

# Analisis Perbandingan *Aggregat Of Function* (AOF) dengan *Non-Dominated Sorting Genetic Algorithm* (NSGA-II) dalam Menentukan Optimasi *Multi-Objective* pada Penjadwalan Mesin Produksi *Flow Shop*

Fifin Sonata\*, Dede Prabowo Wiguna\*\*

\*\*\* Program Studi Sistem Informasi, Sekolah Tinggi Ilmu Komputer Medan

---

## Article Info

### Article history:

Received June 30<sup>th</sup>, 2018

Revised Aug 04<sup>th</sup>, 2018

Accepted Aug 05<sup>th</sup>, 2018

---

### Keyword:

*Makespan*

*Multi-objective*

NSGA-II

AOF

*Flow Shop*

*Total tardiness*

---

## ABSTRACT

Penjadwalan mesin produksi dalam dunia industri memiliki peranan penting sebagai bentuk pengambilan keputusan. Salah satu jenis sistem penjadwalan mesin produksi adalah sistem penjadwalan mesin produksi tipe *flow shop*. Dalam penjadwalan *flow shop*, terdapat sejumlah pekerjaan (*job*) yang tiap-tiap *job* memiliki urutan pekerjaan mesin yang sama. Optimasi penjadwalan mesin produksi *flow shop* berkaitan dengan penyusunan penjadwalan mesin yang mempertimbangkan 2 objek yaitu *makespan* dan *total tardiness*. Optimasi kedua permasalahan tersebut merupakan optimasi yang bertolak belakang sehingga diperlukan model yang mengintegrasikan permasalahan tersebut dengan optimasi *multi-objective A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization : NSGA-II*. Dalam penelitian ini akan dibandingkan 2 buah metode yaitu *Aggregat Of Function* (AOF) dengan *NSGA-II* agar dapat terlihat nilai solusinya. Penyelesaian penjadwalan mesin produksi *flow shop* dengan algoritma *NSGA-II* untuk membangun jadwal dengan meminimalkan *makespan* dan *total tardiness*.

Tujuan yang ingin dicapai adalah mengetahui bahwa model yang dikembangkan akan memberikan solusi penjadwalan mesin produksi *flow shop* yang efisien berupa solusi pareto optimal yang dapat memberikan sekumpulan solusi alternatif bagi pengambil keputusan dalam membuat penjadwalan mesin produksi yang diharapkan. Solusi pareto optimal yang dihasilkan merupakan solusi optimasi *multi-objective* yang optimal dengan *trade-off* terhadap seluruh objek, sehingga seluruh solusi pareto optimal sama baiknya.

Copyright © 2018 STMIK Triguna Dharma.

All rights reserved.

---

## First Author

Nama : Fifin Sonata

Program Studi : Sistem Informasi Sekolah Tinggi Ilmu Komputer Medan (STIKOM Medan)

Email: fifinsonata2012@gmail.com

---

## 1. PENDAHULUAN

Penjadwalan mesin produksi di dalam dunia industri, baik industri manufaktur maupun agroindustri memiliki peranan penting sebagai bentuk pengambilan keputusan. Perusahaan berupaya untuk memiliki penjadwalan yang paling efektif dan efisien sehingga dapat meningkatkan produktivitas yang dihasilkan dengan total biaya dan waktu seminimal mungkin. Salah satu model yang dapat diterapkan dalam keadaan *made-to-order* pada perusahaan manufaktur adalah model penjadwalan *flow shop*.

Dalam penjadwalan *flowshop*, terdapat sejumlah pekerjaan (*job*) yang tiap-tiap *job* memiliki urutan pekerjaan mesin yang sama. Suatu penjadwalan dapat dimodelkan sebagai permasalahan penjadwalan *flowshop* apabila urutan pekerjaannya selaras. Permasalahan penjadwalan mesin produksi biasanya terletak pada penyusunan dan pengaturan *job-job* yang akan di proses pada serangkaian mesin. Salah satu kesulitan dalam melakukan penyusunan dan pengaturan *job* terhadap mesin yang tersedia adalah sulitnya mencari teknik-teknik yang tepat untuk membuat model penjadwalan mesin produksi yang optimal dan memenuhi segala kriteria-kriteria penjadwalan yang telah ditetapkan.

Penjadwalan *flow shop* berkembang dari *single-objective* (optimasi dengan satu fungsi) menjadi *multi-objective* (optimasi dengan beberapa fungsi objektif). Dalam kasus *multi-objective*, akan dihasilkan sekumpulan solusi optimal yang dikenal dengan *pareto-optimal solutions* (solusi pareto-optimal) [4].

Beberapa penelitian telah mengembangkan model yang berkaitan dengan penjadwalan mesin produksi *flow shop*, baik yang menggunakan optimasi *single-objective* ataupun *multi-objective*. Penelitian yang hampir serupa juga pernah dilakukan oleh beberapa ahli diantaranya [1],[2],[11] dan [3]. [12] meneliti 2 fungsi objektif yaitu *makespan* dan *total flow time*. Secara umum, semua penelitian diatas memiliki kinerja komputasi yang baik, mampu memformulasikan secara matematis fungsi objektif dalam penjadwalan *flow shop*. Tetapi sebagian besar metode yang digunakan hanya mampu menyelesaikan permasalahan secara dependent yaitu hanya dapat digunakan pada permasalahan tertentu bergantung pada jenis permasalahan (*heuristic*).

Dalam kasus optimasi *multi-objective*, algoritma *Non-Dominated Sorting Genetic Algorithm for Multi-objective Optimization : NSGA-II* yang merupakan kelompok Algoritma *Metaheuristic* yang telah diuji kehandalannya dibandingkan dengan optimasi *multi-objective* lainnya. NSGA-II merupakan metode pengembangan dari Genetik Algoritma (GA) dan NSGA. Dibandingkan dengan GA dan NSGA, NSGA-II dibedakan pada penggunaan operator *crowding distance* agar menghasilkan solusi pareto optimal yang lebih baik. Penelitian menggunakan NSGA-II [8] yang meneliti tentang *multi-objective* untuk *vehicle routing problems*, [9] melakukan penelitian kasus optimasi *multi-objective* pada kasus manajemen portofolio dan [4] membuat penelitian yang mampu menciptakan metode baru bernama *omni optimizer* yang di adopsi dari NSGA-II untuk kasus optimasi baik *single* maupun *multi-objective*.

Dari permasalahan di atas, maka perlu dilakukan penelitian untuk menganalisis perbandingan antara metode *Aggregat Of Function* (AOF) yang prinsipnya menerapkan optimasi *single-objective* seperti Algoritma Genetika pada permasalahan *multi-objective* dengan algoritma *multi-objective* NSGA-II dalam penjadwalan mesin produksi *flow shop* untuk mengoptimalkan 2 fungsi objektif yaitu *makespan* dan *total tardiness* sehingga memberikan sekumpulan solusi alternatif bagi pengambil keputusan.

## 2. METODE PENELITIAN

### 2.1 Jenis Penelitian

Penelitian ini menggunakan metode penelitian kuantitatif. Dimana data yang diolah berupa angka sedangkan *ouput* yang dihasilkan berupa angka dan *chart* statistik. Pada penelitian ini menguji 2 algoritma yang berbeda yaitu AOF dan NSGA-II pada penjadwalan mesin produksi *flow shop*.

### 2.2 Lokasi Penelitian

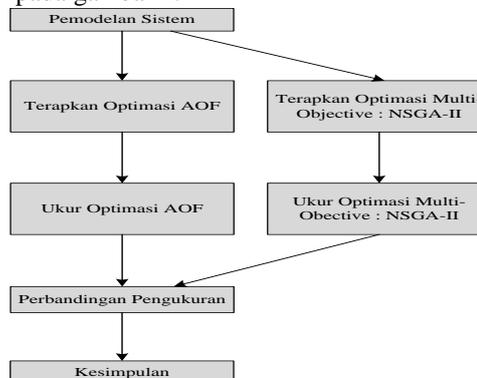
Lokasi penelitian dilakukan di Laboratorium Komputer Sekolah Tinggi Ilmu Komputer (STIKOM) Medan.

### 2.3 Alat Penelitian

Penelitian ini menggunakan *software* yang mendukung untuk mengolah dan analisa data seperti uji normalitas dan uji homogenitas yaitu dengan menggunakan *software* SPSS. Selain itu juga menggunakan *software* penunjang yaitu *MathLab* dan *Dreamweaver*.

### 2.4 Metode dan Desain Penelitian

Dalam optimasi penjadwalan *flow shop* mesin produksi akan di bandingkan nilai *makespan* dan *total tardiness* sebelum dan sesudah diterapkannya optimasi *multi-objective* NSGA-II. Secara umum, diagram alir rancang penelitiannya dapat digambarkan pada gambar 1.



Gambar 1. Diagram Alir Rancangan Penelitian

## 2.5 Pengumpulan Data

Untuk menguji sistem, data yang diujikan berupa data sekunder yang diperoleh dari Eva Instances [6], [10] : <http://www.upv.es/gio/rruiz> yang terdiri dari 110 *instance* kasus data yang akan diujikan.

Tabel 1. Tabel Data Penjadwalan Mesin Produksi

Job/Mesin	M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>	....	M <sub>n</sub>	Due-Date
J <sub>1</sub>	10	13	4	32	....	100	6
J <sub>2</sub>	24	30	4	70	....	80	8
....	....	....	....	....	....	....	....
J <sub>n</sub>	18	66	1	43	....	95	9

Data-data tersebut selanjutnya akan digunakan untuk memodelkan sistem dan diproses dengan menggunakan metode AOF dan algoritma *Multi-objective* : NSGA-II. Kemudian hasilnya akan dibandingkan.

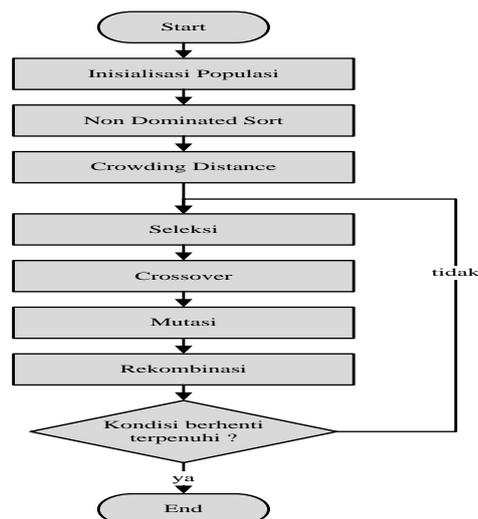
## 2.6 Pemodelan Sistem

Untuk mendapatkan solusi yang optimal dengan algoritma *multi-objective* NSGA-II, maka permasalahan penjadwalan mesin produksi *flow shop* akan dimodelkan secara matematis dalam bentuk persamaan *multi-objective* yang terdiri beberapa fungsi objektif dan pembatas. Persamaan *multi-objective* terdiri dari dua buah fungsi objektif yaitu fungsi yang memformulasikan nilai *makespan*, serta fungsi yang memformulasikan total tardiness.

Selain itu perlu untuk mendefinisikan variabel solusi yang ingin dihasilkan, sebab optimasi NSGA-II dimulai dengan inialisasi populasi secara acak sesuai dengan definisi variabel solusi.

## 2.7 Tahapan NSGA-II

Optimasi *Multi-objective* : NSGA-II yang digunakan untuk mencari solusi pareto-optimal dari model matematis *multi-objective* dapat dibagi menjadi tahapan-tahapan yaitu Inialisasi Populasi, *Non-Dominated Sort*, *Crowding Distance*, Seleksi, Operator Genetika, *crossover*, mutasi, Rekombinasi. Dimana pada beberapa tahapan diatas, akan ditentukan prosedur yang lebih rinci untuk penerapan NSGA-II pada permasalahan penjadwalan mesin produksi *flow shop*. Diagram alir secara umum dari NSGA-II ditampilkan dalam Gambar 2 dibawah ini.



Gambar 2. Diagram Alir Umum NSGA-II

## 2.8 Pengukuran dan Perbandingan Optimasi Menggunakan NSGA-II

Hasil optimasi *multi-objective* NSGA-II akan diukur dan dibandingkan dengan hasil optimasi dengan AOF, dengan cara menghitung dan membandingkan hasil dari kedua fungsi objektif yaitu *makespan* dan total tardiness menggunakan optimasi AOF dengan NSGA-II.

Untuk mengevaluasi kualitas populasi solusi yang dihasilkan oleh NSGA-II, akan digunakan indikator *hypervolume* [13] dan [14]. Indikator *hypervolume* didefinisikan sebagai volume dari seluruh kemungkinan solusi yang didominasi oleh sekumpulan solusi yang dihasilkan EMO. Semakin besar nilai *hypervolume* maka semakin baik solusi dari EMO tersebut. Pendekatan *Monte Carlo* digunakan untuk memperkirakan *hypervolume* dengan menghitung sekumpulan nilai acak pada ruang solusi yang didominasi oleh solusi dari EMO.

Karena NSGA-II merupakan algoritma stokastik, maka perlu dilakukan analisis statistik dari hasil penelitian yang diperoleh agar hasil tersebut dapat dibandingkan dengan tingkat kepercayaan tertentu. Uji normalitas dan uji homogenitas akan dilakukan terlebih dahulu, agar dapat ditentukan uji selanjutnya untuk membandingkan rata-rata hasil penelitian. Uji normalitas yang digunakan adalah uji Kolmogorov-Smirnov dan uji Shapiro-Wilk, sedangkan uji homogenitas menggunakan uji Levene. Hasil penelitian yang berdistribusi normal dengan variansi yang homogen akan diselesaikan dengan uji ANOVA, sedangkan untuk yang berdistribusi normal dengan variansi tidak homogen maka digunakan uji Welch

## 2.9 Analisis Data

Analisis Data menggunakan model matematis :

[12] telah memformulasikan permasalahan minimasi *makespan* dengan model matematis sebagai berikut :

$$M(S) = C_{m,n} \quad (1)$$

$$C_{m,n} = \max \{C_{m-1,n}, C_{m,n-1}\} + t_{m,n} \quad (2)$$

Dimana :

$M(S)$	=	<i>makespan</i> (waktu untuk memproses seluruh <i>job</i> hingga selesai)
$C_{m,n}$	=	waktu komplit (completion time) <i>job</i> ke - m pada mesin ke - n
$t_{m,n}$	=	waktu proses untuk <i>job</i> m pada mesin n
m	=	jumlah total <i>job</i> dalam penjadwalan
n	=	jumlah total mesin dalam penjadwalan

selain memformulasikan *makespan*, juga memformulasikan permasalahan minimasi total *tardiness* dengan model matematis sebagai berikut :

$$T = \sum_{i=0}^n \max [0, C_i - d_i] \quad (3)$$

Dimana :

T = total *tardiness* (total waktu terlambat suatu *job* dapat diselesaikan lebih cepat dari *due-date*-nya)

$C_i$  = waktu komplit dimana seluruh mesin menyelesaikan suatu proses pada *job* i

Sehingga dari persamaan (1) sampai dengan persamaan (3) dapat dimodelkan menjadi sebuah permasalahan *multi-objective* dengan model matematis sebagai berikut :

$$\left. \begin{aligned} \text{Min } Z_M &= \max \{C_{m-1,n}, C_{m,n-1}\} + t_{m,n} \\ \text{Min } Z_T &= \sum_{i=0}^n \max [0, C_i - d_i] \end{aligned} \right\} \quad (4)$$

Dimana :

m = jumlah total *job* dalam penjadwalan

n = jumlah total mesin dalam penjadwalan

$t_{m,n}$  = waktu proses untuk *job* ke-m pada mesin ke-n

$C_i$  = waktu komplit dimana semua mesin menyelesaikan suatu proses pada *job* i ( $i=1, \dots, m$ ); .

## 3. ANALISA DAN HASIL

### 3.1 ANALISA

#### 3.1.1 Model Matematis

Selain [12] yang telah memformulasikan permasalahan minimasi *makespan* dengan model matematis pada persamaan 2, pada penelitian ini juga menggunakan persamaan matematis yang telah sebagai berikut :

$$T = \sum_{i=0}^n \max [0, C_i - d_i] \quad (5)$$

Dimana : T = total *tardiness* (total waktu terlambat suatu *job* dapat diselesaikan lebih cepat dari *due-date*-nya),  $C_i$  = waktu komplit dimana seluruh mesin menyelesaikan suatu proses pada *job* i

Sehingga dari persamaan (1) sampai dengan persamaan (5) dapat dimodelkan menjadi sebuah permasalahan *multi-objective* dengan model matematis sebagai berikut :

$$\left. \begin{aligned} \text{Min } Z_M &= \max \{C_{m-1,n}, C_{m,n-1}\} + t_{m,n} \\ \text{Min } Z_T &= \sum_{i=0}^n \max [0, C_i - d_i] \end{aligned} \right\} \quad (6)$$

Dimana : m = jumlah total *job* dalam penjadwalan, n = jumlah total mesin dalam penjadwalan,  $t_{m,n}$  = waktu proses untuk *job* ke-m pada mesin ke-n,  $C_i$  = waktu komplit dimana semua mesin menyelesaikan suatu proses pada *job* i ( $i=1, \dots, m$ );

#### 3.1.2 Implementasi Tahapan NSGA-II

Optimasi *multi-objective* NSGA-II yang digunakan untuk mencari solusi model matematis dari Persamaan (7) terbagi atas beberapa tahapan sebagai berikut :

### 1. Inisialisasi Populasi

Populasi diinisialisasi secara acak, namun harus tetap mematuhi kaidah pembatas dari model matematis.

### 2. Non-Dominated Sort

Populasi yang telah diinisialisasi lalu diurutkan berdasarkan *non domination* menggunakan algoritma *Fast Non-dominated Sorting* yang didefinisikan oleh Deb [4] sebagai berikut :

- Untuk setiap individu  $p$  di populasi  $P$ , dilakukan :
    - Inisialisasi  $S_p = \emptyset$ . Himpunan ini akan berisi semua individu yang didominasi oleh  $p$ .
    - Inisialisasi  $n_p = 0$ . Menunjukkan jumlah individu yang mendominasi individu  $p$ .
    - Untuk setiap individu  $q$  di  $P$ 
      - \* Jika  $p$  mendominasi  $q$  maka tambahkan  $q$  ke himpunan  $S_p$ . ( $S_p = S_p \cup q$ )
      - \* Jika  $q$  mendominasi  $p$  maka increment *counter* dominasi  $n_p$ . ( $n_p = n_p + 1$ )
    - Jika  $n_p = 0$  berarti tidak ada individu yang mendominasi  $p$  sehingga  $p$  berada di *front* pertama; Beri nilai peringkat  $p$  menjadi satu ( $p_{\text{rank}} = 1$ ). Tambahkan  $p$  ke *front* pertama ( $F_1 = F_1 \cup \{p\}$ )
  - Inisialisasi *counter front* dengan satu ( $i=1$ )
  - Selama *front* yang ke- $i$  tidak kosong ( $F_i \neq \emptyset$ ), lakukan :
    - $Q = \emptyset$
    - Untuk setiap individu  $p$  di *front*  $F_i$ 
      - \* Untuk setiap individu  $q$  pada  $S_p$
    - $n_q = n_q - 1$ , decrement jumlah dominasi untuk individu  $q$
    - jika  $n_q = 0$  maka tidak ada individu di *front* berikutnya yang mendominasi  $q$ , sehingga nilai  $q_{\text{rank}} = i + 1$ .
- Tambahkan  $q$  ke  $Q$  ( $Q = Q \cup q$ ).
- increment *counter front* dengan satu ( $i=i+1$ )
  - $Q$  menjadi *front* berikutnya ( $F_i = Q$ )

### 3. Crowding Distance

*Crowding distance* digunakan sebagai pembanding antara dua individu dalam *front* yang sama, agar solusi yang dihasilkan dapat mewakili keseluruhan solusi pareto-optimal.

Cara untuk menghitung *crowding distance* didefinisikan oleh [5] adalah sebagai berikut :

- Untuk setiap individu pada *front*  $F_i$ , dengan  $n$  adalah jumlah individu
    - Inisialisasi jarak dengan nol untuk semua individu ( $F_i(d_j)=0$ ), dimana  $j$  menunjukkan individu ke- $j$  pada *front*  $F_i$
    - Untuk setiap fungsi tujuan  $m$ 
      - \* Urutkan individu di *front*  $F_i$  berdasarkan nilai obyektif  $m$
      - \* Beri nilai tak terhingga untuk jarak individu pertama dan terakhir ( $I(d_1)=\infty$  dan  $I(d_n)=\infty$ )
      - \* Untuk  $k$  mulai dari 2 sampai  $(n-1)$
      - $I(d_k) = I(d_k) + \frac{I(k+1).m - I(k-1).m}{f_m^{\max} - f_m^{\min}}$
      - $I(k).m$  adalah nilai obyektif  $m$  untuk individu  $k$  pada  $I$
- ### 4. Seleksi

Digunakan adalah *Roulette Wheel Selection*. Pada metode roda rolet kromosom orang tua (*parents*) yang akan dipindah-silangkan dipilih berdasarkan nilai *fitness*-nya. Nilai *fitness* dari masing-masing kromosom dibagi dengan total nilai *fitness* seluruh kromosom yang ada pada populasi. Setiap kromosom dianggap merupakan potongan/bagian dari roda rolet dengan ukuran potongan yang proporsional dengan nilai *fitness*-nya. Suatu nilai target antara angka nol dan angka satu ditetapkan secara acak. Kemudian roda rolet diputar sebanyak  $N$  kali, dimana  $N$  adalah jumlah individual atau kromosom dalam populasi. Dari setiap putaran, kromosom dengan nilai *fitness* yang berada di bawah nilai target dipilih untuk menjadi *parents* bagi generasi berikutnya.

### 5. Operator Genetika

Operator genetika yang digunakan adalah *Precedence Preservative Crossover* (PPX) untuk *crossover* dan operator mutasi adalah *remove* dan *insert* mutasi. PPX [7] dapat dijelaskan sebagai berikut :

- a. String baru disusun secara acak dari *allele string* induk.
- b. Angka acak 1 atau 2 dipakai untuk memilih induk.
- c. Jika 1 diturunkan *allele* paling kiri dari induk pertama, jika 2 diturunkan *allele* paling kiri dari induk kedua. Selanjutnya terpilih tadi dihapus dari kedua induk.
- d. Proses dilakukan sampai karakter di kedua induk habis.

## 6. Rekombinasi

Populasi hasil *crossover* dan mutasi kemudian dikombinasikan dengan populasi induk yang kemudian dipilih dengan *Non-Dominated Sort* dan *Crowding Distance* untuk mendapatkan populasi generasi berikutnya.

### 3.2 HASIL

Data yang akan digunakan dalam penelitian adalah data yang diambil dari Eva Instances: [6]. Data yang diujikan pada penelitian terdiri dari 150 Job 30 Mesin disebut dengan Instance I\_150\_30 dan 250 Job 50 Mesin disebut dengan Instance I\_250\_50.

#### 3.2.1 Optimasi penjadwalan mesin produksi menggunakan AOF

Pada prinsipnya, metode ini menerapkan optimasi *single-objective* seperti Algoritma Genetika pada permasalahan *multi-objective*.

Kedua fungsi objektif pada persamaan (6) digabung menjadi sebuah fungsi objektif yang baru dengan operator penjumlahan (+) dan masing-masing fungsi diberi bobot antara 0 dan 1, dimana jumlah kedua bobot tersebut harus sama dengan 1. Fungsi objektif baru dari hasil penggabungan dari kedua fungsi objektif. Persamaan (7) adalah :  $\text{Min } Z = \alpha * (\max\{C_{i-1,j}, C_{i,j-1}\} + t_{i,j}) + (1-\alpha) * (\sum_{i=0}^n \max[0, c_i - d_i])$  dengan  $\alpha$  adalah bobot bernilai antara 0 dan 1. Digunakan Algoritma Genetika, dengan penentuan bobot  $\alpha$  dilakukan melalui penelitian dengan variasi  $\alpha = 0,1, 0,5$  dan  $0,9$  yang datanya ditabulasikan dalam Tabel 2 dan Tabel 3 dibawah ini.

Tabel 2. Hasil Penelitian AOF Instance I\_150\_30

$\alpha$	Parameter	Hasil Penelitian				
		1	2	3	4	5
0,1	ZM	11327	11153	11192	11200	11162
	ZT	133433	119589	123218	122785	129088
	Jumlah Generasi	178	255	140	251	209
0,5	ZM	11049	11332	11333	11184	11258
	ZT	130392	134616	138815	130638	130654
	Jumlah Generasi	294	176	157	324	226
0,9	ZM	11046	11345	11228	11464	10994
	ZT	121534	137081	139031	154819	129969
	Jumlah Generasi	188	158	138	173	220

Tabel 3. Hasil Penelitian AOF Instance I\_250\_50

$\alpha$	Parameter	Hasil Penelitian				
		1	2	3	4	5
0,1	ZM	19196	19731	19422	19689	19459
	ZT	453440	479169	522631	534560	482118
	Jumlah Generasi	500	268	256	125	374
0,5	ZM	19750	19459	19380	19584	19615
	ZT	508587	505159	471894	497512	539113
	Jumlah Generasi	169	165	270	294	180
0,9	ZM	19773	19599	19442	20015	19288
	ZT	503433	482715	499647	500287	468795
	Jumlah Generasi	124	324	154	131	451

Untuk menganalisa hasil penelitian AOF diatas, terlebih dahulu dilakukan uji normalitas dan uji homogenitas, agar dapat menentukan penggunaan uji yang membandingkan rata-rata hasil penelitian. Uji normalitas yang digunakan adalah uji Kolmogorov-Smirnov dan uji Shapiro-Wilk, sedangkan uji homogenitas menggunakan uji Levene. Hasil penelitian yang berdistribusi normal dengan variansi yang homogen akan dibandingkan dengan uji ANOVA, sedangkan untuk yang berdistribusi normal dengan variansi tidak homogen menggunakan uji Welch. Khusus untuk hasil penelitian yang tidak berdistribusi normal maka digunakan uji Kruskal-Wallis. Perhitungan uji statistik diatas menggunakan bantuan perangkat lunak SPSS, dengan hasil analisa ditabulasikan dalam tabel dibawah ini.

Tabel 4. Hasil Uji Normalitas AOF

Instance	Alpha	Kolmogorov-Smirnov			Shapiro-Wilk			
		Statistic	df	Sig.	Statistic	df	Sig.	
I_150_30	ZM	0,10	0,339	5	0,062	0,794	5	0,073
		0,50	0,201	5	0,200	0,889	5	0,353
		0,90	0,204	5	0,200	0,945	5	0,702
	ZT	0,10	0,268	5	0,200	0,933	5	0,616
		0,50	0,340	5	0,060	0,792	5	0,070
		0,90	0,218	5	0,200	0,970	5	0,878
	Jumlah Generasi	0,10	0,218	5	0,200	0,924	5	0,556
		0,50	0,194	5	0,200	0,924	5	0,558

I_250_50	ZM	0,90	0,142	5	0,200	0,989	5	0,975
		0,10	0,208	5	0,200	0,932	5	0,611
		0,50	0,173	5	0,200	0,977	5	0,920
	ZT	0,90	0,139	5	0,200	0,986	5	0,962
		0,10	0,243	5	0,200	0,928	5	0,582
		0,50	0,232	5	0,200	0,963	5	0,831
	Jumlah Generasi	0,90	0,321	5	0,102	0,847	5	0,186
		0,10	0,203	5	0,200	0,977	5	0,917
		0,50	0,319	5	0,107	0,803	5	0,085
		0,90	0,316	5	0,115	0,827	5	0,131

Hasil uji normalitas pada Tabel 4 dengan uji Kolmogorov-Smirnov dan Shapiro-Wilk, menunjukkan bahwa hasil penelitian AOF untuk ZM, ZT dan Generasi pada  $\alpha = 0,1, 0,5$  dan  $0,9$  berdistribusi normal dengan taraf signifikansi  $0,05$ .

Tabel 5. Hasil Uji Homogenitas AOF

Instance		Levene Statistic	df1	df2	Sig.
I_150_30	ZM	3,126	2	12	0,081
	ZT	1,865	2	12	0,197
	Jumlah Generasi	2,694	2	12	0,108
I_250_50	ZM	1,084	2	12	0,369
	ZT	1,948	2	12	0,185
	Jumlah Generasi	2,130	2	12	0,162

Hasil uji homogenitas pada Tabel 5 dengan uji Levene, menunjukkan bahwa hasil penelitian AOF untuk ZM, ZT dan Generasi pada  $\alpha = 0,1, 0,5$  dan  $0,9$  mempunyai variansi yang homogen dengan taraf signifikansi  $0,05$  (kolom Sig. pada tabel 5 bernilai lebih besar dari  $0,05$ ). Dari hasil kedua uji, hasil eksperimen AOF berdistribusi normal dengan variansi yang homogen, sehingga digunakan uji ANOVA untuk membandingkan nilai rata-rata ZM, ZT dan Generasi pada  $\alpha = 0,1, 0,5$  dan  $0,9$  dengan hasil uji ditabulasikan dalam tabel dibawah ini.

Tabel 6. Hasil Uji ANOVA AOF

Instance		Sum of Squares	df	Mean Square	F	Sig.	
I_150_30	ZM	Between Groups	1531,600	2	765,800	0,039	0,961
		Within Groups	232752,800	12	19396,067		
		Total	234284,400	14			
	ZT	Between Groups	307991120,400	2	153995560,200	2,350	0,138
		Within Groups	786520738,000	12	65543394,833		
		Total	1094511858,400	14			
	Jumlah Generasi	Between Groups	9004,800	2	4502,400	1,567	0,248
		Within Groups	34475,600	12	2872,967		
		Total	43480,400	14			
I_250_50	ZM	Between Groups	38488,133	2	19244,067	0,389	0,686
		Within Groups	593259,600	12	49438,300		
		Total	631747,733	14			
	ZT	Between Groups	491090575,600	2	245545287,800	0,384	0,689
		Within Groups	7674834746,400	12	639569562,200		
		Total	8165925322,000	14			
	Jumlah Generasi	Between Groups	21612,133	2	10806,067	0,727	0,503
		Within Groups	178319,200	12	14859,933		
		Total	199931,333	14			

Hasil uji ANOVA, menunjukkan bahwa nilai rata-rata ZM, ZT dan Generasi tidak memiliki perbedaan signifikan untuk nilai  $\alpha=0,1, 0,5$  dan  $0,9$  pada taraf signifikansi  $0,05$  (nilai kolom Sig pada Tabel 6 lebih besar dari  $0,05$ ). Hal ini berarti perubahan  $\alpha$  tidak berpengaruh secara signifikan terhadap perubahan nilai rata-rata dari fungsi objektif ke-1 (ZM), fungsi objektif ke-2 (ZT) dan jumlah Generasi. Algoritma Genetika untuk optimasi AOF menerapkan operator PPX dan Remove dan insert yang sama dengan Operator Genetika untuk tahapan NSGA-II dengan parameternya sebagaimana ditabulasikan pada Tabel 7 dibawah ini.

Tabel 7. Parameter Algoritma Genetika untuk AOF

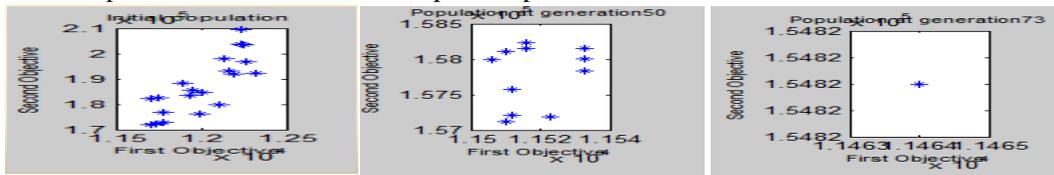
No	Nama Parameter	Nilai
1	Jumlah Populasi	20
2	Bobot ( $\alpha$ )	0,1

3	Probability of Crossover ( $P_c$ )	0,9
4	Probability of Mutation ( $P_m$ )	0,1

Untuk menunjukkan perubahan populasi berkaitan dengan bertambahnya generasi, maka Gambar 3 menampilkan populasi Algoritma Genetika untuk penjadwalan mesin produksi pada saat inisialisasi populasi, generasi ke-50 dan 73. Ternyata nilai fungsi objektif ke-1 (ZM) dan nilai fungsi objektif ke-2 (ZT) selalu lebih baik dari generasi sebelumnya.

**3.2.2 Optimasi Penjadwalan Mesin Produksi Menggunakan NSGA-II**

Untuk mendapatkan sekumpulan solusi pareto-optimal terbaik dengan efisien, perlu dilakukan percobaan untuk menentukan jumlah generasi terkecil yang dibutuhkan untuk mendapatkan solusi pareto-optimal. Kualitas kumpulan solusi pareto-optimal dari generasi yang berbeda dibandingkan berdasarkan estimasi rata-rata dari indikator hypervolume. Indikator hypervolume diestimasi menggunakan pendekatan Monte Carlo, yaitu dengan menormalisasi nilai semua fungsi objektif antara 0 dan 1, kemudian membangkitkan sekumpulan nilai fungsi objektif secara acak, dan menguji setiap nilai fungsi objektif acak tersebut apakah didominasi salah solusi pareto-optimal.



(i) Inisialisasi Populasi (ii) Populasi Generasi ke-50 (iii) Populasi Generasi ke-73

Gambar 3. Populasi pada Algoritma Genetika untuk *Makespan* dan Total *Tardiness*

Penelitian untuk estimasi hypervolume dilakukan sebanyak dua kali untuk setiap instance pada generasi yang sama, yang hasilnya ditabulasikan ke dalam Tabel 8 dibawah ini.

Tabel 8. Hasil Penelitian Estimasi Hypervolume NSGA-II

Instance	Estimasi Hypervolume pada Penelitian ke-					Rata-rata
	1	2	3	4	5	
I_250_50	0,74693	0,56053	0,62099	0,43587	0,79323	0,6315100
I_150_30	0,44204	0,54995	0,69228	0,7856	0,62527	0,6190280

Pengujian normalitas dengan SPSS terhadap data pada Tabel 8 ditabulasikan ke dalam tabel 9

Pengujian normalitas adalah pengujian tentang kenormalan distribusi data. Uji ini merupakan pengujian yang paling banyak dilakukan untuk analisis statistik parametrik. Karena data yang berdistribusi normal merupakan syarat dilakukannya tes parametrik.

Tabel 9. Hasil Uji Normalitas Estimasi Hypervolume NSGA-II

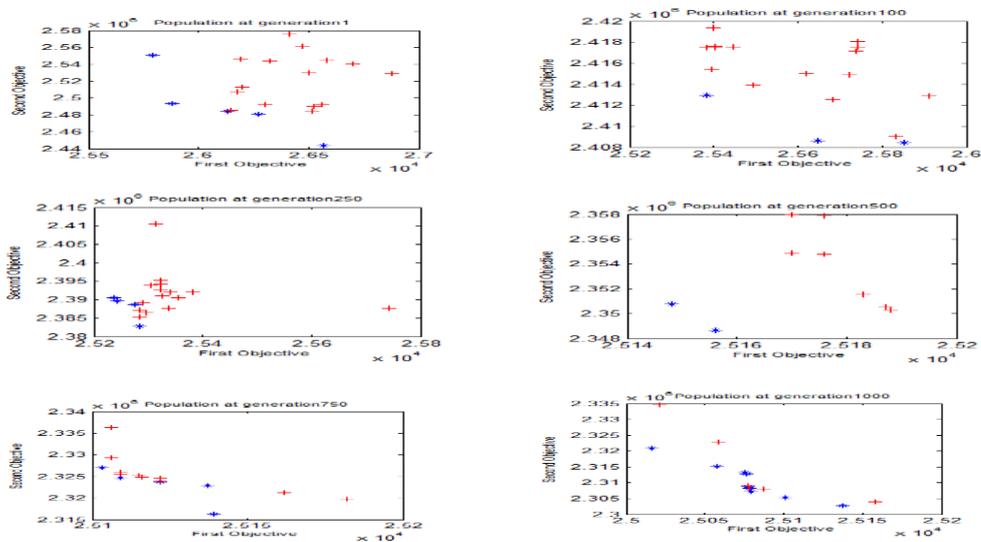
Instances		Kolmogorov-Smirnov			Shapiro-Wilk		
		Statistic	df	Sig.	Statistic	df	Sig.
Hypervolume	I_150_30	0,119	5	0,200	0,997	5	0,997
	I_250_50	0,189	5	0,200	0,963	5	0,827

Parameter yang digunakan untuk penelitian optimasi penjadwalan mesin produksi dengan NSGA-II ditabulasikan ke dalam Tabel 10 hampir sama dengan parameter Algoritma Genetika untuk AOF.

Untuk menunjukkan perubahan populasi berkaitan dengan bertambahnya generasi, maka Gambar 4 menampilkan populasi NSGA-II pada saat inisialisasi populasi, generasi ke-100, 250, 500, 750 dan 1000.

Tabel 9. Parameter NSGA-II

No	Nama Paramater	Nilai
1	Jumlah Populasi	20
2	Generasi	1000
3	Probability of Crossover ( $P_c$ )	0,9
4	Probability of Mutation ( $P_m$ )	0,1



Gambar 4. Populasi pada NSGA-II untuk *Makespan* dan *Total Tardiness*

Pada Gambar 4 terlihat bahwa 20 solusi yang terdapat pada populasi tersebar merata mendekati solusi pareto-optimal dengan semakin bertambahnya jumlah generasi. Bersama dengan bertambahnya generasi, maka nilai fungsi objektif ke-1 dan ke-2 akan semakin mengecil secara bersamaan. Terlihat bahwa dari 20 solusi yang dihasilkan, tidak semuanya merupakan solusi pareto, optimal, pada contoh gambar ini hanya ada 12 solusi pareto optimal yang berada pada front 1 (disimbolkan dengan ‘\*’ berwarna biru), sedangkan 8 solusi lainnya berada pada front 2 (simbol ‘+’ berwarna merah). Populasi yang didapat pada generasi ke-1000 NSGA-II ditabulasikan kedalam Tabel 10 dibawah ini.

Tabel 10. Populasi NSGA-II pada generasi ke-1000

Solusi ke-	$Z_M$	$Z_T$	Front	Crowding Distance
1	25101	2305256	1	0,321962797
2	25075	2313357	1	0,281232594
3	25078	2308285	1	0,067711391
4	25076	2312753	1	0,263258517
5	25058	2315153	1	0,900309604
6	25078	2308285	1	0
7	25077	2308887	1	0,263148123
8	25079	2307208	1	0,357273681
9	25137	2302717	1	65535
10	25078	2308285	1	0,041492922
11	25016	2320834	1	65535
12	25101	2305256	1	0,405264769
13	25021	2334534	2	65535
14	25158	2304022	2	65535
15	25087	2308031	2	0,726049629
16	25077	2308971	2	0,613319063
17	25078	2308358	2	0,018016365
18	25078	2308358	2	0
19	25059	2322840	2	1,246560645
20	25078	2308358	2	0

### 3.2.3 Perbandingan AOF dan NSGA-II

Solusi-solusi dari permasalahan multi-objective dapat dibandingkan dengan menggunakan dominasi. Solusi ke-i dinyatakan lebih baik dari solusi ke-j jika solusi ke-i mendominasi solusi ke-j. Solusi ke-i mendominasi solusi ke-j apabila semua nilai fungsi objektif solusi ke-i tidak lebih buruk dari solusi ke-j, dan minimal satu nilai fungsi objektif solusi ke-i mempunyai nilai yang lebih baik dari fungsi objektif solusi ke-j. Perbandingan dominasi dilakukan antara solusi Algoritma Genetika AOF untuk  $\alpha = 0.1$  dan solusi

NSGA-II. Digunakan simbol 'M' untuk mendominasi, simbol 'D' untuk didominasi, dan simbol '-' berarti solusi tidak saling mendominasi.

Tabel 11. Perbandingan dominasi Solusi AOF dengan Solusi NSGA-II untuk Instance I\_150\_30

Solusi	NSGA-II									
	1	2	3	4	5	6	7	8	9	10
AOF1	-	-	-	-	-	-	-	-	-	-
AOF2	-	-	-	-	-	-	-	-	-	-
AOF3	-	-	-	-	-	-	-	-	-	-
AOF4	-	-	-	-	-	-	-	-	-	-
AOF5	-	-	-	-	-	-	-	-	-	-

Tabel 12. Perbandingan dominasi Solusi AOF dengan Solusi NSGA-II untuk Instance I\_250\_50

Solusi	NSGA-II									
	1	2	3	4	5	6	7	8	9	10
AOF1	-	-	-	-	-	-	-	-	-	-
AOF2	-	-	D	-	D	-	-	-	D	D
AOF3	D	D	D	D	D	D	D	D	D	D
AOF4	D	D	D	D	D	D	D	D	D	D
AOF5	D	D	D	D	D	D	D	D	D	D

Hasil perbandingan dominasi solusi AOF dengan solusi NSGA-II pada Tabel 11 s.d Tabel 12 menunjukkan bahwa selalu ada solusi AOF yang didominasi oleh solusi NSGA-II, sedangkan solusi AOF tidak ada yang mendominasi solusi NSGA-II. Hal ini berarti solusi pareto-optimal NSGA-II berhasil memberikan solusi yang lebih baik dari solusi AOF. Berdasarkan perbandingan diatas, terlihat bahwa populasi solusi pareto-optimal NSGA-II lebih baik dengan berhasil mendominasi 100 % solusi sistem AOF. Begitu juga perbandingan solusi NSGA-II terhadap solusi AOF menunjukkan bahwa populasi solusi pareto-optimal NSGA-II lebih baik dengan berhasil mendominasi 100 % solusi AOF.

## 4. KESIMPULAN

### 4.1 Kesimpulan

Dari hasil penelitian dan pembahasan yang telah dilakukan untuk menjawab setiap topik permasalahan, maka dapat disimpulkan beberapa hal sebagai berikut :

1. Hasil penelitian pada NSGA-II diperoleh 20 solusi yang terdapat pada populasi semakin tersebar merata mendekati solusi pareto-optimal dengan semakin bertambahnya jumlah generasi, dan bersamaan dengan bertambahnya generasi maka nilai *makespan* dan total *tardiness* akan semakin minimal secara bersamaan.
2. Hasil analisa solusi pareto-optimal yang dihasilkan NSGA-II menunjukkan bahwa solusi pareto-optimal lebih baik dengan mendominasi 100 % solusi dari sistem yang dihasilkan Algoritma Genetika dengan *Aggregat of Function* (AOF).

### 4.2 Saran

1. Solusi pareto optimal dapat ditingkatkan kualitasnya dengan melakukan penelitian lebih mendalam terhadap parameter yang mempengaruhi solusi NSGA-II seperti Probabilitas *Crossover* dan Probabilitas Mutasi
2. Selain penelitian terhadap parameter, solusi pareto-optimal dapat ditingkatkan kualitasnya dengan melakukan penelitian terhadap operator genetika lain yang dapat diterapkan pada penjadwalan mesin produksi dengan NSGA-II.
3. Pengembangan lebih lanjut diperlukan untuk dapat menyelesaikan permasalahan penjadwalan mesin produksi yang lebih kompleks, seperti penjadwalan mesin produksi yang juga mempertimbangkan jenis multiobjektif yang lainnya.

## UCAPAN TERIMA KASIH

Penulis menyampaikan ucapan terima kasih sebesar-besarnya kepada Direktorat Riset dan Pengabdian Masyarakat, Direktorat Jenderal Penguatan Riset dan Pengembangan, Kementerian Riset, Teknologi, dan Pendidikan Tinggi. Penulis sekaligus sebagai peneliti mendapatkan pembiayaan dalam menyelesaikan penelitian dan tulisan ini.

## REFERENSI

- Balasundaram, R., Valavan, D. & Baskar, N. "Heuristic Based Approach for BI-Criteria Optimization of Minimizing Makespan and Total Flow Time of Flowshop Scheduling". *International Journal of Mechanical & Mechatronics Engineering IJMME-IJENS* Vol:14 No:02, 2014.
- Chakraborty, U.K. & Laha, D. An improved heuristic for permutation flowshop scheduling. *Int. J. Information and Communication Technology*, Vol. 1, No. 1, 2007.
- Choudhury, B.B., Mishra, D. & Biswal, B.B. "Task assignment and scheduling in a constrained manufacturing system using GA". *International Journal Agile System & Management (IJASM)*. 2007.
- Deb, K. & Tiwari, S. "Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization". *European Journal of Operational Research* 185 (2008) 1062–1087. Elsevier. 2008.
- Deb, K. "Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction". Department of Mechanical Engineering Indian Institute of Technology Kanpur, PIN 208016, India. KanGAL Report Number 2011003. Springer, 2011.
- Eva Instances. <http://www.upv.es/gio/rruiz>, diakses april 2015, 11.30 wib
- Ginting, R. 2009. Penjadwalan Mesin. Graha Ilmu. Yogyakarta.
- Josezefowicz, N., Semet, F. & Talbi, E. "Multi-objective vehicle routing problems". *European Journal of Operational Research* 189 (2008) 293–309. Elsevier. 2008.
- Mishra, S.K., Panda, G. & Meher, S. "Comparative Performance Evaluation of Multiobjective Optimization Algorithm For Portfolio Management". *Presented International Symposium on Biologically Inspired Computing and Application (BICA-2009)*. Bhubaneswar, India. December 2009.
- Ruiz, R., Minella, G. & Ciavotta, M. "A review and evaluation of multi-objective algorithms for the flowshop scheduling problem". *Grupo de Sistemas de Optimización Aplicada*. 29 Maret 2007
- Tyagi, N., Seidgar, H., Abedi, M. & Chandramouli, A.B. "Learning and Forgetting Effects of Flexible Flow shop Scheduling". *International Journal of Innovation and Applied Studies* . ISSN 2028-9324 Vol. 7 No. 3 Aug. 2014, pp. 857-867, 2014.
- Yagmahan, B. & Yenisey, M.M. 2010. "A multi-objective ant colony system algorithm for flow shop scheduling problem". *Journal Expert Systems with Applications*. Elsevier.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M. & Fonseca, G. 2002. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *TIK-Report No. 139*. Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich Gloriastrasse 35, ETH-Zentrum, CH-8092 Zürich, Switzerland.
- Zitzler, E. & Thiele, L. 1998. Multiobjective Optimization Using Evolutionary Algorithms-A Comparative Case Study. *Parallel Problem Solving from Nature PPSN V Amsterdam*. Page 292 301. Springer.

## BIBLIOGRAFI PENULIS



Fifin Sonata lahir di Banyuwangi, 24 Desember 1982. Menyelesaikan studi Magister (S2) di Fakultas Ilmu Komputer dan Teknologi Informasi (FASILKOM) Universitas Sumatera Utara. Saat ini bekerja sebagai Dosen S1 di Sekolah Tinggi Ilmu Komputer Medan dengan mengampu mata kuliah Analisa Perancangan Sistem Informasi, Struktur Data dan Pemrograman. Selain mengajar di kampus tersebut penulis juga mengajar sebagai Dosen freelance di Sekolah Tinggi Manajemen dan Informatika Komputer Triguna Dharma,. Selain itu penulis aktif mengikuti event yang diselenggarakan Pemerintah maupun swasta. Penulis merupakan copy editor pada Jurnal Kementerian Komunikasi dan Informatika Republik Indonesia (KOMINFO) Medan dan Koordinator Bidang Kerjasama pada Asosiasi Peneliti Sumatera Utara (APSU). Penulis juga aktif menulis pada beberapa jurnal diantaranya : Sistem Penjadwalan Mesin Produksi Menggunakan Algoritma Johnson dan Campbell, Jurnal Buana Informatika, Universitas Atma Jaya Yogyakarta; Optimasi Makespan dan Total Tardiness Dalam Penjadwalan Mesin Produksi Type Flow shop Menggunakan Metode *Non-Dominated Sorting Genetic Algorithm (NSGA-II)*, Jurnal Teknologi Informasi dan Komunikasi (JTIK), Kementerian Komunikasi dan Informatika Republik Indonesia (KOMINFO), Vol. 5/No. 1/2016 Juni 2016 p-ISSN: 2338-5839 e-ISSN: 2548-7582; Implementasi Metode *Simple Additive Weighting (SAW)* Dengan Proses Fuzzifikasi Dalam Penilaian Kinerja Dosen, Jurnal Teknologi Informasi dan Komunikasi (JTIK), Kementerian Komunikasi dan Informatika Republik Indonesia (KOMINFO), Vol. 5/No.2/2016 Desember 2016 p-ISSN: 2338-5839 e-ISSN: 2548-7582; Analisis Survei Faktor-Faktor Knowledge Sharing Dengan Teknik Focus Group Discussion (Fgd) Di Stikom Medan, Jurnal Teknologi Informasi dan Komunikasi (JTIK), Kementerian Komunikasi dan Informatika Republik Indonesia (KOMINFO), Vol.6 No. 1, Juni 2017 p-ISSN: 2338-5839 e-ISSN: 2548-7582; Pemanfaatan Metode Analytical Hierarchy Process (Ahp) Dalam Studi

	<p>Kelayakan Sarana Teknis Kereta Api Pada Pt. Kereta Api Indonesia Divisi Regional I Sumatera Utara Medan, Jurnal Pembangunan Perkotaan, Balitbang Pemerintah Kota Medan, Vol. 5 No. 2 (2017) Juli-Desember 2017 p-ISSN: 2338 6754 e- ISSN : 2581-1304; Strategi Penguatan Audit Tata Kelola Dan Manajemen Risiko Teknologi Informasi Pada Usaha <i>Startup</i> Menggunakan <i>Control Objective For Information And Related Technology Framework</i> (Studi Kasus : Otoritas Jasa Keuangan), Jurnal Komunikasi, Media dan Informatika, Kementerian Komunikasi dan Informatika Republik Indonesia (KOMINFO), Vol.7 No.1 (2018) April 2018 p-ISSN = 2303-1700 e-ISSN = 2579-5899; Audit Tata Kelola Dan Manajemen Risiko Teknologi Informasi Pada Infrastruktur Usaha <i>Start-Up</i> Oleh Pemerintah Kota Medan Menggunakan <i>Control Objective For Information And Related Technology (Cobit) Framework</i>, Jurnal Pembangunan Perkotaan, Balitbang Pemerintah Kota Medan, Vol. 6 No.1 2018 Januari-Juni 2018 p-ISSN: 2338-6754 e-ISSN : 2581-1304</p>
	<p>Dede Prabowo Wiguna lahir di Kisaran, 18 Juli 1989. Menyelesaikan studi Magister (S2) di Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Universitas Indonesia. Saat ini bekerja sebagai Dosen S1 di Sekolah Tinggi Ilmu Komputer Medan dengan mengampu mata kuliah Sistem Informasi Geografis, Statistika, Matematika Dasar, dan Metodologi Penelitian. Selain mengajardi kampus tersebut penulis juga sebagai mengajar Dosen freelance di Universitas Negeri Medan, Pelatih dan Juri di beberapa perlombaan yang diselenggarakan Pemerintah maupun swasta. Penulis pernah meraih beberapa penghargaan dari pemerintah bidang penulisan karya ilmiah dan penulis kreatif. Penulis juga aktif sebagai Pendiri dan Dewan Pembina di beberapa lembaga seperti Asosiasi Peneliti Sumatera Utara (APSU), Asosiasi Pembina Kelompok Ilmiah Remaja (APKIR) dan Asosiasi Pembina dan Pelatih Olimpiade Sains Sumatera Utara (APPOS).</p>