

## PENJADWALAN JOB SHOP MENGGUNAKAN ARTIFICIAL FISH SWARM ALGORITHM (AFSA)

Dyah Pythaloka<sup>1</sup>, Agung Toto Wibowo<sup>2</sup>, Mahmud Dwi Suliyo<sup>3</sup>

<sup>1</sup>Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

---

### Abstrak

Salah satu masalah optimasi kombinatorial yang paling sulit dalam penelitian yang berkembang saat ini adalah penjadwalan job shop. Penjadwalan job shop yang juga memegang kunci profitabilitas perusahaan merupakan masalah krusial yang dihadapi oleh berbagai perusahaan manufaktur. Penjadwalan yang tersusun dengan baik berpotensi untuk mengurangi biaya operasi dan meningkatkan keuntungan. Artificial Fish Swarm Algorithm (AFSA) merupakan salah satu algoritma optimasi untuk menyelesaikan permasalahan kombinatorial. Oleh karena itu, pada Tugas Akhir ini, akan diimplementasikan AFSA pada kasus penjadwalan job shop untuk menghasilkan solusi jadwal yang optimal, yaitu jadwal dengan waktu total penyelesaian seluruh pekerjaan (makespan) yang minimum. Hasil penelitian memperlihatkan bahwa AFSA yang dirancang untuk optimasi permasalahan penjadwalan job shop mampu memberikan solusi dengan nilai efisiensi terbaik yang pernah dicapai adalah sebesar 75%. Angka tersebut dinilai masih belum memuaskan dilihat dari makespan yang dihasilkan. Namun, kemampuan AFSA dalam mencari solusi sudah cukup baik mengingat tingkat efisiensi tersebut dicapai dengan hanya membangkitkan 10000 artificial fish pada 100 generasi untuk ruang solusi sebesar  $3,72e+41$ .

Kata Kunci : artificial fish, Artificial Fish Swarm Algorithm, AFSA, job shop, makespan

---

### Abstract

One of the most difficult combinatorial optimization problems in the developed research at this time is job shop scheduling. Job shop scheduling which also holds the key to the company's profitability is a crucial problem faced by many manufacturing companies. Well-structured scheduling has the potential to reduce operating costs and increase profits. Artificial Fish Swarm Algorithm (AFSA) is one of optimization algorithms to solve combinatorial problems. Therefore, in this Final Project, AFSA will be implemented in the case of job shop scheduling to produce an optimal solution schedules, ie schedules with a total time of completion of the entire job (makespan) which is minimum. The results showed that the AFSA which is designed for job shop scheduling problem optimization is able to provide solutions with the best efficiency value ever achieved was 75%. This figure is still considered unsatisfactory, seen from the resulting makespan. However, the ability of AFSA in the search for solutions is quite good considering that level of efficiency is achieved by generating only 10000 artificial fishes at 100 generations in the  $3,72e+41$  solution spaces.

Keywords : artificial fish, Artificial Fish Swarm Algorithm, AFSA, job shop, makespan

---

## Daftar Tabel

Tabel 3- 1 <i>Job shop instance</i> ft06 (6 x 6).....	12
Tabel 3- 2 Contoh permasalahan penjadwalan <i>job shop</i> 3 x 3 .....	16
Tabel 4- 1 Tabel skenario observasi parameter AFSA.....	23
Tabel 4- 2 Tabel skenario pengujian parameter <i>FishNum</i> .....	23
Tabel 4- 3 Tabel skenario pengujian parameter <i>Visual</i> .....	24
Tabel 4- 4 Tabel skenario pengujian parameter $\delta$ .....	25
Tabel 4- 5 Tabel skenario pengujian parameter <i>Trynumber</i> .....	25
Tabel 4- 6 Tabel hasil pengujian observasi parameter AFSA skenario 1-3 .....	26
Tabel 4- 7 Tabel hasil pengujian observasi parameter AFSA skenario 4 - 6 .....	27
Tabel 4- 8 Tabel hasil pengujian observasi parameter AFSA skenario 7 - 9 .....	28
Tabel 4- 9 Tabel hasil pengujian observasi parameter AFSA skenario 10 - 12 .....	29
Tabel 4- 10 Tabel paket parameter terbaik hasil pengujian observasi parameter AFSA.....	29
Tabel 4- 11 Tabel hasil pengujian pengaruh parameter <i>FishNum</i> .....	31
Tabel 4- 12 Tabel hasil pengujian pengaruh parameter <i>Visual</i> .....	33
Tabel 4- 13 Tabel hasil pengujian pengaruh parameter $\delta$ .....	34
Tabel 4- 14 Tabel hasil pengujian pengaruh parameter <i>Trynumber</i> .....	36

# 1. Pendahuluan

## 1.1 Latar Belakang Masalah

Penjadwalan telah menjadi salah satu permasalahan yang penting di berbagai aspek kehidupan. Selain itu, tuntutan penjadwalan yang efektif dan efisien semakin dibutuhkan untuk meningkatkan hasil yang diinginkan [7]. Salah satu contoh bidang yang membutuhkan penjadwalan yang kompetitif adalah bidang perusahaan manufaktur dimana penjadwalan menjadi salah satu pemegang kunci profitabilitas perusahaan. Penjadwalan yang tersusun dengan baik berpotensi untuk mengurangi biaya operasi dan meningkatkan keuntungan. Perusahaan manufaktur beroperasi dengan berbagai sistem produksi, salah satunya adalah *job shop*. Sistem produksi *job shop* merupakan penjadwalan yang memiliki kendala pemrosesan tugas dimana setiap tugas harus melalui setiap mesin yang ada tepat satu kali. Dilihat dari segi waktu kedatangan *job*, penjadwalan *job shop* dapat dibagi menjadi penjadwalan *job shop* statis dan penjadwalan *job shop* dinamis. Pada penjadwalan *job shop* statis semua *job* diterima pada saat yang sama, sedangkan pada penjadwalan *job shop* dinamis setiap *job* memiliki waktu kedatangan masing-masing. Penjadwalan tersebut disebut sebagai penjadwalan *job shop deterministic* jika aspek waktu kedatangan telah diketahui sebelumnya. Namun sebaliknya jika aspek waktu kedatangan tersebut tidak diketahui, disebut sebagai penjadwalan *job shop non-deterministic* atau *stochastic* [11].

Masalah penjadwalan *job shop* termasuk ke dalam *NP-hard* (*Non-deterministic Polynomial-timehard*) *problem*, dimana merupakan salah satu dari masalah optimisasi kombinatorial yang paling sulit [9]. Beberapa karakteristik dari permasalahan penjadwalan *job shop* adalah terdapat sekelompok *job* yang dengan jumlah tertentu dimana setiap *job* terdiri dari serangkaian operasi serta terdapat sekelompok mesin dengan jumlah tertentu dimana setiap mesin dapat memproses lebih dari satu operasi dalam satu waktu. Selain itu, setiap operasi yang sedang diproses dalam mesin tidak dapat disela selama selang waktu yang telah ditentukan untuk menyelesaikan operasi. Tujuan dari masalah penjadwalan *job shop* ini adalah untuk menemukan sebuah jadwal yang optimal dimana alokasi operasi, mesin dan waktu penyelesaiannya minimal [16]. Selama riset sekitar tiga puluh tahun terakhir belum ditemukan algoritma yang cukup efektif dalam menemukan solusi optimal untuk JSP karena algoritma yang tersedia masih memiliki kompleksitas eksponensial. Karena itu, untuk kasus penjadwalan dengan mesin dan *job* yang cukup banyak masih sangat sulit untuk diselesaikan. Namun, hingga saat ini algoritma-algoritma optimasi untuk menghasilkan solusi yang mendekati solusi optimal terus dikembangkan. Di antaranya adalah dengan teknik *Priority Dispatch Rules*, *Bottleneck Based Heuristics*, *Local search methods* and

*Meta-heuristics* seperti *Simulated-Annealing* (SA) maupun *Genetic Algorithm* (GA) [12].

Berbagai algoritma yang termasuk ke dalam *swarm intelligence algorithm*, seperti *Particle Swarm Optimization* (PSO) dan *Ant Colony Optimization* (ACO) telah secara signifikan dibuktikan kemampuannya dalam menyelesaikan berbagai permasalahan *NP-complete*. Hal tersebut dikarenakan, *swarm intelligence algorithms* mempunyai beberapa karakteristik seperti *scalability*, *fault tolerance*, konsistensi, kecepatan yang tinggi, fleksibilitas, paralel, dsb. yang sesuai untuk menyelesaikan permasalahan *NP-complete* [14]. *Artificial Fish Swarm Algorithm* (AFSA) merupakan salah satu jenis algoritma optimisasi yang termasuk ke dalam *swarm intelligence algorithm*. AFSA merupakan algoritma *stochastic* berbasis populasi yang terinspirasi dari kecerdasan kolektif tingkah laku koloni ikan di alam. AFSA telah diaplikasikan ke dalam beberapa permasalahan seperti *fuzzy clustering*, *Resource Leveling*, optimisasi parameter *PID controller*, *spread spectrum code estimation*, *data mining*, optimisasi *DNA encoding sequence*, dsb [14].

Pada Tugas Akhir ini, *Artificial Fish Swarm Algorithm* (AFSA) dicoba diimplementasikan untuk menyelesaikan masalah penjadwalan *job shop*. Dengan cara kerja pencarian yang random dan paralel, AFSA mempunyai kemampuan dalam mengatasi ekstrim lokal untuk mencapai ekstrim global dengan kecepatan dalam konvergensi [5]. Selain itu, AFSA mempunyai karakteristik *non-sensitive initial artificial fish location*, fleksibilitas dan *fault tolerance* yang sesuai untuk menyelesaikan permasalahan *NP-hard* [14]. Oleh karena itu, AFSA diharapkan dapat memberikan hasil terbaik dalam penyelesaian masalah optimisasi penjadwalan *job shop* ini, sehingga menghasilkan penjadwalan yang optimum.

## 1.2 Perumusan Masalah

Permasalahan pada Tugas Akhir ini adalah:

1. Bagaimana mencari waktu penyelesaian *job* (*makespan*) yang minimal pada penjadwalan *job shop* dari kombinasi jadwal yang sangat banyak?
2. Bagaimana menerapkan *Artificial Fish Swarm Algorithm* (AFSA) pada masalah penjadwalan *job shop* yang ruang masalahnya sangat besar agar didapat solusi yang lebih optimal?

Dalam implementasi Tugas Akhir ini dibatasi oleh beberapa hal, sebagai berikut:

1. Data pengujian berupa data test standar untuk masalah penjadwalan *job shop* yang diambil dari website OR-Library dengan alamat <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/jobshop1.txt>
2. Tipe permasalahan penjadwalan *job shop* yang digunakan adalah penjadwalan *job shop* statis, dimana:

- a. Terdapat  $m$  mesin yang harus memproses  $n$  *job* secara tuntas.
  - b. Terdapat serangkaian operasi dari sebuah *job*. Tiap operasi yang ada sudah ditetapkan akan diproses di mesin tertentu.
  - c. Setiap operasi memiliki waktu pemrosesan yang telah ditetapkan sebelumnya.
3. Sebuah mesin hanya diperbolehkan memproses sebuah operasi sekali saja.
  4. Sangkaian operasi dalam sebuah *job* sudah memiliki urutan pemrosesan tertentu. Saat sebuah operasi sedang diproses dalam suatu mesin, maka pemrosesan tersebut tidak boleh dihentikan sebelum pemrosesan operasi itu benar-benar selesai. Dengan kata lain, tidak diijinkan terjadinya *overlap*.

### 1.3 Tujuan

Tujuan dari Tugas Akhir ini adalah mengimplementasikan *Artificial Fish Swarm Algorithm* (AFSA) pada permasalahan penjadwalan *job shop* untuk mendapatkan penjadwalan *job shop* yang optimum yang mempunyai waktu penyelesaian *job* (*makespan*) yang minimal.

### 1.4 Metodologi Penyelesaian Masalah

1. Studi literatur.  
Pencarian materi-materi dan referensi yang berkaitan dengan permasalahan yang dibahas, seperti materi tentang penjadwalan *job shop*, *Artificial Fish Swarm Algorithm* (AFSA) dan materi pendukung lainnya.
2. Analisis dan perancangan kebutuhan sistem.  
Merupakan tahap perancangan sistem yang dibuat, yakni sebuah perangkat lunak yang akan menerapkan *Artificial Fish Swarm Algorithm* (AFSA) pada permasalahan penjadwalan *job shop* untuk mendapatkan waktu penyelesaian *job* (*makespan*) yang minimal.
3. Implementasi sistem.  
Melakukan *coding* menggunakan *tools* MATLAB R2009a untuk membangun sistem sesuai dengan rancangan pada tahap sebelumnya.
4. Pengujian sistem.  
Pada tahap ini, dilakukan pengujian terhadap sistem yang telah dibangun. Hal yang diujikan ialah seperti yang telah dipaparkan pada tahap perancangan.
5. Analisis hasil pengujian.  
Dari tahap pengujian sistem yang dilakukan sebelumnya, dilakukan analisis terhadap hasil pengujian yang terdiri dari verifikasi hasil keluaran, perbandingan hasil keluaran serta analisis terhadap performansi sistem.

Deskripsi *pseudo code* dari proses tersebut adalah sebagai berikut [4]:

```

Artificial_fish_swarm()
{
  nf = N(Xi, Visual);
  if (nf ≠ 0 and  $\frac{nf}{FishNum} < \delta$ )
  then Xc = Center(nf = N(Xi, Visual));
  Yc = f(Xc);
  if Yc > Yi
  then Xi = Xc;
  else AF_move();
  end
  else AF_move();
  end
}

```

#### 3.3.4.4 Perancangan *Move Behaviour*

Jika nilai solusi optimal tidak meningkat setelah dilakukan langkah-langkah optimisasi di atas, *generate* status lain secara random. Jika fungsi obyektif status baru ini lebih baik dibandingkan dengan fungsi obyektif X<sub>i</sub>, adaptasi status baru tersebut sebagai *current status* AF ke-i, jika tidak eksekusi *behaviour* yang lain.

#### 3.3.5 Kriteria Pemberhentian

AFSA merupakan sebuah algoritma iteratif, yang secara bertahap konvergen menuju solusi optimum global. Untuk permasalahan dengan solusi yang diketahui, kriteria pemberhentian yang biasa digunakan adalah iterasi maksimum. Solusi optimal yang ditemukan dari setiap iterasi akan di-rekam yang kemudian akan dipilih yang terbaik setelah iterasi selesai.

## 4. Pengujian dan Analisis

### 4.1 Pengujian Sistem

Pada bab 4.1 akan disampaikan tujuan pengujian yang dilakukan dan bagaimana skenario pengujian untuk mendapatkan hasil yang akan dianalisis.

#### 4.1.1 Tujuan Pengujian

Tujuan dilakukannya proses pengujian antara lain:

1. Mencari kombinasi parameter *Artificial Fish Swarm Algorithm* (AFSA) yang menghasilkan solusi paling optimal dilihat dari *makespan* solusi.
2. Menganalisis pengaruh jumlah *artificial fish* yang digunakan terhadap solusi yang dihasilkan dilihat dari *makespan* solusi.
3. Menganalisis pengaruh besarnya *visual distance* yang dimiliki *artificial fish* terhadap solusi yang dihasilkan dilihat dari *makespan* solusi.
4. Menganalisis pengaruh besarnya *crowd factor* populasi terhadap solusi yang dihasilkan dilihat dari *makespan* solusi.
5. Menganalisis pengaruh banyaknya iterasi pada *prey behaviour* terhadap solusi yang dihasilkan dilihat dari *makespan* solusi.

#### 4.1.2 Skenario Pengujian

Untuk pengujian observasi parameter AFSA, dipilih 4 *instance* sebagai dataset yaitu:

1. *Instance* ft06 (6 *job*, 6 mesin), setiap *job* terdiri dari 6 operasi
2. *Instance* la01 (10 *job*, 5 mesin), setiap *job* terdiri dari 5 operasi
3. *Instance* la06 (15 *job*, 5 mesin), setiap *job* terdiri dari 5 operasi
4. *Instance* ft10 (10 *job*, 10 mesin), setiap *job* terdiri dari 10 operasi

Sedangkan untuk pengujian pengaruh masing-masing parameter jumlah *artificial fish*, *visual distance*, besar *crowd factor*, dan banyaknya iterasi pada *prey behaviour* akan dilakukan pada *instance* ft06 berdasarkan data terbaik yang didapatkan dari pengujian observasi parameter.

##### 4.1.2.1 Observasi Parameter *Artificial Fish Swarm Algorithm* (AFSA)

Pengujian yang pertama dilakukan adalah melakukan observasi untuk mendapatkan parameter yang paling optimal untuk dataset dilihat dari akurasi *makespan* solusi yang dihasilkan. Skenario yang digunakan merupakan kombinasi dari 4 parameter, dimana masing-masing skenario di-*running* sebanyak 10 kali.

Berikut ini adalah skenario observasi parameter AFSA yang digunakan:

Tabel 4- 1 Tabel skenario observasi parameter AFSA

Dataset	FishNum	Visual	$\delta$	Trynumber
ft06	50	2	0.1	10
	100	5	0.5	20
	150	10	0.9	30
la01	100	6	0.1	40
	200	8	0.5	50
	300	10	0.9	60
la06	250	5	0.1	70
	500	10	0.5	80
	750	15	0.9	90
ft10	500	10	0.1	100
	750	15	0.5	110
	1000	20	0.9	120

Tahapan kombinasi pengujian dari keempat parameter yang dilakukan untuk masing-masing skenario bagi setiap dataset akan dijelaskan lebih lanjut pada sub bab 4.2.1.

#### 4.1.2.2 Analisis Pengaruh Parameter Jumlah *Artificial Fish* (*FishNum*)

Di dalam AFSA, seekor *artificial fish* (AF) merepresentasikan sebuah solusi. Sehingga jika terdapat  $n$  buah AF, akan dihasilkan  $n$  buah solusi pada akhir iterasi. Sesuai dengan prinsip *global search*, AF akan bekerja bersama mencari solusi terbaik. Pada pengujian ini akan dianalisis bagaimana pengaruh jumlah AF terhadap performansi AFSA, dilihat dari nilai *makespan* yang didapatkan. Skenario pengujian dilakukan pada dataset ft06 dengan menggunakan variasi jumlah AF = 1000 ~ 10000 dengan jarak nilai sebesar 1000. Masing-masing skenario dilakukan *running* sebanyak 10 kali.

Tabel 4- 2 Tabel skenario pengujian parameter *FishNum*

Skenario	FishNum	Visual	$\delta$	Trynumber
1	1000	2	0.9	100
2	2000	2	0.9	100
3	3000	2	0.9	100
4	4000	2	0.9	100
5	5000	2	0.9	100
6	6000	2	0.9	100
7	7000	2	0.9	100
8	8000	2	0.9	100
9	9000	2	0.9	100
10	10000	2	0.9	100

#### 4.1.2.3 Analisis Pengaruh Parameter *Visual Distance (Visual)*

Parameter *visual* merepresentasikan besarnya *visual distance* (jarak penglihatan) yang dimiliki oleh *artificial fish* (AF) di dalam populasi. AF mencari solusi dalam populasi berdasarkan *visual* yang dimiliki dan kemudian bergerak secara random menuju target yang telah ditentukan. Pada pengujian ini akan dianalisis bagaimana pengaruh besarnya *visual distance artificial fish* terhadap performansi AFSA, dilihat dari nilai *makespan* yang didapatkan. Skenario pengujian dilakukan pada dataset ft06 dengan menggunakan variasi besar *visual distance* = 1 ~ 10. Masing-masing skenario dilakukan *running* sebanyak 10 kali.

Tabel 4- 3 Tabel skenario pengujian parameter *Visual*

Skenario	<i>FishNum</i>	<i>Visual</i>	$\delta$	<i>Trynumber</i>
1	50	1	0.6	10
2	50	2	0.6	10
3	50	3	0.6	10
4	50	4	0.6	10
5	50	5	0.6	10
6	50	6	0.6	10
7	50	7	0.6	10
8	50	8	0.6	10
9	50	9	0.6	10
10	50	10	0.6	10

#### 4.1.2.4 Analisis Pengaruh Parameter *Crowd Factor ( $\delta$ )*

Parameter *crowd factor* ( $\delta$ ) merepresentasikan tingkat keramaian pada lingkungan sekitar dari *artificial fish* (AF). Tingkat keramaian pada lingkungan sekitar AF akan menentukan tingkat kebebasan dari AF tersebut dalam bergerak menuju solusi target. Besarnya batas *crowd factor* harus berada pada rentang  $0 < \delta < 1$ , karena  $\delta=0$  berarti tingkat keramaian = 0 atau tidak ada AF lain di sekitar sedangkan  $\delta=1$  berarti tingkat keramaian = 1 atau lingkungan terlalu ramai sehingga AF tidak dapat bebas bergerak. Pada pengujian ini akan dianalisis bagaimana pengaruh besarnya batas tingkat keramaian dalam populasi AF terhadap performansi AFSA, dilihat dari nilai *makespan* yang didapatkan. Skenario pengujian dilakukan pada dataset ft06 dengan menggunakan variasi besar *crowd factor* = 0.1 ~ 0.9. Masing-masing skenario dilakukan *running* sebanyak 10 kali.

Tabel 4- 4 Tabel skenario pengujian parameter  $\delta$

Skenario	FishNum	Visual	$\delta$	Trynumber
1	50	5	0.1	10
2	50	5	0.2	10
3	50	5	0.3	10
4	50	5	0.4	10
5	50	5	0.5	10
6	50	5	0.6	10
7	50	5	0.7	10
8	50	5	0.8	10
9	50	5	0.9	10

#### 4.1.2.5 Analisis Pengaruh Banyaknya Iterasi pada Prey Behaviour (Trynumber)

Parameter *trynumber* merepresentasikan banyaknya pencarian posisi yang lebih baik yang dilakukan oleh *artificial fish* (AF) saat melakukan *prey behaviour*. Percobaan pencarian dilakukan secara random sebanyak *trynumber* di lingkungan sekitar AF berdasarkan besarnya *visual distance* yang dimiliki oleh AF tersebut. Pada pengujian ini akan dianalisis bagaimana pengaruh banyaknya percobaan pencarian yang dilakukan oleh AF saat melakukan *prey behaviour* terhadap performansi AFSA, dilihat dari nilai *makespan* yang didapatkan. Skenario pengujian dilakukan pada dataset ft06 dengan menggunakan variasi besar *trynumber* = 100 ~ 1000 dengan jarak nilai sebesar 100. Masing-masing skenario dilakukan *running* sebanyak 10 kali.

Tabel 4- 5 Tabel skenario pengujian parameter Trynumber

Skenario	FishNum	Visual	$\delta$	Trynumber
1	1000	2	0.9	100
2	1000	2	0.9	200
3	1000	2	0.9	300
4	1000	2	0.9	400
5	1000	2	0.9	500
6	1000	2	0.9	600
7	1000	2	0.9	700
8	1000	2	0.9	800
9	1000	2	0.9	900
10	1000	2	0.9	1000

## 4.2 Hasil Pengujian dan Analisis

### 4.2.1 Analisis Hasil Pengujian Observasi Parameter *Artificial Fish Swarm Algorithm* (AFSA)

Pengujian observasi parameter AFSA dilakukan dengan mengkombinasikan parameter jumlah *artificial fish* (*FishNum*), besar *visual distance* (*Visual*), besar *crowd factor* ( $\delta$ ), dan banyaknya iterasi pada *prey behaviour* (*Trynumber*) sehingga dihasilkan paket parameter yang memberikan hasil paling optimal untuk masing-masing dataset. Kombinasi dari 4 parameter tersebut dilakukan secara bertahap dimulai dari *FishNum* hingga *Trynumber*, dengan menyimpan nilai parameter terbaik di tiap tahap untuk tahap berikutnya. Kombinasi tersebut kemudian menghasilkan 12 skenario untuk masing-masing dataset.

Skenario 1 - 3 dilakukan dengan menguji variasi nilai parameter *FishNum* yang akan diujikan bagi masing-masing dataset dengan menggunakan nilai parameter *Visual*,  $\delta$  dan *Trynumber* yang tetap. Nilai parameter *Visual* dan *Trynumber* yang digunakan merupakan nilai parameter *Visual* dan *Trynumber* terkecil dari variasi nilai parameter yang akan diujikan untuk masing-masing dataset. Sedangkan nilai parameter  $\delta$  yang digunakan merupakan nilai parameter  $\delta$  terbesar dari variasi nilai parameter yang akan diujikan untuk masing-masing dataset. Hasil pengujian observasi parameter AFSA untuk skenario 1 - 3 dapat dilihat pada tabel 4-6.

Tabel 4- 6 Tabel hasil pengujian observasi parameter AFSA skenario 1-3

Dataset	Best Known Solution	Skenario	<i>FishNum</i>	<i>Visual</i>	$\delta$	<i>Trynumber</i>	Makespan Terbaik	Efisiensi (%)
ft06	55	1	50	2	0.9	10	81	67,90
		2	100	2	0.9	10	82	67,07
		3	150	2	0.9	10	75	73,33
la01	666	1	100	6	0.9	40	946	70,40
		2	200	6	0.9	40	960	69,38
		3	300	6	0.9	40	936	71,15
la06	926	1	250	5	0.9	70	1658	55,85
		2	500	5	0.9	70	1601	57,84
		3	750	5	0.9	70	1525	60,72
ft10	930	1	500	10	0.9	100	2247	41,39
		2	750	10	0.9	100	2264	41,08
		3	1000	10	0.9	100	2044	45,50

Dari tabel hasil pengujian observasi parameter AFSA untuk skenario 1 - 3 bagi masing-masing dataset pada tabel 4 - 6 diatas terlihat bahwa, untuk semua dataset, efisiensi sistem terbaik tercapai pada saat nilai parameter *FishNum* terbesar dari variasi nilai parameter *FishNum* yang diujikan. Nilai parameter *FishNum* dari skenario yang menghasilkan nilai efisiensi sistem yang terbaik dari

skenario 1 - 3 bagi masing-masing dataset ini kemudian disimpan untuk pengujian bagi skenario-skenario berikutnya.

Skenario 4 - 6 dilakukan dengan menguji variasi nilai parameter *Visual* yang akan diujikan bagi masing-masing dataset dengan menggunakan nilai parameter *FishNum*,  $\delta$  dan *Trynumber* yang tetap. Nilai parameter *FishNum* yang digunakan merupakan nilai parameter *FishNum* yang telah menghasilkan nilai efisiensi sistem terbaik yang disimpan dari hasil pengujian skenario 1 - 3 sebelumnya. Nilai parameter *Trynumber* yang digunakan merupakan nilai parameter *Trynumber* terkecil dari variasi nilai parameter yang akan diujikan untuk masing-masing dataset, sedangkan nilai parameter  $\delta$  yang digunakan merupakan nilai parameter  $\delta$  terbesar dari variasi nilai parameter yang akan diujikan untuk masing-masing dataset. Hasil pengujian observasi parameter AFSA untuk skenario 4 - 6 dapat dilihat pada tabel 4-7.

Tabel 4- 7 Tabel hasil pengujian observasi parameter AFSA skenario 4 - 6

Dataset	Best Known Solution	Skenario	FishNum	Visual	$\delta$	Trynumber	Makespan Terbaik	Efisiensi (%)
ft06	55	4	150	2	0.9	10	75	73,33
		5	150	5	0.9	10	97	56,70
		6	150	10	0.9	10	89	61,80
la01	666	4	300	6	0.9	40	936	71,15
		5	300	8	0.9	40	1002	66,47
		6	300	10	0.9	40	955	69,74
la06	926	4	750	5	0.9	70	1525	60,72
		5	750	10	0.9	70	1698	54,53
		6	750	15	0.9	70	1614	57,37
ft10	930	4	1000	10	0.9	100	2044	45,50
		5	1000	15	0.9	100	2108	44,12
		6	1000	20	0.9	100	2243	41,46

Dari tabel hasil pengujian observasi parameter AFSA untuk skenario 4 - 6 bagi masing-masing dataset pada tabel 4-7 diatas terlihat bahwa, untuk semua dataset, efisiensi sistem terbaik tercapai pada saat nilai parameter *Visual* terkecil dari variasi nilai parameter *Visual* yang diujikan. Nilai parameter *Visual* dari skenario yang menghasilkan nilai efisiensi sistem yang terbaik dari skenario 4 - 6 bagi masing-masing dataset ini kemudian disimpan untuk pengujian bagi skenario-skenario berikutnya.

Skenario 7 - 9 dilakukan dengan menguji variasi nilai parameter  $\delta$  yang akan diujikan bagi masing-masing dataset dengan menggunakan nilai parameter *FishNum*, *Visual* dan *Trynumber* yang tetap. Nilai parameter *FishNum* yang digunakan merupakan nilai parameter *FishNum* yang telah menghasilkan nilai efisiensi sistem terbaik yang disimpan dari hasil pengujian skenario 1 - 3 sebelumnya. Nilai parameter *Visual* yang digunakan merupakan nilai parameter *Visual* yang telah menghasilkan nilai efisiensi sistem terbaik yang disimpan dari

hasil pengujian skenario 4 - 6 sebelumnya. Nilai parameter *Trynumber* yang digunakan merupakan nilai parameter *Trynumber* terkecil dari variasi nilai parameter yang akan diujikan untuk masing-masing dataset. Hasil pengujian observasi parameter AFSA untuk skenario 7 - 9 dapat dilihat pada tabel 4-8.

Tabel 4- 8 Tabel hasil pengujian observasi parameter AFSA skenario 7 - 9

Dataset	Best Known Solution	Skenario	FishNum	Visual	$\delta$	Trynumber	Makespan Terbaik	Efisiensi (%)
ft06	55	7	150	2	0.1	10	147	37,41
		8	150	2	0.5	10	103	53,40
		9	150	2	0.9	10	75	73,33
la01	666	7	300	6	0.1	40	1219	54,63
		8	300	6	0.5	40	1251	53,24
		9	300	6	0.9	40	936	71,15
la06	926	7	750	5	0.1	70	1912	48,43
		8	750	5	0.5	70	1912	48,43
		9	750	5	0.9	70	1525	60,72
ft10	930	7	1000	10	0.1	100	2218	41,93
		8	1000	10	0.5	100	2534	36,70
		9	1000	10	0.9	100	2044	45,50

Dari tabel hasil pengujian observasi parameter AFSA untuk skenario 7 - 9 bagi masing-masing dataset pada tabel 4-8 diatas terlihat bahwa, untuk semua dataset, efisiensi sistem terbaik tercapai pada saat nilai parameter  $\delta$  terbesar dari variasi nilai parameter  $\delta$  yang diujikan. Nilai parameter  $\delta$  dari skenario yang menghasilkan nilai efisiensi sistem yang terbaik dari skenario 7 - 9 bagi masing-masing dataset ini kemudian disimpan untuk pengujian bagi skenario-skenario berikutnya.

Skenario 10 - 12 dilakukan dengan menguji variasi nilai parameter *Trynumber* yang akan diujikan bagi masing-masing dataset dengan menggunakan nilai parameter *FishNum*, *Visual* dan  $\delta$  yang tetap. Nilai parameter *FishNum* yang digunakan merupakan nilai parameter *FishNum* yang telah menghasilkan nilai efisiensi sistem terbaik yang disimpan dari hasil pengujian skenario 1 - 3 sebelumnya. Nilai parameter *Visual* yang digunakan merupakan nilai parameter *Visual* yang telah menghasilkan nilai efisiensi sistem terbaik yang disimpan dari hasil pengujian skenario 4 - 6 sebelumnya. Nilai parameter  $\delta$  yang digunakan merupakan nilai parameter  $\delta$  yang telah menghasilkan nilai efisiensi sistem terbaik yang disimpan dari hasil pengujian skenario 7 - 9 sebelumnya. Hasil pengujian observasi parameter AFSA untuk skenario 10 - 12 dapat dilihat pada tabel 4-9.

Tabel 4- 9 Tabel hasil pengujian observasi parameter AFSA skenario 10 - 12

Dataset	Best Known Solution	Skenario	FishNum	Visual	$\delta$	Trynumber	Makespan Terbaik	Efisiensi (%)
ft06	55	10	150	2	0.9	10	75	73,33
		11	150	2	0.9	20	85	64,71
		12	150	2	0.9	30	81	67,90
la01	666	10	300	6	0.9	40	936	71,15
		11	300	6	0.9	50	1071	62,18
		12	300	6	0.9	60	936	71,15
la06	926	10	750	5	0.9	70	1525	60,72
		11	750	5	0.9	80	1457	63,56
		12	750	5	0.9	90	1634	56,67
ft10	930	10	1000	10	0.9	100	2044	45,50
		11	1000	10	0.9	110	2338	39,78
		12	1000	10	0.9	120	2167	42,92

Dari tabel hasil pengujian observasi parameter AFSA untuk skenario 10 - 12 bagi masing-masing dataset pada tabel 4-9 diatas terlihat bahwa nilai efisiensi sistem terbaik tercapai pada saat nilai parameter *Trynumber* = 10 untuk dataset ft06, *Trynumber* = 40 untuk dataset la01, *Trynumber* = 80 untuk dataset la06, dan *Trynumber* = 100 untuk dataset ft10. Nilai efisiensi sistem yang didapat pada dataset la01 untuk skenario 10 dan 12 adalah sama namun karena nilai efisiensi sistem rata-rata yang dihasilkan pada saat pengujian skenario 10 (64,50%) lebih besar dari pada nilai efisiensi sistem rata-rata yang dihasilkan pada saat pengujian skenario 12 (60,95%), diambil skenario 10 sebagai skenario yang menghasilkan nilai efisiensi sistem terbaik bagi dataset la01 dengan nilai parameter *Trynumber* = 40. Nilai parameter *Trynumber* dari skenario yang menghasilkan nilai efisiensi sistem yang terbaik dari skenario 10 - 12 bagi masing-masing dataset ini kemudian disimpan bersama nilai parameter *FishNum*, *Visual*, dan  $\delta$  dari skenario pengujian 1 – 9 sebelumnya.

Setelah dilakukan pengujian dan observasi dari 12 skenario untuk masing-masing dataset didapatkan paket parameter AFSA terbaik bagi masing-masing dataset tersebut seperti yang terlihat pada tabel 4-10.

Tabel 4- 10 Tabel paket parameter terbaik hasil pengujian observasi parameter AFSA

Dataset	Best Known Solution	FishNum	Visual	$\delta$	Trynumber	Makespan Terbaik	Efisiensi (%)
ft06	55	150	2	0.9	10	75	73,33
la01	666	300	6	0.9	40	936	71,15
la06	926	750	5	0.9	80	1457	63,56
ft10	930	1000	10	0.9	100	2044	45,50

Hasil keluaran sistem optimisasi penjadwalan *job shop* berupa *makespan* yaitu total waktu yang dibutuhkan untuk menyelesaikan semua pekerjaan. Hasil

tersebut kemudian dibandingkan dengan solusi terbaik yang diketahui (*best known solution*) untuk setiap dataset sehingga dihasilkan nilai efisiensi sistem dengan rumus penghitungan seperti yang telah dijelaskan pada (2.9).

Berdasarkan hasil pengujian pada tabel 4-10 di atas dapat dilihat bahwa nilai efisiensi sistem terbaik yang didapat untuk masing-masing dataset adalah  $ft06 = 73,33\%$ ,  $la01 = 71,15\%$ ,  $la06 = 48,71\%$ , dan  $ft10 = 45,50\%$ . Sedangkan jika melihat dari seluruh percobaan pengujian untuk 12 skenario bagi masing-masing dataset, nilai efisiensi sistem rata-rata yang didapat adalah  $ft06 = 63,44\%$ ,  $la01=64,50\%$ ,  $la06=54,82\%$ , dan  $ft10 = 39,72\%$ .

Ruang solusi dalam hal ini adalah populasi *artificial fish* (AF) yang kemungkinan dapat terbentuk, seperti yang telah dijelaskan pada sub bab 3.3.1 dan 3.3.2, untuk masing-masing dataset terbilang cukup besar. Contohnya untuk dataset  $ft06$  yang memiliki 6 *job* dan 6 mesin, dengan masing-masing *job* memiliki 6 operasi, kemungkinan ruang solusi yang dapat terbentuk adalah  $(6 \times 6)!$  atau  $36! = 3,7199332678990121746799944815084e+41$ . Nilai ini sangat besar bila dibandingkan dengan ruang solusi pengujian yang relatif kecil. Dari ruang solusi pengujian yang relatif kecil dibandingkan dengan ruang solusi permasalahan yang sangat besar, didapatkan nilai *makespan* terbaik sebesar 75 dengan efisiensi sistem 73,33%.

Selain itu perlu diingat juga bahwa pembentukan populasi awal bersifat random sehingga tidak ada kepastian bahwa dalam ruang solusi pengujian yang terbentuk mengandung solusi jadwal teroptimal dengan *makespan* sebesar *best known solution*.

Namun, dari populasi awal yang terbentuk secara random tersebut memiliki *makespan* terbaik yang pernah ada dalam populasi. Perbandingan keluaran final *makespan* dengan *makespan* terbaik yang pernah ada dalam populasi tersebut untuk masing-masing dataset mencapai hasil efisiensi yang berkisar antara 75% - 100%. Dari nilai ini dapat dilihat bahwa walaupun sistem belum dapat menghasilkan solusi jadwal optimal dengan *makespan* sebesar *best known solution*, kinerja optimisasi sistem berbasis AFSA yang dibangun telah memiliki performansi yang cukup baik.

Permasalahan yang paling umum terjadi pada *Evolutionary Algorithms* adalah konvergensi prematur, dimana individu-individu dalam populasi konvergen pada suatu solusi optimum lokal [13]. Pada awalnya, individu-individu tersebut tersebar secara acak dan memiliki nilai obyektif yang sangat jauh dari nilai optimum (maksimum global). Misal, dalam satu area pencarian lokal ada satu individu A yang memiliki nilai obyektif paling tinggi dan mendekati nilai optimum lokal. Karena nilai obyektif individu A lebih tinggi dari nilai obyektif individu-individu di sekitarnya, individu A tersebut akan sering terpilih sebagai individu terbaik dan akan mempengaruhi individu-individu lain di sekitarnya. Pada suatu generasi tertentu, individu A tersebut dan individu-individu lain di sekitar yang telah terpengaruh akan mendominasi populasi [13]. Solusi optimal

yang tercapai akan terjebak pada solusi optimum lokal padahal terdapat solusi optimum global yang belum tereksplorasi yang nilainya lebih jauh lebih baik.

Pengujian seluruh skenario untuk observasi parameter AFSA di atas menggunakan nilai maksimum generasi yang sama sebagai kriteria pemberhentian. Namun, tidak semua percobaan dari setiap skenario yang diujikan mencapai nilai maksimum generasi yang telah ditetapkan. Proses pencarian untuk beberapa skenario terhenti sebelum mencapai maksimum generasi. *Makespan* keluaran yang dihasilkan tidak dapat mencapai nilai *makespan* terbaik yang pernah ada dalam populasi. Hal ini berarti bahwa terdapat indikasi terjadinya konvergensi prematur seperti yang telah dijelaskan pada paragraf sebelumnya. Pencarian *makespan* terjebak di dalam nilai maksimum lokal, padahal terdapat nilai maksimum global yang lebih baik yang seharusnya dapat terjadi. Hal inilah yang juga mempengaruhi nilai efisiensi sistem yang masih terbilang cukup rendah.

Bagaimana pengaruh besar masing-masing parameter AFSA terhadap performansi sistem akan dibahas pada sub bab berikutnya.

#### 4.2.2 Analisis Hasil Pengujian Pengaruh Parameter Jumlah *Artificial Fish* (*FishNum*)

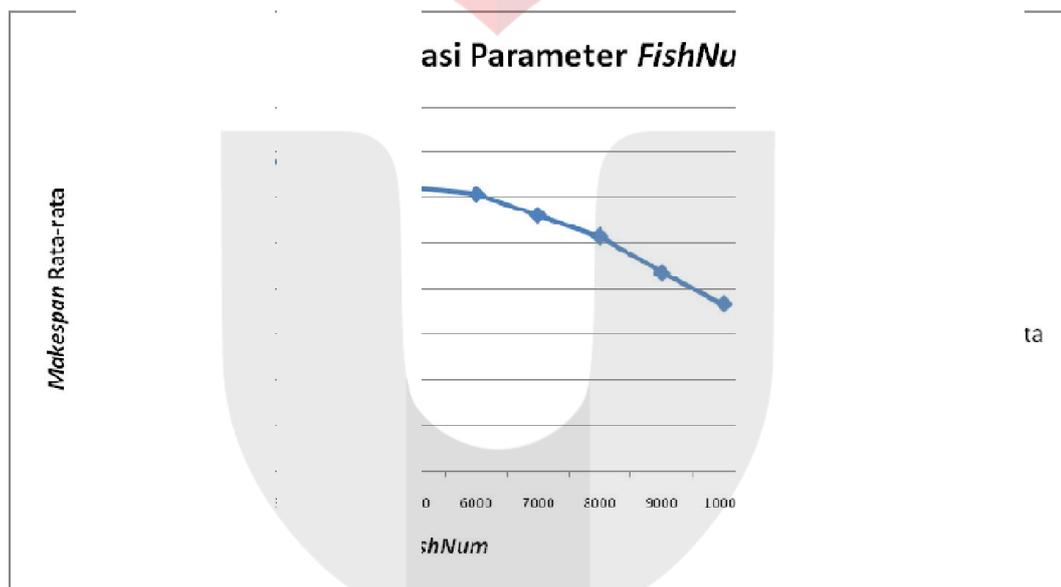
Pengujian pengaruh parameter jumlah *artificial fish* (*FishNum*) dilakukan pada dataset ft06 dengan menggunakan nilai  $Visual = 2$ ,  $\delta = 0.9$  dan  $Trynumber = 100$  yang sama untuk setiap jumlah *artificial fish* (*FishNum*) yang digunakan. Untuk hasil pengujiannya disajikan dalam tabel 4-11.

Tabel 4- 11 Tabel hasil pengujian pengaruh parameter *FishNum*

Skenario	<i>FishNum</i>	<i>Makespan</i> Rata-rata	Efisiensi (%)
1	1000	90,00	61,11
2	2000	89,00	61,80
3	3000	88,67	62,03
4	4000	87,67	62,74
5	5000	86,00	63,95
6	6000	85,33	64,45
7	7000	83,00	66,27
8	8000	80,67	68,18
9	9000	76,67	71,74
10	10000	73,33	75,00

Dari tabel hasil pengujian parameter jumlah *artificial fish* (AF) dapat dilihat dari *makespan* rata-rata setiap skenario bahwa semakin banyak jumlah AF yang digunakan, semakin tinggi nilai efisiensi *makespan* yang didapat. Jumlah AF berpengaruh terhadap performansi AFSA, hal ini dikarenakan semakin banyaknya

jumlah AF berarti semakin besar ukuran populasi dan semakin banyak pula jumlah kemungkinan solusi yang ada karena AF merepresentasikan solusi dari permasalahan yang akan diselesaikan. Selain itu, dengan semakin banyak jumlah AF yang digunakan, kerja pencarian juga menjadi semakin cepat. Nilai maksimum generasi yang digunakan dalam pengujian adalah 100 generasi untuk masing-masing skenario. Kerja pencarian yang dilakukan oleh 10000 AF dalam 100 generasi tentunya lebih efisien dibandingkan dengan kerja pencarian 1000 AF dalam 100 generasi. Peningkatan jumlah AF dapat membantu AFSA bebas dari nilai ekstrim lokal. Jumlah AF yang kecil terkadang dapat membuat AFSA tidak dapat mencapai konvergensi walaupun telah menyelesaikan seluruh iterasi yang ditentukan.



Gambar 4- 1 Grafik hasil pengujian pengaruh parameter *FishNum*

#### 4.2.3 Analisis Hasil Pengujian Pengaruh Parameter *Visual Distance (Visual)*

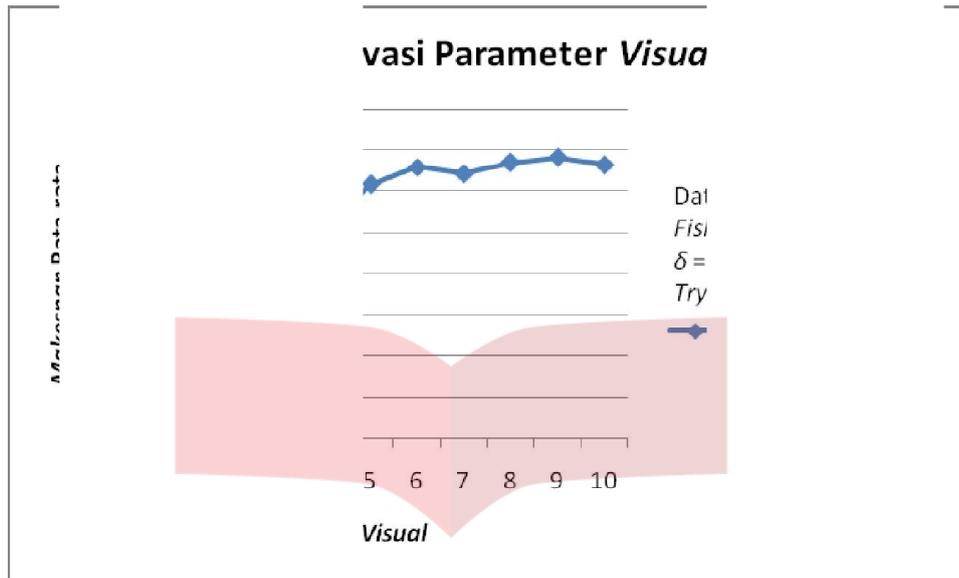
Pengujian pengaruh parameter *visual distance (Visual)* dilakukan pada dataset ft06 dengan menggunakan nilai *FishNum* = 50,  $\delta$  = 0.6 dan *Trynumber* = 10 yang sama untuk setiap nilai *visual distance (Visual)* yang digunakan. Untuk hasil pengujiannya disajikan dalam tabel 4-12.

Tabel 4- 12 Tabel hasil pengujian pengaruh parameter *Visual*

Skenario	<i>Visual</i>	<i>Makespan</i> Rata-rata	Efisiensi (%)
1	1	---	---
2	2	97	56,70
3	3	110,1	49,95
4	4	106,7	51,55
5	5	116,7	47,13
6	6	120,9	45,49
7	7	119,4	46,06
8	8	122	45,08
9	9	123,2	44,64
10	10	121,5	45,27

Dari tabel hasil pengujian di atas dapat dilihat bahwa untuk nilai *visual* = 1 yang diujicobakan pada skenario membuat AFSA tidak dapat mencapai konvergensi. Besarnya nilai *visual* harus sesuai dengan ukuran populasi dari *fish swarm*. Nilai *visual* sangat penting karena menggambarkan hubungan AF dengan AF lain di lingkungan sekitarnya pada populasi. Dengan menggunakan nilai *visual* yang lebih kecil atau tidak sesuai dengan ukuran dari populasi, akan membuat AFSA terjebak dalam nilai ekstrim lokal sehingga tidak akan dapat mencapai konvergensi ke solusi optimum global. Sedangkan nilai *visual* yang terlalu besar akan meningkatkan area pencarian dan mengurangi kecepatan konvergensi algoritma. Oleh karena itu, sangat penting untuk memilih besar nilai *visual* yang sesuai. Dengan melihat hasil nilai *makespan* rata-rata dari tabel hasil pengujian parameter *visual* di atas dapat dilihat bahwa nilai efisiensi rata-rata terbaik didapat ketika nilai *visual* = 2. Hal ini berarti besar nilai *visual distance* (*visual*) yang cocok digunakan untuk skenario di atas adalah sebesar 2.





Gambar 4- 2 Grafik hasil pengujian pengaruh parameter *Visual*

#### 4.2.4 Analisis Hasil Pengujian Pengaruh Parameter *Crowd Factor* ( $\delta$ )

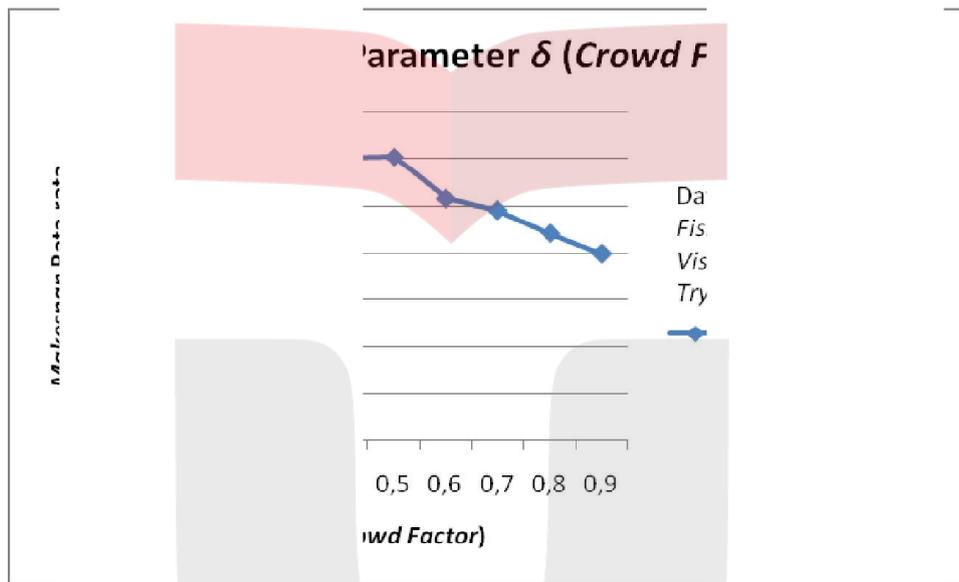
Pengujian pengaruh parameter *crowd factor* ( $\delta$ ) dilakukan pada dataset ft06 dengan menggunakan nilai *FishNum* = 50, *Visual* = 5 dan *Trynumber* = 10 yang sama untuk setiap besar nilai *crowd factor* ( $\delta$ ) yang digunakan. Untuk hasil pengujiannya disajikan dalam tabel 4-13.

Tabel 4- 13 Tabel hasil pengujian pengaruh parameter  $\delta$

Skenario	$\delta$	Makespan Rata-rata	Efisiensi (%)
1	0.1	115,9	47,45
2	0.2	117,9	46,65
3	0.3	114,8	47,91
4	0.4	115,1	47,78
5	0.5	115,4	47,66
6	0.6	106,7	51,55
7	0.7	104,1	52,83
8	0.8	99,3	55,39
9	0.9	94,9	57,96

Seperti halnya nilai parameter *visual*, nilai parameter *crowd factor* ( $\delta$ ) juga harus sesuai dengan ukuran populasi dari *fish swarm*. Semakin besar ukuran populasi dari *fish swarm*, akan semakin besar pula pengaruh besarnya nilai  $\delta$  yang digunakan. Ukuran populasi yang besar dengan nilai  $\delta$  yang besar akan mengarahkan kepada solusi optimal yang lebih kecil pada generasi pertama. Nilai  $\delta$  yang rasional dapat mencegah AFSA terjebak ke dalam nilai ekstrim lokal.

Selain itu juga perlu diperhatikan besar nilai *visual* yang digunakan karena kedua parameter ini bekerja secara berdampingan dalam menentukan posisi *artificial fish* (AF) dan pergerakannya di dalam populasi. Dengan melihat *makespan* rata-rata dari tabel hasil pengujian parameter  $\delta$  di atas dapat dilihat bahwa efisiensi rata-rata terbaik didapat ketika nilai  $\delta = 0.9$ . Hal ini berarti besar nilai *crowd factor* ( $\delta$ ) yang cocok digunakan untuk skenario di atas adalah sebesar 0.9.



Gambar 4- 3 Grafik hasil pengujian pengaruh parameter  $\delta$

#### 4.2.5 Analisis Hasil Pengujian Pengaruh Banyaknya Iterasi pada *Prey Behaviour* (Trynumber)

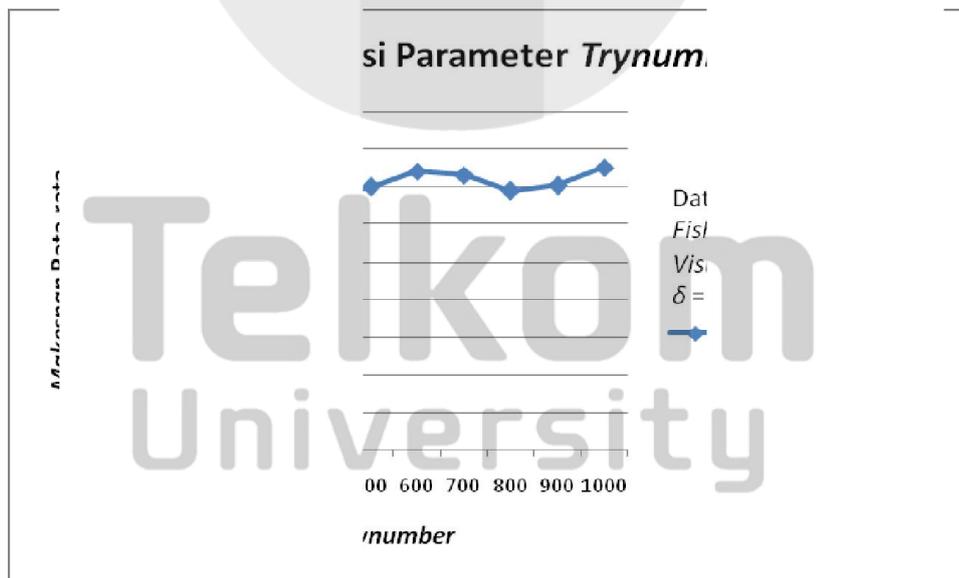
Pengujian pengaruh parameter banyaknya iterasi pada *prey behaviour* (Trynumber) dilakukan pada dataset ft06 dengan menggunakan nilai *FishNum* = 1000, *Visual* = 2 dan  $\delta = 0.9$  yang sama untuk setiap besar nilai iterasi *prey behaviour* (Trynumber) yang digunakan. Untuk hasil pengujiannya disajikan dalam tabel 4-14.



Tabel 4- 14 Tabel hasil pengujian pengaruh parameter *Trynumber*

Skenario	<i>Trynumber</i>	<i>Makespan</i> Rata-rata	Efisiensi (%)
1	100	95,1	57,83
2	200	90,7	60,64
3	300	89,5	61,45
4	400	89,1	61,73
5	500	90	61,11
6	600	92,1	59,72
7	700	91,6	60,04
8	800	89,5	61,45
9	900	90,2	60,98
10	1000	92,6	59,40

Dari tabel hasil pengujian parameter *trynumber* di atas dapat dilihat bahwa dengan rentang nilai *trynumber* = 100 ~ 1000 yang diujikan tidak terlihat perubahan hasil yang signifikan pada performansi. Namun, *makespan* terbaik sebesar 77 dihasilkan pada saat *trynumber* = 1000 dimana nilai efisiensi mencapai 71,43%. Besarnya nilai iterasi yang digunakan pada *prey behaviour* akan berpengaruh terhadap kecepatan konvergensi AFSA. Semakin besar nilai *trynumber* yang digunakan akan membuat AFSA mencapai kondisi konvergen lebih cepat. Namun, besarnya nilai *trynumber* yang digunakan juga harus memperhatikan ukuran dari populasi *fish swarm* yang digunakan.



Gambar 4- 4 Grafik hasil pengujian pengaruh parameter *Trynumber*

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Setelah melakukan implementasi, pengujian dan analisis, maka dapat diambil beberapa kesimpulan, yaitu:

1. *Artificial Fish Swarm Algorithm* (AFSA) yang dirancang untuk optimisasi solusi permasalahan penjadwalan *job shop*, mampu memecahkan kasus *job shop* dengan nilai efisiensi sistem terbaik yang pernah dicapai sebesar 75%.
2. Berdasarkan observasi yang dilakukan, kemampuan AFSA dalam permasalahan penjadwalan *job shop* pada penelitian ini terbukti cukup baik karena efisiensi tertinggi yang berhasil dicapai merupakan hasil pencarian dengan hanya membangkitkan 10000 *artificial fish* pada 100 generasi untuk ruang solusi sebesar  $3.72e+41$ .

### 5.2 Saran

Setelah menyelesaikan proses pengerjaan Tugas Akhir ini, nilai efisiensi sistem yang didapat dinilai masih belum memuaskan. Hal ini disebabkan oleh inisialisasi populasi solusi jadwal awal yang masih dibentuk secara acak serta ruang solusi pengujian yang relatif kecil dibandingkan dengan ruang solusi masalah yang sangat besar. Oleh karena itu saran untuk bisa mengembangkan sistem menjadi lebih baik adalah dengan mengkolaborasi AFSA dengan prosedur pencarian lokal seperti pendekatan *Nowicki and Smutnicki* (1996), Algoritma *Schrage*, dsb. untuk menghasilkan solusi jadwal yang lebih optimal.

## Daftar Pustaka

- [1] 82 JSP test instances. OR-Library [Online]. Diakses: 10 Februari 2014.  
Tersedia: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/jobshop1.txt>.
- [2] Adhy, Satriyo dan Kushartantya. “Penyelesaian Masalah Job Shop Menggunakan Algoritma Genetika”. Semarang, Indonesia: Universitas Diponegoro. Jurnal Masyarakat Informatika, Volume 1, Nomor 1, ISSN 2086-4390.
- [3] Azizi, Reza. “Empirical Study of Artificial Fish Algorithm”. 2014. Bojnourd, Iran: Islamic Azad University. International Journal of Computing , Communication and Networking Volume 3, No.1, January-March 2014. 2319-2720.
- [4] Cai, Yun. 2010. “Artificial Fish School Algorithm Applied in a Combinatorial Optimization Problem”. Wuhan, China: Wuhan University of Science and Technology / Country College of Machinery and Automation. Modern Education and Computer Science. I.J. Intelligent Systems and Applications, 2010, 1, 37-43.
- [5] Farzi, Saeed. 2010. “Efficient Job Scheduling in Grid Computing with Modified Artificial Fish Swarm Algorithm”. International Association of Computer Science and Information Technology (IACSIT). International Journal of Computer Theory and Engineering, Vol.1, No.1, April 2009, 13-17. 1793-8201.
- [6] Fernandes, Edite M. G. P., Martins, Tiago F. M. C. dan Rocha, Ana Maria A. C. 2009. “Fish Swarm Intelligent Algorithm for Bound Constrained Global Optimization”. Portugal: University of Minho. Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2009.
- [7] Heinonen, J. dan Pettersson, F. “Job-shop scheduling and visibility studies with a hybrid ACO algorithm”. Finland: Åbo Akademi University.
- [8] Huadong, Chen, Shuzong, Wang, Jingxi, Li, dan Yunfan, Li. “A Hybrid of Artificial Fish Swarm Algorithm and Particle Swarm Optimization for Feedforward Neural Network Training”. Wuhan, PR.China: Naval Univ. of Engineering and Commanding Communications Academy.
- [9] P, Surekha dan Sumathi, S. 2010. “Solving Fuzzy based Job shop Scheduling Problems using Ga and Aco”. Coimbatore: PSG College of Technology. Journal of Emerging Trends in Computing and Information Sciences Vol.1, No.2, Oct 2010. 95-102. E-ISSN-2218-6301
- [10] Pertiwi, Anggi Putri. 2011. “Penjadwalan Job shop Menggunakan Algoritma Bee Colony Optimization dan Tabu Search”. Bandung, Indonesia: Institut Teknologi Telkom.
- [11] Saidah, Nafiuna Hidayatus, Er, Mahendrawathi, Ph.D. dan Soelaiman,

- Rully, M.Kom. “Implementasi Algoritma Optimasi Bee Colony Untuk Penjadwalan Job shop”. Surabaya, Indonesia: Institut Teknologi Sepuluh November.
- [12] Sempena, Samsu. 2009. “Algoritma Genetik Hibrida dalam Penyelesaian Job-shop Scheduling Problem”. Bandung, Indonesia: Institut Teknologi Bandung.
- [13] Suyanto, S.T., M.Sc. 2008. “Evolutionary Computation”. Penerbit Informatika Bandung.
- [14] Yazdani, Danial, Toosi, Adel Nadjaran dan Meybodi, Mohammad Reza. “Fuzzy Adaptive Artificial Fish Swarm Algorithm”. Qazvin, Mashhad, Tehran, Iran: IAU of Qazvin, IAU of Mashhad and Amirkabir University of Technology.
- [15] Ye, Li dan Yan, Chen. 2010. “A Genetic Algorithm for Job-Shop Scheduling”. Dalian, PR.China: Dalian Maritime University. Journal of Software, Vol. 5, No. 3, March 2010.
- [16] Xu, Qianjun. 2001. “Introduction to Job shop Scheduling Problem”.