

# **Implementasi Metode Algoritma Genetika pada Aplikasi Otomasi Penjadwalan Perkuliah**



Andre Arsyian Jordie  
1112001029

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS BAKRIE  
JAKARTA  
2016**

## **Halaman Pernyataan Orisinalitas**

Tugas Akhir ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun yang dirujuk  
telah saya nyatakan dengan benar

Nama : Andre Arsyian Jordie

NIM : 112001029

Tanda Tangan

Tanggal : 2 Juni 2016

## Halaman Pengesahan

Judul Skripsi : Implementasi Metode Algoritma Genetika pada Aplikasi Otomatisasi Penjadwalan Perkuliahan  
Peneliti Utama : Andre Arsyian Jordie  
Jenis Kelamin : Laki-laki  
Unit Kerja : Program Studi Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Bakrie  
Alamat Kerja : Gelanggang Mahasiswa GOR Soemantri Brojonegoro Suite GF-22 Jl. H.R Rasuna Said Kav C-22, Jakarta Selatan  
Alamat Email : andrearsyanj@gmail.com  
Lama Peneliiian : 7(tujuh) Bulan  
Usulan Penelitian Tahun : 2016

Menyetujui  
Pembimbing Tugas Akhir

Jakarta, 2 Juni 2016  
Peneliti

Yusuf Lestanto, ST., M.Sc

Andre Arsyian Jordie

## Kata Pengantar

Segala puji dan syukur penulis panjatkan kehadirat Allah SWT, yang telah memberikan rahmat, berkat, taufik dan hidayah-Nya, sehingga penulisan tugas akhir dengan judul "Implementasi Metode Algoritma Genetika pada Aplikasi Otomatisasi Penjadwalan Perkuliahan" dapat diselesaikan dengan baik. Penulisan tugas akhir ini dibuat sebagai syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika Universitas Bakrie.

Penyusunan tugas akhir ini tidak lepas dari bantuan berbagai pihak yang telah mendorong dan membimbing penulis, baik dari ide, pemikiran dan semangat. Oleh karena itu dalam kesempatan ini penulis ingin mengucapkan terima kasih sebesar-besarnya kepada :

1. Orang tua penulis, Bapak Alm. dr. Arif Sulistyawan dan Ibu dr. Tri Juni Angkasawati serta adik penulis, Chelsea Tamara Aisyah, dan Enrique Muhammad Ilham yang tak henti-hentinya memberikan dukungan dan semangat sampai saat ini.
2. Bapak Yusuf Lestanto, ST., M.Sc selaku dosen pembimbing yang dengan sabar telah memberikan bimbingan, arahan, dukungan dan solusi dalam penelitian yang dilakukan hingga penulisan tugas akhir ini dapat terselesaikan.

Penulis berharap semoga semua yang ditulis dan dikerjakan di dalam tugas akhir ini dapat bermanfaat bagi pembaca dan semua pihak terkait. Tidak ada manusia yang sempurna sehingga penulis mohon maaf atas segala kekurangan yang ada. Kritik dan saran yang membangun sangat diharapkan untuk menyempurnakan tugas akhir ini

Jakarta, 2 Juni 2016

Penulis

## Halaman Pernyataan Persetujuan Publikasi

Sebagai sivitas akademik Universitas Bakrie, saya yang bertanda tangan dibawah ini :

Nama : Andre Arsyian Jordie  
NIM : 1112001029  
Program Studi : Informatika  
Fakultas : Fakultas Teknik dan Ilmu Komputer  
Judul Skripsi : Implementasi Metode Algoritma Genetika pada Aplikasi Otomatisasi Penjadwalan Perkuliahan

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Bakrie **Hak Bebas Royalti Noneksklusif** (*Non-exclusive-Free Right*) atas karya ilmiah saya yang berjudul :

### **Implementasi Metode Algoritma Genetika pada Aplikasi Otomatisasi Penjadwalan Perkuliahan**

beserta perangkat yang ada (Jika dibutuhkan). Dengan hak bebas royalti noneksklusif ini Universitas Bakrie berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir ini selama tetap mencantumkan nama penulis sebagai pemilik Hak Cipta untuk kepentingan akademis.

Demikian pernyataan ini dibuat dengan sebenar-benarnya.

Dibuat di : Jakarta  
Pada Tanggal : 2 Juni 2016

Yang Menyatakan

Andre Arsyian Jordie

# Implementasi Metode Algoritma Genetika pada Aplikasi Otomatisasi Penjadwalan Perkuliahan

---

## ABSTRAK

Penjadwalan perkuliahan di Universitas Bakrie saat ini menggunakan metode manual, yaitu dengan menetapkan satu persatu mata kuliah di ruangan dan waktu yang tersedia menggunakan Microsoft Excel. Proses tersebut memakan waktu hingga 1 minggu dan karena resource waktu dan ruangan yang terbatas maka dapat terjadi kesalahan *human error* yang memungkinkan terjadinya jadwal yang bertabrakan. Oleh karena itu dibutuhkan sebuah aplikasi yang dapat mengeluarkan keluaran berupa jadwal kuliah berdasarkan batasan atau constraint yang telah ditentukan. Aplikasi tersebut akan menggunakan metode Algoritma Genetika sebagai algoritma yang digunakan untuk membetuk jadwal perkuliahan, kemudian menggunakan HTML5 dan CSS sebagai tampilan aplikasi, PHP sebagai bahasa yang digunakan untuk mengaplikasikan metode Algoritma Genetika, dan MySQL sebagai *database*. Metode Algoritma Genetika dinilai cocok dikarenakan kemampuannya dalam memanfaatkan keterbatasan *resource*, dan dapat mudah beradaptasi dengan perubahan batasan.

**Kata Kunci:** Algoritma Genetika, Sistem Penjadwalan Perkuliahan

# Daftar Isi

Halaman Pernyataan Orisinalitas . . . . .	i
Halaman Pengesahan . . . . .	ii
Kata Pengantar . . . . .	iii
Halaman Pernyataan Persetujuan Publikasi . . . . .	iv
Abstrak . . . . .	v
Daftar Isi . . . . .	vi
<b>1 Pendahuluan</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	2
1.3 Tujuan Penelitian . . . . .	2
1.4 Batasan Masalah . . . . .	2
1.5 Kontribusi Penelitian . . . . .	3
<b>2 Tinjauan Pustaka</b>	<b>4</b>
2.1 Penelitian Terkait . . . . .	4
2.2 Penjadwalan Mata Kuliah Universitas Bakrie . . . . .	5
2.2.1 <i>Constraint</i> . . . . .	5
2.3 Algoritma Genetika . . . . .	6
2.3.1 Komparasi Algoritma . . . . .	8
2.4 <i>Software Development Life Cycle</i> . . . . .	9
2.5 <i>Tools</i> . . . . .	9
<b>3 Metodologi Penelitian</b>	<b>10</b>
3.1 Analisis Kebutuhan . . . . .	10
3.2 Desain Sistem . . . . .	10
3.2.1 <i>Use Case Diagram</i> . . . . .	11
3.2.2 <i>Flowchart</i> Program . . . . .	11
3.2.3 <i>Relasi Database</i> . . . . .	12
3.2.4 <i>Pseudocode</i> Program . . . . .	13
3.2.5 Pemodelan Masalah dan Inisialisasi . . . . .	13
3.3 Coding . . . . .	15
3.4 Pengujian . . . . .	15
3.4.1 Alat Pengujian . . . . .	16
3.5 Rencana Waktu Penelitian . . . . .	16
<b>Daftar Pustaka</b>	<b>17</b>

# Daftar Gambar

Gambar 2.1	<i>Flowchart</i> Algoritma Genetika (Liao & Sun, 2001) . . . . .	8
Gambar 2.2	Proses Metode <i>Waterfall</i> (Rizky, 2015) . . . . .	9
Gambar 3.1	<i>Use Case Diagram</i> . . . . .	11
Gambar 3.2	<i>Flowchart</i> Program . . . . .	12
Gambar 3.3	Relasi <i>Database</i> . . . . .	12
Gambar 3.4	Skema Populasi . . . . .	13
Gambar 3.5	Mutasi . . . . .	15
Gambar 3.6	<i>Crossover</i> . . . . .	15



# BAB 1

## Pendahuluan

Bab ini menjelaskan latar belakang penelitian, menarik rumusan masalah berdasarkan latar belakang, merumuskan tujuan penelitian, memberi batasan masalah dalam penelitian, dan kontribusi terhadap penelitian yang telah dilakukan sebelumnya.

### 1.1 Latar Belakang

Saat ini Universitas Bakrie menggunakan metode manual dalam melakukan perancangan jadwal perkuliahan. Metode yang dimaksud adalah dengan merumuskan durasi mata kuliah, ruangan yang tersedia, hari dan jam yang disesuaikan dengan ketersediaan dosen, kemudian menyusun jadwal mata kuliah berdasarkan batasan-batasan tersebut. Metode ini memiliki beberapa kelemahan diantaranya adalah memakan banyak waktu dan tenaga yang dihabiskan untuk menyusun jadwal tersebut. Kelemahan lainnya adalah adanya kemungkinan jadwal yang bertabrakan sehingga berakibat mahasiswa tidak dapat mengambil mata kuliah. Selain itu, metode tersebut tidak lepas dari adanya *human error* karena banyaknya batasan dan syarat yang harus dipenuhi. Berdasarkan wawancara yang dilakukan oleh Doni (2015), batasan dan syarat yang dimaksud adalah:

1. Dosen yang berbeda tidak boleh mengajar mata kuliah yang sama pada satu *timeslot*
2. Setiap mata kuliah yang ada di dalam satu angkatan program studi tidak boleh dijadwalkan pada hari dan jam yang sama
3. Kapasitas ruang kelas harus lebih besar sama dengan jumlah mahasiswa yang akan menggunakan kelas tersebut
4. Frekuensi mengajar tiap dosen sebaiknya tidak melebihi batas yang telah ditetapkan yaitu 1 hari 1 kali

Metode Algoritma Genetika merupakan metode yang telah berhasil dimanfaatkan untuk menyelesaikan masalah kombinatorial dan mampu menangani ruang pencarian yang luas dalam masalah penjadwalan di dunia nyata (Tormos et al., n.d.). Algoritma Genetika merupakan salah satu jenis dari *Evolutionary Programming*, yang mana

berbasis pada teori evolusi alam. Algoritma Genetika bekerja dengan melakukan seleksi, *crossover*, dan mutasi, sehingga individu yang terkuat yang akhirnya muncul pada generasi terbaru. Nilai *fitness* digunakan sebagai indikator dari kekuatan individu pada setiap generasi. Setiap generasi akan dilakukan tahap seleksi, sehingga hanya terpilih individu yang kuat dan individu yang lemah akan tersingkirkan. Kemudian dilakukan *crossover* untuk dapat menghasilkan individu yang memiliki nilai *fitness* yang lebih baik. Tahap selanjutnya adalah mutasi, yaitu faktor dari luar yang akan mempengaruhi hasil *fitness* dari individu tersebut. Proses ini akan diulang hingga mendapatkan individu dengan nilai *fitness* yang paling besar.

Berdasarkan latar belakang tersebut, penulis akan membuat sebuah aplikasi berbasis web yang mengaplikasikan metode algoritma genetika yang dapat mengeluarkan hasil berupa jadwal perkuliahan selama satu semester dan jadwal yang dihasilkan terhindar dari masalah seperti jadwal yang bertabrakan dan dapat memaksimalkan penggunaan *resource* ruangan dan waktu.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah adalah sebagai berikut.

1. Bagaimana membangun aplikasi yang mengimplementasikan metode algoritma genetika untuk menghasilkan jadwal perkuliahan di Universitas Bakrie?

## 1.3 Tujuan Penelitian

Berdasarkan rumusan masalah diatas, maka tujuan penelitian adalah

1. membangun aplikasi untuk menghasilkan keluaran berupa jadwal perkuliahan dengan mengimplementasikan metode Algoritma Genetika

## 1.4 Batasan Masalah

Batasan masalah dalam penelitian ini adalah

1. Aplikasi dibuat dengan berbasis *website*
2. Penjadwalan hanya pada perkuliahan normal saja, tidak termasuk ujian dan mata kuliah pengganti
3. Jadwal yang dibuat berdasarkan kurikulum 2015 untuk semester ganjil
4. Data nama kuliah dan dosen pengajar dibuat secara manual menggunakan excel
5. Asumsi semua dosen menerima hasil jadwal perkuliahan
6. Tidak menghubungkan dengan sistem KRS ataupun sistem internal Universitas Bakrie

7. Aplikasi dijalankan dalam jaringan lokal
8. Keamanan program hanya sebatas mencegah adanya SQL Injection
9. Asumsi mahasiswa tidak mengambil semester diluar semester aktif mahasiswa tersebut

## **1.5 Kontribusi Penelitian**

Kontribusi penelitian ini adalah menghasilkan aplikasi berbasis website yang mengimplementasikan metode algoritma genetika yang dapat secara otomatis menghasilkan jadwal perkuliahan dan analisa dari pengaruh aplikasi tersebut terhadap penjadwalan perkuliahan.

## BAB 2

# Tinjauan Pustaka

Bab ini membahas penelitian terkait metode algoritma genetika, kemudian membahas proses dan sistem penjadwalan perkuliahan di Universitas Bakrie, menjelaskan definisi serta proses algoritma genetika, metode pengembangan software (*software development life cycle*) dan *tools* pengembangan.

## 2.1 Penelitian Terkait

Doni(2015) membangun aplikasi penjadwalan perkuliahan Universitas Bakrie dengan menggunakan metode integer 0-1. Program tersebut berhasil menghasilkan jadwal perkuliahan dalam waktu 19.10 menit. Namun sistem milik Doni tidak memiliki fungsi untuk validasi terhadap jadwal mahasiswa sehingga ada kemungkinan terjadi jadwal yang bertabrakan. Selain itu, Doni menggunakan pemrograman struktural dalam pembangunan aplikasi sehingga kurang fleksibel apabila akan diintegrasikan dengan sistem informasi lain.

(Rohmasnyah, Soebroto, & Hidayat, n.d.) berhasil melakukan implementasi algoritma genetika pada penyusunan jadwal KRS berbasis website. Penelitian tersebut menunjukkan bahwa semakin besar nilai probabilitas *crossover* maka nilai *fitness* cenderung semakin tinggi. Dan semakin besar nilai jumlah generasi maka semakin besar juga nilai *fitness* yang dihasilkan. Disamping itu, parameter nilai probabilitas mutasi yang tinggi mengakibatkan penurunan nilai *fitness*. Penelitian tersebut juga menyimpulkan bahwa semakin besar jumlah populasi maka semakin besar juga nilai *fitness*-nya. Akan tetapi, aplikasi tersebut memiliki kelemahan yaitu keterbatasan PHP dalam memproses individu. Jumlah individu maksimal yang dapat di proses adalah 40 individu. Apabila lebih dari 40 individu maka akan terjadi *error Request Time Out*.

(Marwana, 2012) melakukan optimasi penjadwalan mata kuliah dengan algoritma genetika namun menambahkan parameter yaitu jadwal ketersediaan mahasiswa. Dalam penelitian Marwana, penelitian menggunakan masukan berupa mahasiswa dengan jumlah mata kuliah, kode dan nama kuliah, serta hari dan sesi jadwal yang bisa dihadiri oleh mahasiswa tersebut. Marwana memiliki 3 kondisi selesai, yaitu:

1. Apabila secara berturut-turut beberapa generasi dari populasi tersebut mencapai nilai *fitness* terbaik dan tidak mengalami perubahan.

2. Jumlah iterasi atau generasi tercapai.
3. Nilai *fitness* terbaik minimal telah tercapai.

Penelitian tersebut membuktikan bahwa algoritma genetika mampu beradaptasi dengan perubahan batasan, dalam hal ini adalah adanya penambahan jadwal mahasiswa disamping jadwal perkuliahan.

## 2.2 Penjadwalan Mata Kuliah Universitas Bakrie

Penjadwalan kuliah adalah merupakan kombinasi dari pengalokasian sumber daya mata kuliah, dosen, mahasiswa, dan ruang kelas tanpa adanya pelanggaran *constraint* atau batasan dalam waktu satu *timeslot*. Sedangkan *timeslot* itu sendiri adalah satu kolom waktu pada jadwal mata kuliah. Tujuan dari pembuatan jadwal mata kuliah adalah dengan memasukkan kombinasi *resource* kedalam *timeslot* yang tersedia dengan mengikuti *constraint* yang berlaku. Biro Administrasi Akademik(BAA) bertanggung jawab terhadap penjadwalan perkuliahan di Universitas Bakrie. Pada Universitas Bakrie terdapat 2 sesi waktu, yaitu waktu siang (pukul 7.30 - pukul 18.00) dan waktu malam (pukul 18.30 - pukul 22.00). Berdasarkan wawancara dari (Rizky, 2015) kepada BAA, langkah-langkah yang dilakukan dalam menyusun jadwal perkuliahan di Universitas Bakrie adalah sebagai berikut:

1. Setiap kaprodi menetapkan jumlah kelas dan dosen yang ditugaskan untuk mengajar mata kuliah.
2. Kaprodi menyerahkan informasi tersebut kepada BAA.
3. Dengan menggunakan *Software Microsoft Excel*, BAA mengalokasikan waktu dan ruang untuk mata kuliah pada jadwal yang disusun.
4. Jika mata kuliah X tidak mendapatkan *slot*, maka mata kuliah Y akan dipindahkan ke *slot* lain agar mata kuliah X dapat masuk kedalam *slot* tersebut.
5. Jadwal perkuliahan yang telah disusun diperiksa kembali dan apabila memungkinkan akan dilakukan optimasi jadwal.
6. Jadwal perkuliahan yang telah selesai diberikan kepada masing-masing dosen pengajar mata kuliah.

### 2.2.1 *Constraint*

*Constraint* atau batasan adalah syarat yang dapat dipenuhi dalam pembuatan jadwal. Dalam penyusunan jadwal di Universitas Bakrie, dapat dibagi menjadi dua jenis *constraint*, yaitu *soft constraint* dan *hard constraint*.

## Hard Constraint

*Hard Constraint* merupakan batasan yang wajib dipenuhi dalam penjadwalan perkuliahan. *Hard Constraint* yang dimaksud meliputi:

1. Semua mata kuliah pada kurikulum harus diajarkan.
2. Setiap dosen hanya mengajar sekali pada satu *timeslot*.
3. Setiap ruangan hanya dipakai sekali pada satu *timeslot*.
4. Setiap kelas hanya mempunyai satu sesi kuliah pada satu *timeslot*.
5. Tidak ada sesi kuliah pada saat waktu solat jumat, yaitu pada hari jumat pukul 11.00 - 13.30.
6. Jumlah kursi dalam ruangan harus lebih banyak sama dengan jumlah kuota mahasiswa dalam perkuliahan.

## Soft Constraint

*Soft constraint* adalah batasan yang dapat dilewati atau tidak wajib dipenuhi namun dapat dipenuhi apabila memungkinkan. *Soft constraint* dapat berfungsi sebagai optimasi jadwal perkuliahan. Dalam penyusunan jadwal perkuliahan di Universitas Bakrie, *soft constraint* meliputi:

1. Frekuensi mengajar dosen hanya sekali dalam sehari.
2. Pembagian jadwal mata kuliah pada jurusan dibagi rata setiap hari, sehingga dalam waktu senin - jumat tidak ada penumpukan jadwal mata kuliah.
3. Mata kuliah dengan beban 4 SKS dimasukkan pada sesi waktu siang dengan batasan pukul 7.30 - 11.00 dan ditempatkan pada ruang kelas di lantai atas.
4. Mata kuliah dengan beban 3 SKS dimasukkan pada sesi waktu siang dengan batasan pukul 7.30 - 15.00 tanpa ada prioritas ruang kelas atas maupun bawah.
5. Mata kuliah dengan beban 2 SKS dimasukkan pada sesi waktu siang dengan batasan pukul 13.00 - 18.00 tanpa ada prioritas ruang kelas atas maupun bawah.
6. Mata kuliah semester tujuh berada di waktu malam.
7. Mata kuliah semester tujuh yang berada di waktu malam di tempatkan di ruang kelas atas.

## 2.3 Algoritma Genetika

Algoritma Genetika merupakan algoritma yang menerapkan pemahaman mengenai evolusi alamiah pada tugas pemecahan masalah (*problem solving*) (Nugraha, 2008). Algoritma genetika merupakan metode untuk mendapatkan kombinasi optimal dari suatu masalah yang memiliki banyak kemungkinan solusi. Pada dasarnya, Algoritma Genetika bekerja seperti teori evolusi pada biologi. Teori evolusi menyatakan bahwa yang terkuat yang dapat bertahan hidup, dan dalam perjalanan menjadi yang terkuat, kromosom akan menjalani proses reproduksi, *crossover*, dan mutasi. Algoritma Genetika memiliki proses yang sama dengan teori evolusi, namun parameter tersebut disesuaikan dengan masalah yang akan diselesaikan. Parameter kekuatan dari suatu kromosom adalah nilai *fitness*, dan nilai ini adalah penentu apakah suatu kromosom sudah mencapai nilai terbaik dalam iterasi generasinya. Dalam pemecahan masalahnya, Algoritma Genetika melakukan lima tahap, yaitu:

1. Inisialisasi : Kromosom dibentuk dengan melakukan kombinasi dari gen-gen yang telah diberi nilai awal secara random. Sebuah populasi akan terbentuk dari kumpulan beberapa kromosom tersebut. Jumlah kromosom dapat menyesuaikan kemampuan komputasi, namun penelitian oleh (Rohmasnyah et al., n.d.) menyatakan bahwa semakin besar jumlah populasinya, maka semakin besar pula nilai *fitness*-nya.
2. Evaluasi : Kromosom yang telah terbentuk akan di evaluasi nilai *fitness*-nya. Nilai fitness berfungsi sebagai parameter kekuatan dari suatu kromosom.
3. Seleksi : Setelah mendapatkan nilai *fitness* pada seluruh kromosom dalam populasi, maka kromosom yang memiliki nilai *fitness* paling rendah akan dihilangkan dari populasi. Populasi yang tersisa setelah proses seleksi ini disebut dengan populasi induk. Agar jumlah populasi tetap sama, maka akan di *generate* kembali kromosom baru dengan nilai random.
4. Mutasi : Proses mutasi adalah penggantian nilai pada kromosom. Nilai penggantian merupakan nilai random. jumlah kromosom yang akan di mutasi adalah sesuai dengan nilai *mutation rate*.
5. *Crossover* : *Crossover* atau kawin silang adalah proses menukar gen pada kromosom satu dengan kromosom yang lain. Jumlah gen yang ditukar adalah sebanyak jumlah nilai *crossover rate*.

Seluruh proses evaluasi, seleksi, mutasi, dan crossover akan dilakukan proses iterasi dan setiap iterasi yang dilakukan disebut dengan Generasi.

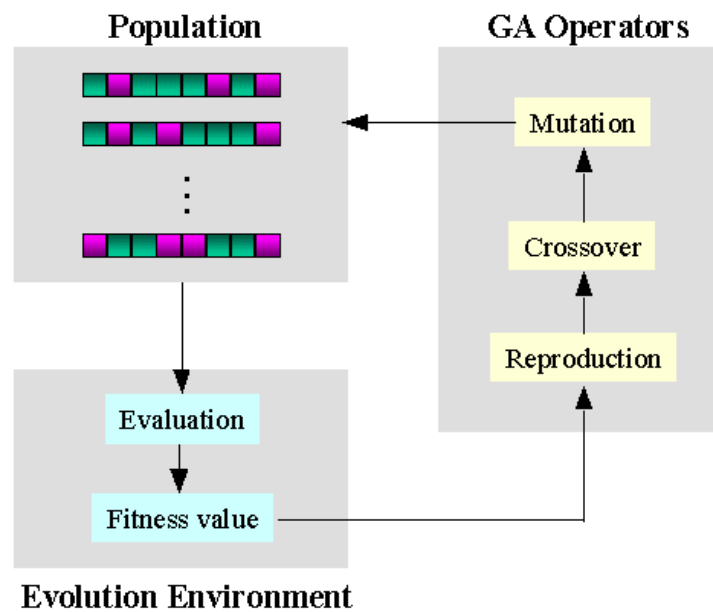


Figure 2.1 *Flowchart* Algoritma Genetika (Liao & Sun, 2001)

### 2.3.1 Komparasi Algoritma

(William Aprilius, Lorentzo Augustino, 2013) melakukan penelitian dengan membuat aplikasi jadwal perkuliahan dengan metode Max-Min Ant System. Sistem yang dibuat dapat menghasilkan jadwal kuliah, namun tidak dapat mengeluarkan jadwal yang optimal. Jadwal yang dihasilkan tidak seimbang dan meskipun dengan menambahkan ruangan tetap tidak dapat mencapai hasil yang optimal.

(Ali, Umrah, & Senggarang, 2013) membandingkan algoritma genetika dengan *particle swarm optimization* dalam menghasilkan jadwal mata kuliah. Dalam penelitian tersebut algoritma genetika mampu menghasilkan jadwal perkuliahan dalam waktu 8,79 detik, sedangkan PSO menyelesaikan dalam waktu 41,636 detik dengan besar data yang sama. Algoritma genetika memiliki nilai *fitness* yang lebih baik, namun PSO memiliki standar deviasi yang lebih rendah sehingga hasil *fitness* yang dihasilkan menjadi lebih stabil.

Table 2.1 Komparasi Algoritma

Metode	Kelemahan	Kelebihan
Max-Min ANT System	Tidak dapat menghasilkan solusi jadwal mata kuliah yang optimal	Berhasil mengeluarkan jadwal kuliah dengan batasan-batasan tertentu
Particle Swarm Optimization	Jadwal yang dihasilkan masih terjadi tabrakan	Memiliki hasil fitness yang stabil
Algoritma Genetika	Proses generasi jadwal dengan menggunakan banyak individu menjadi sangat berat	Dapat beradaptasi dengan berbagai macam constraint



## 2.4 *Software Development Life Cycle*

*Software Development Life Cycle* adalah proses yang dilakukan dalam melakukan pengembangan *software*. Proses ini diperlukan agar pengembangan *software* menjadi sistematis dan terstruktur. Metode yang digunakan dalam pengembangan aplikasi ini adalah metode *waterfall*. Metode *waterfall* merupakan pendekatan sistematis atau secara bertahap di dalam pengembangan *software*-nya yang dimulai pada level sistem dan berkembang melalui analisis, desain, coding, testing, dan support (Rizky, 2015). Metode *waterfall* dipilih karena kebutuhan sistem sudah lengkap dan fase-fase pengembangan *software* tidak perlu dilakukan berulang kali. Metode *waterfall* memiliki empat fase yaitu analisis, desain sistem, *coding*, dan pengujian.

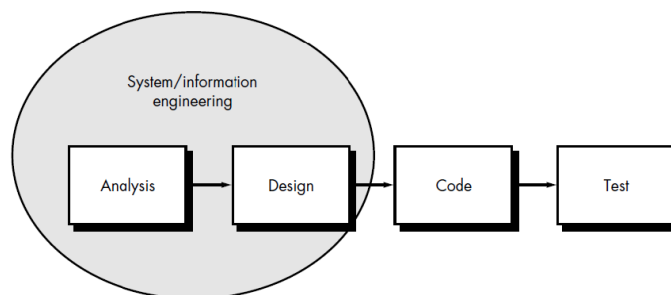


Figure 2.2 Proses Metode *Waterfall* (Rizky, 2015)

## 2.5 *Tools*

Pembuatan *software* penjadwalan perkuliahan menggunakan *tools* atau alat-alat sebagai berikut:

1. XAMPP : *Software* penyedia layanan *web server* APACHE dan layanan *Database* MySQL.
2. HTML : Bahasa pemrograman untuk peletakan konten dalam halaman *web*.
3. CSS : Bahasa pemrograman untuk *styling* halaman *web*.
4. Javascript : Bahasa pemrograman untuk menyediakan konten yang bersifat *user interactive*.
5. PHP : Bahasa Pemrograman untuk menghubungkan halaman *web* dengan *database*.
6. Bootstrap : *Framework* untuk HTML, CSS, dan Javascript.
7. Notepad++ : *Software* untuk membuat dan melakukan edit *code*.
8. Google Chrome : *browser* untuk menampilkan *web*.

## BAB 3

# Metodologi Penelitian

Bab ini membahas perancangan pengembangan program dengan metode *waterfall*. Perancangan dimulai dengan proses analisis kebutuhan, kemudian desain sistem dan implementasi algoritma genetika pada sistem, kemudian *coding* dan diakhiri dengan proses *testing* terhadap program yang telah dibuat.

### 3.1 Analisis Kebutuhan

Pada tahap ini akan dilakukan pengumpulan data yang berupa kebutuhan sistem. Data yang dimaksud antara lain *software requirement*, jadwal perkuliahan semester genap 2015/2016, data mata kuliah beserta dosen yang mengajar, data informasi mata kuliah, data ruangan, data *constraint*, data *timeslot*. Data dikumpulkan dan diolah menjadi *Software Requirement Specification* dan elisitasi yang menjadi acuan dari pengembangan program penjadwalan perkuliahan. Data tersebut dikumpulkan dengan metode:

- Studi Pustaka  
*Software Requirement* didapatkan dari studi literatur penelitian (Rizky, 2015) yang telah melakukan wawancara terhadap Badan Perencanaan Akademik Universitas Bakrie.
- Wawancara  
Wawancara dilakukan untuk mendapatkan data tabel mata kuliah beserta dosen yang mengajar.
- Observasi  
Melakukan observasi secara langsung sebagai mahasiswa Universitas Bakrie untuk memahami jadwal perkuliahan.

### 3.2 Desain Sistem

User dari program adalah staff Bagian Perencanaan Akademik Universitas Bakrie. Program penjadwalan perkuliahan dimulai dengan *user* masuk ke dalam sistem dengan melakukan login menggunakan *Username* dan *Password* yang telah disediakan.

Kemudian *User* akan melakukan Upload file Tabel Dosen dalam bentuk .csv. File Tabel Dosen ini berupa kombinasi kode mata kuliah dan dosen yang mengajar. Kemudian setelah proses *upload* selesai maka user akan melakukan proses pembuatan jadwal dengan menekan tombol *Generate Jadwal*. User menunggu hingga proses selesai dan dialihkan ke halaman *Download* jadwal. Pada halaman tersebut user akan dapat melakukan *download* jadwal kuliah yang telah berhasil dibuat dalam bentuk .xls. Rancangan alur sistem dapat dilihat pada figure 3.2

### 3.2.1 Use Case Diagram

Berikut ini merupakan *Use Case Diagram* dari program penjadwalan perkuliahan:

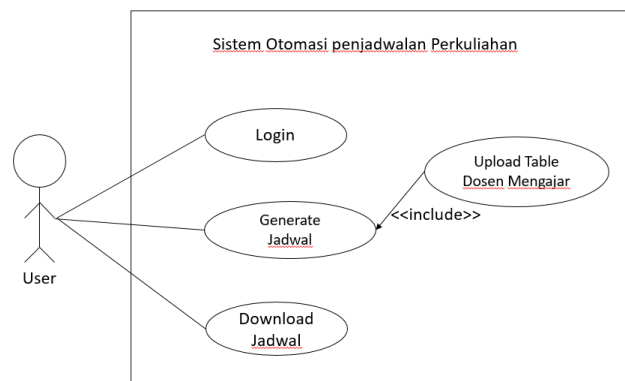


Figure 3.1 Use Case Diagram

### 3.2.2 Flowchart Program

Berikut ini merupakan *flowchart* dari program penjadwalan perkuliahan:

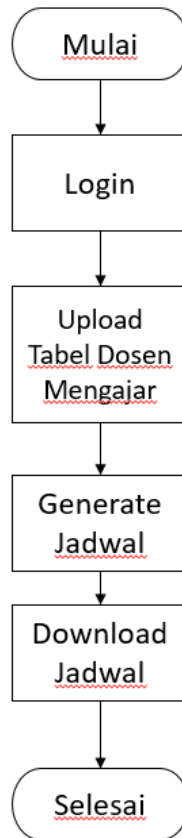


Figure 3.2 *Flowchart Program*

### 3.2.3 Relasi Database

Berikut ini merupakan relasi *database* untuk menyimpan data perkuliahan:

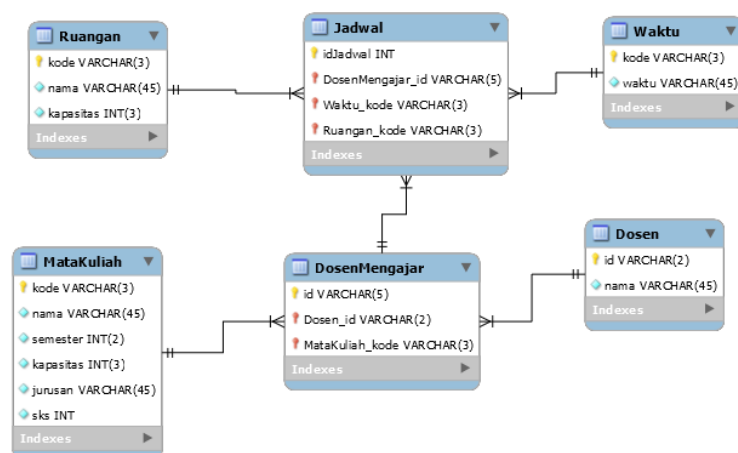


Figure 3.3 *Relasi Database*

### 3.2.4 Pseudocode Program

Berikut ini merupakan Pseudocode dari program penjadwalan perkuliahan berdasarkan *flowchart* program:

```
input Username, Password
if login success then
    | uploadTabelDosenMengajar()
    | generateJadwal()
    | downloadJadwal()
else
    | login()
end
```

*Pseudocode 3.1: Pseudocode Program*

### 3.2.5 Pemodelan Masalah dan Inisialisasi

Berdasarkan kebutuhan penjadwalan perkuliahan maka disusun variabel sesuai dengan *entity* jadwal kuliah. Berikut ini adalah variabel yang akan diubah menjadi Gen dalam algoritma genetika:

- Dosen Mengajar(DM) : Merupakan kombinasi dari kode mata kuliah dengan kode dosen yang mengajar.
- Ruangan(R) : inisialisasi kapasitas ruangan dengan kode ruangan.
- Waktu(W) : inisialisasi waktu kegiatan pembelajaran dan kode waktu.

Setiap Gen Dosen Mengajar, Ruangan, dan Waktu akan menyusun sebuah kromosom, sehingga setiap kromosom mengandung tiga Gen yaitu DM,R,dan W. Kemudian setiap kromosom tersebut akan bergabung dengan kromosom lain hingga membentuk individu. Individu inilah yang akan menjadi jadwal perkuliahan. Dari beberapa individu akan membentuk sebuah populasi. Setelah dilakukan iterasi Generasi, maka Individu dengan nilai *fitness* terbaik lah yang akan dipilih menjadi jadwal perkuliahan.

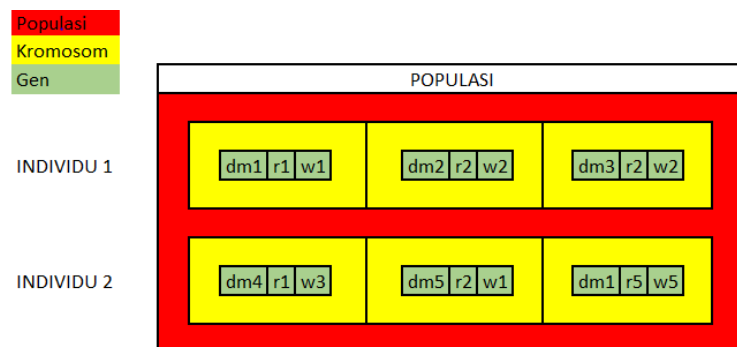


Figure 3.4 Skema Populasi

### Inisialisasi Kromosom

Tahap pertama dalam algoritma genetika adalah melakukan inisialisasi kromosom. Gen yang dibentuk dalam kromosom diisi dengan nilai *random* namun tetap mengacu pada Tabel Dosen Mengajar yang telah di *upload*. Jumlah kromosom adalah sebanyak baris dalam tabel dosen mengajar. Jumlah individu dalam populasi akan di inisialisasi dalam *coding* dan akan diuji dalam tahap implementasi.

### Evaluasi *Fitness*

Setelah populasi dibentuk, maka tiap individu didalamnya akan diuji nilai *fitness*-nya. Nilai *fitness* semakin kecil akan semakin baik. Nilai *fitness* diukur dengan menggunakan rumus blabla. Variable yang mempengaruhi Nilai *fitness* adalah penalti yang diberikan terhadap kromosom yang melanggar *hard constraint* maupun *soft constraint*.

Nilai *fitness* ditentukan dengan rumus :

$$f(g) = 1 / (1 + \sum P_i v_i(g)) \quad (3.1)$$

Dimana:

- $P_i$  = Pinalti yang diberikan untuk aturan  $i$
- $v_i(g) = 1$  apabila jadwal  $g$  melanggar aturan  $i$ , dan bernilai 0 jika sebaliknya

Dengan tabel pinalti sebagai berikut :

Table 3.1 Tabel Pinalti

Batasan	Pinalti
Dosen mengajar lebih dari sekali dalam satu timeslot	3
Ruangan dipakai lebih dari sekali dalam satu timeslot	3
Kelas memiliki lebih dari satu kuliah dalam satu timeslot	3
Jumlah kursi kurang dari jumlah mahasiswa	2
Dosen mengajar lebih dari sekali dalam sehari	1
Mata kuliah 4 SKS di lantai bawah dan diluar jam 7.30 - 11.00	1
Mata kuliah 3 SKS diluar jam 7.30 - 15.00	1
Mata kuliah 2 SKS diluar jam 7.30 - 18.00	1
Mata kuliah semester tujuh di waktu siang	2
Mata kuliah semester tujuh pada jam malam ditempatkan di ruang bawah	2

### Seleksi

Tahap seleksi merupakan tahap menghilangkan individu yang memiliki Nilai *fitness* paling rendah. Setelah individu tersebut dihapus, maka akan diganti dengan individu baru dengan nilai random.

## Mutasi

Mutasi dilakukan dengan mengganti nilai pada gen dengan nilai *random*. Jumlah banyaknya gen yang dilakukan mutasi adalah sebanyak presentase *mutation rate*. *mutation rate* ditentukan dalam pengujian dan di inialisasi secara *hard code*.



Figure 3.5 Mutasi

## Crossover

Proses *crossover* dilakukan dengan menentukan satu titik dimana akan dilakukan persilangan, kemudian gen tersebut akan ditukar dengan kromosom lain pada tempat yang sama dan dengan jumlah yang sama. Penentuan titik persilangan didapatkan secara random, dan jumlah gen yang disilangkan adalah sebanyak presentasi dalam *crossover rate*. Nilai *crossover rate* didapatkan dengan melakukan pengujian dan diinisialisasi secara *hard code*.

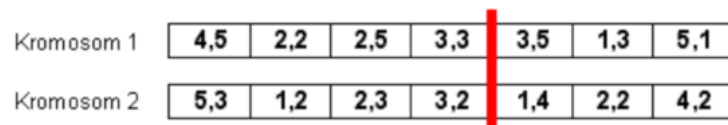


Figure 3.6 Crossover

## 3.3 Coding

Proses *coding* adalah proses pembuatan aplikasi berdasarkan rancangan sistem hingga menjadi sebuah program. Kerangka *coding* dibentuk dari *pseudocode*. *Pseudocode* adalah susunan algoritma yang dibuat dengan bahasa manusia yang akan dikonversi ke bahasa pemrograman. Tahapan *coding* dilakukan menggunakan aplikasi Notepad++ sebagai *code editor* dan aplikasi Google Chrome sebagai *software* untuk menampilkan hasil *coding*.

## 3.4 Pengujian

Pengujian adalah tahap untuk memastikan program yang berjalan sudah sesuai dengan sistem yang dirancang dan dapat mengeluarkan *output* yang dikehendaki. Proses pengujian dilakukan dengan metode *whitebox testing*. *Whitebox Testing* adalah metode untuk melakukan pengujian program dengan mengevaluasi kinerja sistem secara internal. Metode ini dipilih karena pengujian dilakukan oleh pengembang program yang mengetahui rancangan dan implementasi sistem secara menyeluruh,

sehingga apabila terjadi kesalahan maupun malfungsi akan dapat segera ditangani. *Whitebox testing* dilakukan dengan dua cara, yaitu *regression testing* dan *integration testing*. Data pengujian yang digunakan adalah data jadwal perkuliahan Universitas Bakrie Semester Genap tahun 2015/2016.

### 3.4.1 Alat Pengujian

Pengujian Dilakukan dengan menggunakan satu unit laptop Sony Vaio DUO dengan spesifikasi Intel Core i5 @1.7GHz, RAM 4 GB, Hardisk SSD 120GB dengan sistem operasi Microsoft Windows 10.

## 3.5 Rencana Waktu Penelitian

Berdasarkan rancangan penelitian yang telah disusun maka *timeline* pengerjaan tugas akhir adalah sebagai berikut:

Table 3.2 *Timeline* Pengerjaan

Kegiatan	Mei				Juni				Juli			
	1	2	3	4	1	2	3	4	1	2	3	4
Pengumpulan Data	■											
Penyusunan Proposal		■	■	■								
Seminar Proposal					■							
Pembuatan Program					■	■	■	■				
Perbaikan Program								■				
Perbaikan Laporan Tugas akhir									■	■	■	■



# Daftar Pustaka

- Ali, R., Umrah, H., & Senggarang, J. P. (2013). Perbandingan Algoritma Genetika dan Particle Swarm Optimization dalam Optimasi Penjadwalan Matakuliah. , 1–7.
- Marwana. (2012). Optimasi Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika Berbasis Permintaan Mahasiswa. *Prosiding Konfrensi Nasional Ilmu Komputer*, 31–34.
- Nugraha, I. (2008). Aplikasi Algoritma Genetik Untuk Optimasi Penjadwalan Kegiatan Belajar Mengajar. *Makalah IF2251 Strategi Algoritmik*.
- Rizky, D. K. (2015). *Rancang Bangun Aplikasi Penjadwalan Mata Kuliah Menggunakan Metode Pemrograman Integer 0-1 di Universitas Bakrie*.
- Rohmasnyah, F., Soebroto, A. A., & Hidayat, N. (n.d.). Implementasi Algoritma Genetika pada Sistem Penyusunan Jadwal KRS Berbasis Website di Program Teknologi Informasi dan Ilmu Komputer.
- Tormos, P., Lova, A., Barber, F., Ingolotti, L., Abril, M., & Salido, M. A. (n.d.). A Genetic Algorithm for Railway Scheduling. , 276(2008), 255–276.
- William Aprilius, Lorentzo Augustino, O. Y. M. H. (2013). Implementasi Algoritma MAX-MIN Ant System pada Penjadwalan Mata Kuliah. , V(2), 48–53.