	0765315	SISTEM INFORMASI CERDAS	3	C	65016	Linda M.T	B.306
Jumat	09.00 - 10.40	0865102	BAHASA INDONESIA	2	A	11036	Dra Siti Annijat Maimunah M.Pd	B.306
Jumat	13.10 - 14.50	0865102	BAHASA INDONESIA	2	C	11036	Dra Siti Annijat Maimunah M.Pd	B.306
Jumat	14.50 - 16.30	0765112	SEJARAH PERADABAN ISLAM	2	C	33333	Mukhlis Fakhruddin M.Si	B.306
Jumat	06.30 - 09.00	0765103	BAHASA INGGRIS I	3	D	65111	Robert Aprillianto M.Pd	B.307
Jumat	09.00 - 10.40	0765108	STUDI AL-QUR AN	2	B	65017	Imamudin M.A	B.307
Jumat	13.10 - 14.50	0765108	STUDI AL-QUR AN	2	D	65017	Imamudin M.A	B.307
Jumat	14.50 - 16.30	0765109	STUDI AL-HADITS	2	A	65199	Sugeng Ali Mansyur M.Pd	B.307
Jumat	06.30 - 09.00	0765214	DESAIN BASIS DATA	3	B	65130	Ainatul M M.Kom	B.314
Jumat	09.00 - 10.40	0765109	STUDI AL-HADITS	2	B	65199	Sugeng Ali Mansyur M.Pd	B.314
Jumat	13.10 - 14.50	0765109	STUDI AL-HADITS	2	D	65199	Sugeng Ali Mansyur M.Pd	B.314
Jumat	14.50 - 16.30	0765110	STUDI FIQH	2	B	65151	Muhammad Karim M.Pd	B.314
Jumat	06.30 - 08.10	0765215	SITEM OPERASI	2	A	65009	null SATRIA MANDALA, MCS	B.315
Jumat	08.10 - 10.40	0765201	MATEMATIKA DASAR	3	A	65001	SUHARTONO, M.Kom	B.315
Jumat	13.10 - 15.40	0765201	MATEMATIKA DASAR	3	C	65001	SUHARTONO, M.Kom	B.315
Jumat	06.30 - 09.00	0765201	MATEMATIKA DASAR	3	D	65112	Dr Ali Mahmudi	B.316
Jumat	09.00 - 10.40	0765208	PENGANTAR TEKNOLOGI INFOR...	2	D	65103	Ir. Jasmani M.Kom	B.316
Jumat	13.10 - 14.50	0765215	SITEM OPERASI	2	E	65009	null SATRIA MANDALA, MCS	B.316
Jumat	06.30 - 09.00	0765306	POMROGRAMAN WEB	3	A	65121	null Bayu Adhi Nugroho M.Kom	B.317

### Hasil Jadwal Kuliah dengan Multithread

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Senin	06.30 - 09.00	0765103	BAHASA INGGRIS I	3	A	65111	Robert Aprilianto M.Pd	B.306
Senin	09.00 - 10.40	0765402	ISU PROFESIONAL & PROYEK G...	2	E	65021	Ali Wafa M.Kom	B.306
Senin	10.40 - 11.30	0765106	ILMU BUDAYA DASAR	1	B	13014	NI MATUZ ZUHROH, M.Si	B.306
Senin	12.20 - 14.00	0765402	ISU PROFESIONAL & PROYEK G...	2	D	65021	Ali Wafa M.Kom	B.306
Senin	14.00 - 16.30	0765314	SISTEM INFORMASI GEOGRAFIS	3	E	65018	Fahrul Kurniawan M.MT	B.306
Senin	06.30 - 09.00	0765305	PEMROGRAMAN MULTIMEDIA	3	C	65019	Fressy MT	B.307
Senin	09.00 - 11.30	0765314	SISTEM INFORMASI GEOGRAFIS	3	B	65018	Fahrul Kurniawan M.MT	B.307
Senin	12.20 - 14.00	0765310	INTERAKSI MANUSIA & KOMPUT...	2	D	65005	SYAHIDUZ ZAMAN M.Kom	B.307
Senin	14.00 - 15.40	0765310	INTERAKSI MANUSIA & KOMPUT...	2	C	65005	SYAHIDUZ ZAMAN M.Kom	B.307
Senin	15.40 - 16.30	0765105	ILMU SOSIAL DASAR	1	B	13014	NI MATUZ ZUHROH, M.Si	B.307
Senin	06.30 - 09.00	0765314	SISTEM INFORMASI GEOGRAFIS	3	A	65018	Fahrul Kurniawan M.MT	B.314
Senin	09.00 - 11.30	0765305	PEMROGRAMAN MULTIMEDIA	3	D	65019	Fressy MT	B.314
Senin	12.20 - 14.50	0765303	PENGOLAHAN CITRA	3	D	65015	Dr Cahyo Crisdian	B.314
Senin	14.50 - 16.30	0765215	SITEM OPERASI	2	C	65009	null SATRIA MANDALA, MCS	B.314
Senin	06.30 - 09.00	0765303	PENGOLAHAN CITRA	3	A	65015	Dr Cahyo Crisdian	B.315
Senin	09.00 - 11.30	0765214	DESAIN BASIS DATA	3	C	65130	Ainatul M M.Kom	B.315
Senin	12.20 - 14.50	0765214	DESAIN BASIS DATA	3	B	65130	Ainatul M M.Kom	B.315
Senin	14.50 - 16.30	0765208	PENGANTAR TEKNOLOGI INFOR...	2	C	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.315
Senin	06.30 - 08.10	0765215	SITEM OPERASI	2	B	65009	null SATRIA MANDALA, MCS	B.316
Senin	08.10 - 09.50	0765208	PENGANTAR TEKNOLOGI INFOR...	2	A	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.316
Senin	09.50 - 11.30	0765113	TEOLOGI ISLAM	2	B	60002	AHMAD BARIZI, M.A	B.316
Senin	12.20 - 14.00	0765113	TEOLOGI ISLAM	2	A	60002	AHMAD BARIZI, M.A	B.316
Senin	14.00 - 15.40	0765110	STUDI FIQH	2	D	65151	Muhammad Karim M.Pd	B.316
Senin	06.30 - 08.10	0765110	STUDI FIQH	2	C	65151	Muhammad Karim M.Pd	B.317
Senin	08.10 - 09.50	0765110	STUDI FIQH	2	A	65151	Muhammad Karim M.Pd	B.317
Senin	09.50 - 11.30	0765108	STUDI AL-QUR AN	2	B	65017	Imamudin M.A	B.317
Senin	12.20 - 14.00	0765108	STUDI AL-QUR AN	2	A	65017	Imamudin M.A	B.317
Senin	14.00 - 14.50	0765105	ILMU SOSIAL DASAR	1	A	13014	NI MATUZ ZUHROH, M.Si	B.317
Senin	14.50 - 16.30	0765108	STUDI AL-QUR AN	2	C	65017	Imamudin M.A	B.317
Senin	06.30 - 07.20	0765105	ILMU SOSIAL DASAR	1	C	13014	NI MATUZ ZUHROH, M.Si	B.318
Selasa	06.30 - 09.00	0765103	BAHASA INGGRIS I	3	D	65111	Robert Aprilianto M.Pd	B.306
Selasa	09.00 - 10.40	0765404	MANAJ. PROYEK PERANGKAT LU...	2	D	65010	MUHAMMAD ANUL YAQIN, M.Kom	B.306
Selasa	10.40 - 11.30	0865122	TARBIYAH ULUL ALBAB	1	D	65007	RORO INDA MELANI, S.Kom	B.306
Selasa	12.20 - 14.00	0765402	ISU PROFESIONAL & PROYEK G...	2	A	65021	Ali Wafa M.Kom	B.306

Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Selasa	12.20 - 14.00	0765402	ISU PROFESIONAL & PROYEK G...	2	A	65021	Ali Wafa M.Kom	B.306
Selasa	14.00 - 16.30	0765314	SISTEM INFORMASI GEOGRAFIS	3	D	65018	Fahrul Kurniawan M.MT	B.306
Selasa	06.30 - 09.00	0765313	KOMPUTASI BERGERAK	3	C	65008	ZAINAL ABIDIN, S.Kom	B.307
Selasa	09.00 - 10.40	0765402	ISU PROFESIONAL & PROYEK G...	2	C	65021	Ali Wafa M.Kom	B.307
Selasa	10.40 - 11.30	0765106	ILMU BUDAYA DASAR	1	A	13014	NI MATUZ ZUHROH, M.Si	B.307
Selasa	12.20 - 14.50	0765315	SISTEM INFORMASI CERDAS	3	C	65016	Linda M.T	B.307
Selasa	14.50 - 16.30	0765309	REKAYASA PERANGKAT LUNAK	2	E	65002	FATCHURROCHMAN, M.Kom	B.307
Selasa	06.30 - 09.00	0765313	KOMPUTASI BERGERAK	3	A	65101	Syauqi, M.Kom	B.314
Selasa	09.00 - 11.30	0765305	PEMROGRAMAN MULTIMEDIA	3	A	65019	Fressy MT	B.314
Selasa	12.20 - 14.50	0765303	PENGOLAHAN CITRA	3	C	65015	Dr Cahyo Crystian	B.314
Selasa	14.50 - 16.30	0765216	RANGKAIAN DIGITAL	2	C	65011	TOTOK CHAMIDI, M.Kom	B.314
Selasa	06.30 - 09.00	0765301	KECERDASAN BUATAN	3	B	65102	Hani, M.T	B.315
Selasa	09.00 - 11.30	0765218	JARINGAN KOMPUTER	3	C	65020	Yunifa MT	B.315
Selasa	12.20 - 14.00	0765208	PENGANTAR TEKNOLOGI INFOR...	2	B	65006	MOKHAMMAD AMIN HARIYADI, M.T	B.315
Selasa	14.00 - 16.30	0765210	STRUKTUR DATA	3	A	65008	ZAINAL ABIDIN, S.Kom	B.315
Selasa	06.30 - 08.10	0765209	LOGIKA DAN ALGORITMA	2	C	65007	RORO INDA MELANI, S.Kom	B.316
Selasa	08.10 - 09.50	0765209	LOGIKA DAN ALGORITMA	2	B	65007	RORO INDA MELANI, S.Kom	B.316
Selasa	09.50 - 11.30	0765206	STATISTIKA	2	B	65113	Irwan M.Kom	B.316
Selasa	12.20 - 14.00	0765206	STATISTIKA	2	A	65113	Irwan M.Kom	B.316
Selasa	14.00 - 16.30	0765203	ALJABAR LINIER DAN MATRIKS	3	D	65003	RIRIEN KUSUMAWATI, S.Si, M.Kom	B.316
Selasa	06.30 - 09.00	0765203	ALJABAR LINIER DAN MATRIKS	3	B	65003	RIRIEN KUSUMAWATI, S.Si, M.Kom	B.317
Selasa	09.00 - 10.40	0765112	SEJARAH PERADABAN ISLAM	2	A	33333	Mukhlis Fakhruddin M.Si	B.317
Selasa	12.20 - 14.00	0765109	STUDI AL-HADITS	2	D	65199	Sugeng Ali Mansyur M.Pd	B.317
Selasa	14.00 - 14.50	0765106	ILMU BUDAYA DASAR	1	D	13014	NI MATUZ ZUHROH, M.Si	B.317
Selasa	14.50 - 15.40	0765106	ILMU BUDAYA DASAR	1	C	13014	NI MATUZ ZUHROH, M.Si	B.317
Selasa	15.40 - 16.30	0765105	ILMU SOSIAL DASAR	1	D	13014	NI MATUZ ZUHROH, M.Si	B.317
Rabu	06.30 - 07.20	0865122	TARBIYAH ULUL ALBAB	1	C	65007	RORO INDA MELANI, S.Kom	B.306
Rabu	07.20 - 09.00	0765404	MANAJ. PROYEK PERANGKAT LU...	2	E	65010	MUHAMMAD AINUL YAQIN, M.Kom	B.306
Rabu	09.00 - 10.40	0765404	MANAJ. PROYEK PERANGKAT LU...	2	C	65010	MUHAMMAD AINUL YAQIN, M.Kom	B.306
Rabu	10.40 - 11.30	0865122	TARBIYAH ULUL ALBAB	1	A	65007	RORO INDA MELANI, S.Kom	B.306
Rabu	12.20 - 14.50	0765314	SISTEM INFORMASI GEOGRAFIS	3	C	65018	Fahrul Kurniawan M.MT	B.306
Rabu	14.50 - 16.30	0765310	INTERAKSI MANUSIA & KOMPUT...	2	A	65005	SYAHIDUZ ZAMAN M.Kom	B.306
Rabu	06.30 - 08.10	0765309	REKAYASA PERANGKAT LUNAK	2	B	65002	FATCHURROCHMAN, M.Kom	B.307
Rabu	08.10 - 10.40	0765305	PEMROGRAMAN MULTIMEDIA	3	B	65019	Fressy MT	B.307



Hari	Jam	Kode MK	Nama MK	SKS	Kelas	Kode Dosen	Nama Dosen	Ruang
Kamis	12.20 - 14.50	0765210	STRUKTUR DATA	3	D	65101	Syauqi, M.Kom	B.315
Kamis	14.50 - 16.30	0765209	LOGIKA DAN ALGORITMA	2	A	65007	RORO INDA MELANI, S.Kom	B.315
Kamis	06.30 - 09.00	0765210	STRUKTUR DATA	3	B	65008	ZAINAL ABIDIN, S.Kom	B.316
Kamis	09.00 - 10.40	0765206	STATISTIKA	2	D	65113	Irwan M.Kom	B.316
Kamis	12.20 - 14.50	0765201	MATEMATIKA DASAR	3	C	65001	SUHARTONO, M.Kom	B.316
Kamis	14.50 - 16.30	0765110	STUDI FIQH	2	B	65151	Muhammad Karim M.Pd	B.316
Kamis	06.30 - 09.00	0765201	MATEMATIKA DASAR	3	A	65001	SUHARTONO, M.Kom	B.317
Kamis	09.00 - 10.40	0765109	STUDI AL-HADITS	2	A	65199	Sugeng Ali Mansyur M.Pd	B.317
Kamis	12.20 - 14.50	0765103	BAHASA INGGRIS I	3	B	65111	Robert Aprillianto M.Pd	B.317
Jumat	06.30 - 08.10	0865102	BAHASA INDONESIA	2	C	11036	Dra Siti Annijat Maimunah M.Pd	B.306
Jumat	08.10 - 10.40	0765315	SISTEM INFORMASI CERDAS	3	D	65016	Linda M.T	B.306
Jumat	13.10 - 14.50	0765310	INTERAKSI MANUSIA & KOMPUT...	2	B	65005	SYAHIDUZ ZAMAN M.Kom	B.306
Jumat	14.50 - 16.30	0865102	BAHASA INDONESIA	2	D	11036	Dra Siti Annijat Maimunah M.Pd	B.306
Jumat	06.30 - 09.00	0765306	POMROGRAMAN WEB	3	A	65121	null Bayu Adhi Nugroho M.Kom	B.307
Jumat	09.00 - 10.40	0865102	BAHASA INDONESIA	2	A	11036	Dra Siti Annijat Maimunah M.Pd	B.307
Jumat	13.10 - 15.40	0765306	POMROGRAMAN WEB	3	B	65121	null Bayu Adhi Nugroho M.Kom	B.307
Jumat	06.30 - 09.00	0765306	POMROGRAMAN WEB	3	C	65004	MUHAMMAD FAISAL, M.T	B.314
Jumat	09.00 - 10.40	0765215	SITEM OPERASI	2	E	65009	null SATRIA MANDALA, MCS	B.314
Jumat	13.10 - 14.50	0765215	SITEM OPERASI	2	D	65009	null SATRIA MANDALA, MCS	B.314
Jumat	14.50 - 16.30	0765215	SITEM OPERASI	2	A	65009	null SATRIA MANDALA, MCS	B.314
Jumat	06.30 - 09.00	0765214	DESAIN BASIS DATA	3	D	65130	Ainatul M M.Kom	B.315
Jumat	09.00 - 10.40	0765208	PENGANTAR TEKNOLOGI INFOR...	2	D	65103	Ir. Jasmani M.Kom	B.315
Jumat	13.10 - 14.50	0765113	TEOLOGI ISLAM	2	D	60002	AHMAD BARIZI, M.A	B.315
Jumat	14.50 - 16.30	0765207	METEDOLOGI PENELITIAN	2	A	65005	SYAHIDUZ ZAMAN M.Kom	B.315
Jumat	06.30 - 09.00	0765201	MATEMATIKA DASAR	3	D	65112	Dr Ali Mahmudi	B.316
Jumat	09.00 - 10.40	0765113	TEOLOGI ISLAM	2	C	60002	AHMAD BARIZI, M.A	B.316
Jumat	13.10 - 15.40	0765201	MATEMATIKA DASAR	3	B	65001	SUHARTONO, M.Kom	B.316
Jumat	06.30 - 08.10	0765112	SEJARAH PERADABAN ISLAM	2	C	33333	Mukhlis Fakhruddin M.Si	B.317
Jumat	08.10 - 09.50	0765112	SEJARAH PERADABAN ISLAM	2	B	33333	Mukhlis Fakhruddin M.Si	B.317
Jumat	13.10 - 14.50	0765108	STUDI AL-QUR AN	2	D	65017	Imamudin M.A	B.317