

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Terdapat beberapa kesamaan dari penelitian yang mendukung penelitian skripsi ini, antara lain : Purwana, N., Djamal, C, E & Renaldi, F., (2016) dalam penelitiannya berjudul “Optimalisasi Penempatan Dosen Pembimbing dan Penjadwalan seminar Tugas Akhir Menggunakan Algoritma Genetika”. penelitian ini berfokus pada penempatan dosen dan penjadwalan seminar tugas akhir Universitas Jendral Achmad jurusan informatika. Pada penelitian permasalahannya yaitu terdapat 77 peserta seminar, 13 dosen dan 10 bidang penelitian, maka total penempatan dosen sebanyak  $77^{130}$  jika terdapat dua pembimbing dan dua dosen penguji lalu satu hari terdapat 8 jam kegiatan, dua ruangan yang digunakan maka dalam 5 hari menghasilkan kombinasi  $77^{320}$ . Untuk dapat menyelesaikannya penelitian ini menggunakan dua algoritme genetika yang masing-masing digunakan untuk penempatan dosen pembimbing dan penjadwalan seminar tugas akhir. Dari hasil yang diperoleh pada penelitian ini menghasilkan solusi yang baik dengan durasi program yang dapat disesuaikan dengan panjang kromosom.

Penelitian kedua dilakukan oleh Setyaningsih (2014) yang berjudul “*System Application of Genetic Algorithm for Schedulling Optimization Study Using Java*”. penelitian ini berfokus pada penjadwalan perkuliahan di *departement of computer systems* (UNTAN) menggunakan algoritma genetika. permasalahan penelitian ini yaitu penjadwalan perkuliahan harus mempertimbangkan jumlah mahasiswa dengan dosen yang tidak sebanding dengan jadwal mengajar, jumlah kelas, serta waktu kuliah yang telah ditentukan. Masalah yang muncul lagi ketika dosen mampu untuk mengajar di semua jadwal perkuliahan dan apakah kelas yang telah dijadwalkan pada waktu yang sama juga terdapat jadwal mengajar dikelas lain. Untuk menyelesaikannya penelitian ini memiliki beberapa tahap seperti kuliah di mulai pukul 07.00-16.00, durasi kuliah 1 jam, dosen tidak diperbolehkan ada jadwal mengajar yang sama pada waktu yang sama. Seorang dosen bisa mengajar dengan beberapa kelas, untuk dosen yang berhalangan hadir pada waktu tertentu harus dihindari pembentukan jadwal mengajar. Penelitian ini menghasilkan penjadwalan yang optimal dan juga dapat mencegah tumpang tindih jadwal dosen atau kelas.

Penelitian ketiga dilakukan oleh Seisarrina (2016) yang berjudul “Penjadwalan Pengawas Ujian Semester Menggunakan Algoritma Genetika”. penelitian ini menggunakan algoritme genetika yang berfokus pada penjadwalan pengawas ujian di Fakultas Ilmu Komputer Universitas Brawijaya. Untuk inialisasi koromosom ini memiliki dua komponen yaitu dosen dan karyawan. Hasil dari penelitian ini menghasilkan penjadwalan yang optimal dengan solusi mencari nilai *fitness* tertinggi, yaitu memiliki jadwal pengawas ujian yang paling sedikit mengalami pelanggaran. Dengan populasi terbaik 900, jumlah generasi 300, niali *cr* terbaik sebesar 0,4, nilai *mr* terbaik sebesar 0,6.

Penelitian keempat dilakukan oleh Chircu (2010) yang berjudul “*Using Genetic Algorithms for Producing Scheduling*”. Permasalahan penelitian ini adalah memproduksi dengan 6 mesin dimana membutuhkan waktu dan sumberdaya yang cukup agar jadwal produksi efisien dan efektif. Representasi solusi yang digunakan yaitu  $G1$  sampai  $G15$  dimana  $G_i$  adalah struktur yang mengkodekan nomor seri, jenis dan waktu. Disini terdapat 5 seri waktu untuk setiap jenis. Misalkan  $G_i=(s_i, p_i[t_1, t_2, t_3, t_4, t_5])$  dimana  $p_i$  merepresentasikan jenis produk,  $t_i$  adalah waktu,  $s_i$  adalah nomor seri. Pada Tabel 2.1 dapat dijelaskan secara singkat kajian pustaka yang ada.

**Tabel 2.1 Kajian Pustaka**

NO	Judul	Objek	Metode	Hasil
1	Optimalisasi Penempatan Dosen Pembimbing dan Penjadwalan seminar Tugas Akhir Menggunakan Algoritma Genetika	Penempatan dosen pembimbing dan penjadwalan seminar tugas akhir	Algoritme Genetika	Pengujian yang dilakukan dengan 100 kali evolusi dengan lima kali pengujian mengeluarkan solusi yang berbeda.
2	<i>System Application of Genetic Algorithm for Scheduling Optimization Study Using Java (Case Study: Department of Computer System UNTAN)</i>	Penjadwalan kuliah pada perguruan tinggi	Algoritme Genetika	Dengan kromosom yang berisi 2 komponen yaitu dosen dan jadwal jam mengajar permasalahan dapat diselesaikan dengan algoritme genetika.
3	Penjadwalan Pengawas Ujian Semester Menggunakan Algoritma Genetika (Studi Kasus: Fakultas Ilmu Komputer Universitas Brawijaya).	Penjadwalan ujian pada perguruan tinggi	Algoritme Genetika	Penelitian ini menunjukkan bahwa algoritme genetika merupakan metode yang baik dalam menyelesaikan masalah penjadwalan yang memakan waktu
4	<i>Using Genetic Algorithms for Productions Scheduling</i>	Penjadwalan produksi	Algoritme Genetika	Dengan algoritme genetika masalah dapat terselesaikan dengan 6 mesin dimana membutuhkan waktu dan

				sumberdaya yang cukup agar jadwal produksi efisien dan efektif.
--	--	--	--	---

Sumber: (Purwana, N., Djamal, C, E & Renaldi, F., 2016), (Setyaningsih, 2014), (Seisarrina, 2016), (Chircu, 2010).

## 2.2 Penjadwalan

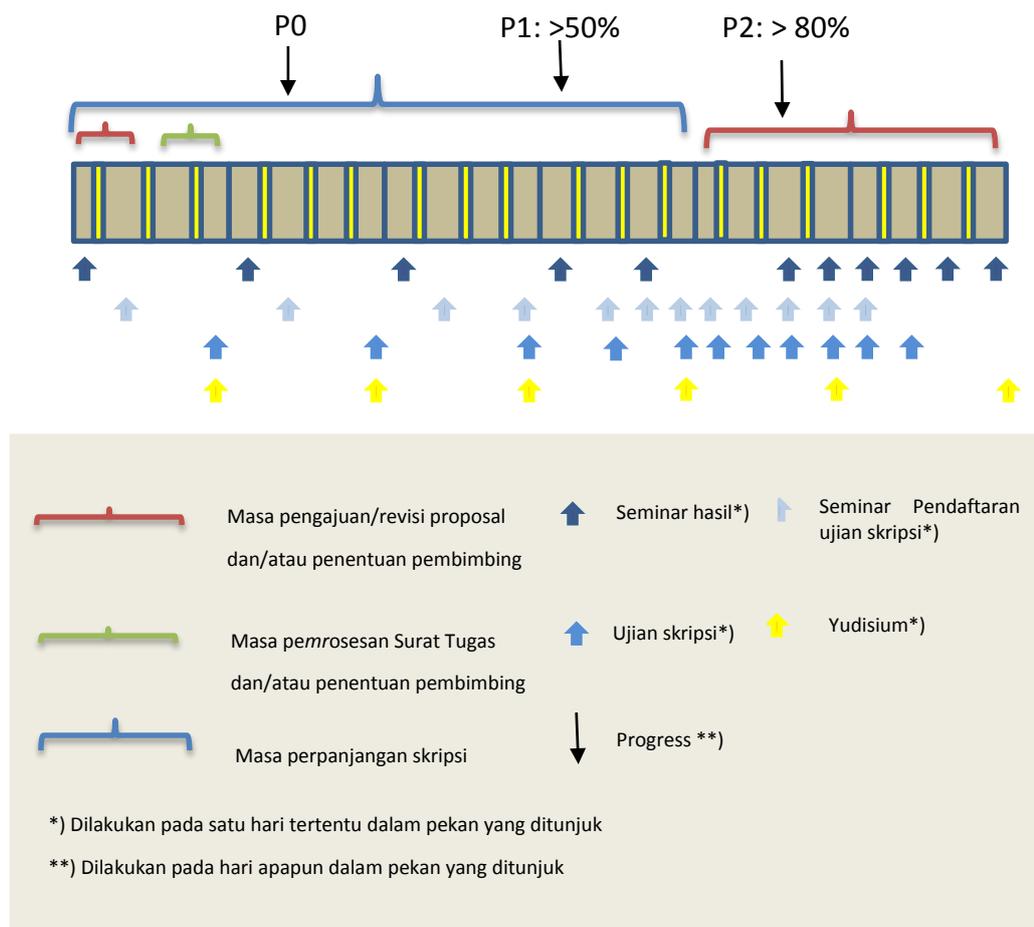
Penjadwalan yaitu acuan yang diperoleh berdasarkan penetapan keputusan, dalam hal ini terdapat rentang waktu didalam suatu pekerjaan. Untuk memaksimalkan tujuan yang ada rentang waktu harus digunakan semaksimal mungkin. Dengan hasil yang maksimal maka jadwal yang dihasilkan akan menjadi lebih baik dan tepat.

Permasalahan penjadwalan sering bermunculan dalam berbagai bentuk, yang paling sering yaitu dalam konteks pendidikan yang melibatkan alokasi sumber daya, dimana juga berhubungan dengan alokasi slot waktu yang harus sesuai dengan sejumlah permasalahan (Yabo & sani, 2016).

Menurut yabo & sani (2016) terdapat beberapa tipe *constraint* yang kemudian mereka membagi *constraint* tersebut kedalam 2 tipe *constraint* yaitu *hard constraint* dan *soft constraint*, berikut adalah penjelasan mengenai dua tipe *constraint* tersebut:

1. *Hard constraints* dimana dalam setiap penjadwalan itu tidak bisa dilanggar.
2. *Soft constraints* dimana kemungkinan dilanggar, tetapi pelanggaran tersebut bisa diminimalkan agar penjadwalan maksimal.

Di Falkultas Ilmu Komputer Universitas Brawijaya (FILKOM UB) pada setiap semester yang mana mahasiswa dapat mengambil skripsi. Dalam hal ini mahasiswa sudah memenuhi syarat pengambilan skripsi) maka akan diberlakukan beberapa penjadwalan yang telah ditentukan pertama yaitu pengajuan proposal yang mana sudah bisa dimulai semenjak dua bulan terakhir sebelum semester pengambilan skripsi disini mahasiswa sebelumnya bisa mencari calon dosen pembimbing pertama secara sendiri atau individu yang nantinya akan didiskusikan dengan calon dosen pembimbing tersebut atau akan di pilihkan dosen pembimbing pertama oleh akademik sesuai dengan judul yang mahasiswa inginkan. Mahasiswa diwajibkan mempersiapkan proposal skripsi yang terdiri dari bab 1, 2, dan 3 (sesuai format pada panduan skripsi). Mahasiswa yang mengambil skripsi harus memenuhi syarat yaitu lulus 120 sks dan mengambil skripsi pada Kartu Rencana Studi (KSR).



**Gambar 2.1 Keterangan Waktu Pelaksanaan Skripsi**

**Sumber: FILKOM UB (2015)**

Kedua, pemrosesan surat tugas dimana pembimbingan akan diproses pada pekan ketiga sampai keempat di bulan pertama semester pengambilan skripsi. Ketiga presentasi proposal (P0) adalah presentasi yang dilakukan pada salah satu hari pada pekan kedua di bulan kedua semester pengambilan skripsi untuk menyamakan konsep dan persepsi antara mahasiswa dan dosen pembimbing. Keempat yaitu penilaian kemajuan 1 (P1), proses ini merupakan penilaian yang dilakukan antara mahasiswa dan dosen pembimbing pada salah satu hari pada pekan keempat semester skripsi untuk melakukan evaluasi pencapaian target minimal 50% skripsi yang sudah dilakukan.

Kelima, penilaian kemajuan 2 (P2), penilaian ini sama seperti P1 tetapi dilakukan satu hari pada pekan kedua di bulan kelima semester skripsi untuk evaluasi target minimal 80% skripsi yang sudah dilakukan. Keenam yaitu perpanjangan skripsi pada tahap ini diberikan maksimum selama 4 bulan berikutnya. Ketujuh yaitu seminar hasil dimana mahasiswa yang selesai dalam pengerjaan skripsi diwajibkan melakukan seminar hasil. Proses ini dilakukan mahasiswa untuk mempresentasikan hasil yang diperoleh selama menyelesaikan skripsi di dalam forum yang dihadiri oleh para mahasiswa yang lain. Tujuannya

adalah sebagai latihan presentasi dari mahasiswa tersebut dimana nanti akan dihadiri oleh dosen fasilitator. Dosen fasilitator ini akan mengatur jalannya presentasi dan memberikan saran atau masukan terkait dengan teknik presentasi yang dilakukan oleh mahasiswa.

Kedelapan, pendaftaran ujian skripsi, pada proses ini mahasiswa wajib mendaftar ujian skripsi setelah melakukan seminar hasil. Kesembilan yaitu ujian skripsi, setelah mahasiswa melakukan pendaftaran ujian mahasiswa akan melakukan ujian skripsi dihadapan para dosen penguji dan dosen pembimbing pada hari tertentu. Kesepuluh yaitu yudisium, pada proses ini setelah mahasiswa melakukan ujian skripsi dan dinyatakan lulus (FILKOM UB,2015).

## **2.3 Algoritme Genetika**

Algoritme Genetika termasuk algoritme komputasi yang memanfaatkan teori evolusi. Dengan meniru teori evolusi, individu yang terdapat pada algoritme genetika akan mengalami suatu perubahan salah satunya karena faktor mutasi. Dengan perubahan tersebut diharapkan individu yang lolos adalah individu terbaik yang mampu bertahan dan beradaptasi dengan lingkungan barunya.

Salah satu definisi yang ada di algoritme genetika yaitu gen. Gen merupakan suatu nilai dapat berupa biner, float, integer ataupun karakter yang kemudian berkumpul menjadi satu kesatuan yang disebut kromosom. Sedangkan individu sendiri berarti mewakili salah satu solusi dari suatu permasalahan. Untuk dapat mengukur tingkat baiknya nilai dari suatu individu harus diukur dengan membandingkan nilai *fitness* pada setiap individu yang ada sesuai dengan proses algoritme genetika.

### **2.3.1 Struktur Algoritme Genetika**

Pada struktur algoritme genetika pembentukan kromosom baru dapat terjadi dimana nantinya akan terjadi penyilangan antara dua kromosom, mutasi dan seleksi. Berikut adalah struktur utama dari algoritme genetika.

#### **2.3.1.1 Inisialisasi Kromosom**

Hanya satu kromosom yang dapat mewakili satu solusi. Satu solusi ini bisa langsung di implementasi pada permasalahan genetika algoritme atau kita melakukan sebuah pengkodean. Pada pengkodean ini kita menentukan jumlah populasi (*popsiz*) terlebih dahulu. Nilai yang akan ditampung ke dalam populasi dinamakan *popsiz*, sedangkan kumpulan dari kromosom dinamakan populasi (Mahmudy, 2013). Misalkan terdapat *popsiz*=10 yang mana individunya dipilih secara acak maka akan dihasilkan populasi seperti berikut:

**Tabel 2.2 Contoh Inisialisasi Kromosom**

Induk	Kromosom		$f(x_1, x_2)$
	X1	X2	
Induk1	1,4898	2,0944	19,8206
Induk2	8,4917	2,5754	34,7058
Induk3	1,4054	6,3035	10,6707
Induk4	5,8114	5,07779	14,5624
Induk5	-1,8461	1,7097	11,5858
Induk6	4,0206	4,4355	24,7106
Induk7	-0,1634	2,974	19,653
Induk8	5,2742	0,7183	2,1813
Induk9	9,4374	6,6919	12,4694
Induk10	-4,5575	0,1679	28,4324

Sumber: (Mahmudy,2013)

### 2.3.1.2 Reproduksi

Pada proses adalah untuk memberikan generasi baru dari setiap individu pada populasi. Terdapat dua operator yaitu *crossover* dan *mutation*. Penentuan nilai *cr* (*crossover*) dan nilai *mr* (*mutation*) menjadi tahap yang penting pada proses ini (Mahmudy, 2013).

### 2.3.1.3 Crossover

Pada proses ini dua kromosom induk di pilih secara acak dan dilakukan penyilangan untuk menghasilkan anak (*offspring*) yang diharapkan kromosom ini akan lebih baik daripada kromosom induk sebelumnya. Pada penelitian ini peneliti menggunakan metode *extended intermediate crossover*. Misalkan induk1 dan induk2 adalah induk yang terpilih untuk melakukan proses *crossover*, maka untuk *C1* dan *C2* bisa dibangkitkan dengan rumus (Mahmudy,2013) dapat dilihat pada Persamaan 2.1:

$$\begin{aligned}
 C1 &= \text{induk1} + a(\text{induk2} - \text{induk1}) \\
 C2 &= \text{induk2} + a(\text{induk1} - \text{induk2})
 \end{aligned}
 \tag{2.1}$$

Keterangan :

- *C1* dan *C2* = *Child 1*, *Child 2*
- induk1 dan induk2= Induk yang terpilih untuk dijadikan *parent 1* dan *parent 2*
- *a*(bilangan *random* interval[-0,25;1,25])

Misal yang terpilih sebagai induk adalah induk4 dan induk9 pada Tabel 2.7,  $\alpha=[0.1104,1.2336]$  maka akan dihasilkan dua *offspring* (C1 dan C2) sebagai berikut:

$$C1: x1= 5,8114+0,1104(9,3474-5,8114) =6,2118$$

$$x2= 5,0779+1,2336(6,6919-5,0779) =7,0690$$

$$C2: x1= 9,4374+0,1104(5,8114-9,4374) =9,0370$$

$$x2= 6,6919+1,2336(5,0779-6,6919) =4,700$$

Jika ditentukan  $cr=0,4$  maka ada  $0,4*10=4$  anak yang dihasilkan dari proses *crossover*. Setiap *crossover* akan memiliki dua anak, berarti terdapat dua kali operasi *crossover* yang akan menghasilkan dua anak yaitu C3 dan C4.

### 2.3.1.4 Mutasi

Dengan adanya mutasi pembentukan individu baru dilakukan dengan merubah gen pada individu yang sama menghasilkan individu yang berbeda dari induknya. Selama proses seleksi terdapat beberapa gen yang hilang dari populasi untuk itu mutasi dilakukan agar menggantikan gen yang hilang tersebut. Pada proses ini kita memilih satu induk secara *random* yang dilakukan dengan penambahan dan pengurangan nilai gen yang terpilih dengan bilangan *random* yang kecil, sesuai dengan Persamaan 2.2. Misalkan domain dari variabel  $x_j$  adalah  $[\min_j, \max_j]$  dan *offspring* yang dihasilkan adalah  $C=[x'_1...x'_n]$ , yang berarti nilai dari gen *offspring* bisa dibandingkan dengan rumus gen pada hasil mutasi (Mahmudy, 2013):

$$x'_i = x_i + r(\max_i - \min_j) \quad (2.2)$$

Keterangan :

- $(\max_i - \min_j)$  = domain variabel  $ij$
- $X_i'$  = *offspring* yang dihasilkan
- $r$  = range (misalkan bilangan *random* interval  $[-0,1;0,1]$ )

Misal yang terpilih sebagai induk adalah induk2 pada Tabel 2.2, gen yang terpilih nomor 2 ( $x_2$ ) dan  $r=-0,0584$ . Maka akan dihasilkan *offspring* (C5) sebagai berikut:

$$C5: x1=8,491 \text{ (tetap)}$$

$$x2= 2,5754-0,0584 (7,3-0,0) = 2,1491$$

Nilai  $mr$  telah ditentukan misal  $mr = 0,2$  maka  $0,2*10 = 2$  *offspring* yang dihasilkan dari proses mutasi. *Offspring* selanjutnya akan dianggap sebagai C6. Hasil dari *offspring* ditunjukkan pada Tabel 2.3.

**Tabel 2.3 Contoh Hasil Crossover dan Mutasi**

Offspring	Kromosom		$f(x_1, x_2)$
	x1	x2	

C1	6,2118	7,0690	22,2048
C2	9,0370	4,7008	22,2313
C3	7,1636	0,00000	15,4774
C4	7,5479	7,3	9,3531
C5	8,4917	2,1494	31,0389
C6	-1,1238	1,7097	12,0177

Sumber: (Mahmudy,2013)

### 2.3.1.5 Evaluasi

Setelah kromosom dibuat dan dilakukan reproduksi, proses selanjutnya adalah evaluasi. Setiap individu yang ada dilakukan perhitungan nilai *fitness*. Jika semakin tinggi nilai *fitness* maka semakin baik untuk dijadikan calon solusi untuk kromosom tersebut (Mahmudy, 2015). Untuk perhitungan *fitness* menggunakan Persamaan 2.3 (Syafiq, Cholissodin, & Aryadita, 2017).

$$fitness = \sum \frac{k1}{a1+1} + \frac{k2}{a2+1} + \frac{k3}{a3+1} + \frac{k4}{b1+1} \quad (2.3)$$

Keterangan:

k1 = Nilai konstanta bobot pada *hard constraint* pertama

k2 = Nilai konstanta bobot pada *hard constraint* kedua

k3 = Nilai konstanta bobot pada *hard constraint* ketiga

k4 = Nilai konstanta bobot pada *soft constraint* pertama

a1 = Jumlah dari pelanggaran pada *hard constraint* pertama

a2 = Jumlah dari pelanggaran pada *hard constraint* kedua

a3 = Jumlah dari pelanggaran pada *hard constraint* ketiga

b1 = Jumlah dari pelanggaran pada *soft constraint* pertama

### 2.3.1.6 Seleksi

Pada proses ini dilakukan pemilihan individu yang akan di pertahankan untuk generasi selanjutnya dari himpunan populasi yang ada (Mahmudy, 2015). Metode seleksi yang sering digunakan yaitu *roulette wheel selection*, *binary tournament selection*, *elitism selection* dan *replacment selection* (Mahmudy, 2015).

#### 1. Seleksi *Elitism*

Metode seleksi yang digunakan peneliti adalah metode seleksi *elitism*. Metode seleksi *elitism* mengumpulkan semua individu dari induk dan anak kedalam satu tempat penampungan. Hal yang pertama dilakukan pada metode ini adalah individu-individu ditempatkan dalam penampungan sementara dan diurutkan berdasarkan nilai dari *fitness* terbesar. Kemudian akan diambil sesuai

dengan popsize yang telah ditentukan diawal dan akan diproses kegenerasi selanjutnya.

## 2. *Roulette Whell*

Metode ini menghitung nilai dari probabilitas seleksi (*prob*) pada setiap individu dimana individu tersebut memiliki nilai *fitness*. Nilai *prob* akan menghasilkan nilai probabilitas kumulatif (*probCum*) yang kemudian digunakan untuk proses seleksi.