

PERANCANGAN SISTEM PENJADWALAN MESIN *HYBRID FLOW SHOP* DENGAN ALGORITMA *LEVYFLIGHT DISCRETE FIREFLY*

Andrew Verrayo Limas; J. Sudirwan; Siti Nur Fadlilah

Information Systems Department, School of Information Systems, Binus University
Jl. K.H. Syahdan No. 9, Palmerah, Jakarta Barat 11480
xdr3w@ymail.com; jsudirwan@yahoo.com; nurfadlilah@binus.ac.id

ABSTRACT

The main issues that have been encountered in PT Surya Toto Indonesia is companies do not know how to measure the performance of the production process so as the utilization of resources that will be used isn't yet to be known with efficiently. The purpose of this case studies is creating a better production processes by proposing a better and systematic scheduling information system. An Object Oriented Analysis and Design method is used to determine system's requirements and design of the system architecture. Metaheuristic methods such as Discrete algorithms Levy-flight Firefly is used to improve the performance of the Hybrid Flow Shop machine scheduling become more better. Indicators that are used to determine the performance of the scheduling is makespan and lateness. Results from this algorithm is makespan value of 305.27 hours with lateness value of 10.92 which determine that there is no delays in scheduling. Scheduling information system was designed three primary capabilities, which are integrated data management, inventory management and a systematic and accurate scheduling system.

Keywords: *hybrid flow shop, machine scheduling, object oriented analysis and design, levy-flight discrete firefly algorithm, makespan, lateness*

ABSTRAK

Permasalahan utama yang telah dihadapi di PT Surya Toto Indonesia adalah perusahaan belum mengetahui bagaimana mengukur performa dari proses produksi sehingga pengendalian sumber daya yang akan digunakan belum dapat diketahui dengan efisien. Tujuan studi kasus adalah membuat proses produksi menjadi lebih baik dengan mengusulkan sebuah sistem informasi penjadwalan yang lebih baik dan sistematis. Metode Object Oriented Analysis and Design digunakan untuk menentukan kebutuhan sistem dan perancangan arsitektur sistem. Metode metaheuristik berupa algoritma Levy-flight Discrete Firefly digunakan untuk meningkatkan performa penjadwalan mesin Hybrid Flow Shop menjadi lebih baik. Indikator yang digunakan untuk menentukan performa penjadwalan adalah makespan dan lateness. Hasil dari algoritma ini adalah nilai makespan sebesar 305,27 jam dengan nilai lateness 10,92 hari yang menandakan penjadwalan tidak mengalami keterlambatan. Sistem informasi penjadwalan dirancang dengan tiga kemampuan utama, yaitu manajemen data yang terintegrasi, manajemen inventori dan penjadwalan yang sistematis dan akurat.

Kata kunci: *penjadwalan mesin, hybrid flow shop, object oriented analysis and design, algoritma levy-flight discrete firefly, makespan, lateness*

PENDAHULUAN

Sistem manufaktur modern mengutamakan efisiensi dan efektivitas di rantai produksi dan salah satu aspek yang mampu meningkatkan hal ini adalah peningkatan pada kinerja *output* produksi (Kaban, Othman, & Rohmah 2012), oleh karena itu, pengaturan pada penjadwalan setiap mesin harus dilakukan secara efisien dan efektif (Liao, *et al.*, 2013). Penjadwalan yang dimaksud adalah proses menyusun urutan dari proses produksi harus menjadi lebih baik dengan tujuan agar sumber daya yang digunakan menjadi lebih sedikit dibanding sebelumnya (Righi, 2012).

PT Surya Toto Indonesia, Tbk atau PT STI merupakan perusahaan yang bergerak di bidang manufaktur barang dengan jenis *sanitary wares*, *kitchen set* dan *plumbing fitting*. Dalam perkembangannya di bidang manufaktur, perusahaan ini mengawali produksinya sejak 11 Juli 1977 dan telah menerima beberapa sertifikasi internasional baik dari Jepang atau negara lain. Hingga saat ini, PT STI telah memiliki tiga lokasi yang tersebar di beberapa wilayah Jabodetabek.

Studi kasus ini berfokus pada pabrik kedua yang berada di Serpong dengan luas 58.256 m² yang memproduksi badan (*body*) dari setiap barang. Permasalahan utama pada pabrik kedua ini adalah perusahaan belum mengetahui bagaimana cara mengukur performa dari proses produksi sehingga penggunaan sumber daya yang digunakan belum dapat diketahui dengan pasti. Salah satu solusi yang ingin dikaji dalam studi kasus ini adalah bagaimana mengendalikan penggunaan sumber daya ini dengan mengusulkan sebuah sistem penjadwalan yang lebih baik dan sistematis menggunakan metode-metode yang diusulkan dengan tujuan agar proses produksi menjadi lebih baik dan didukung oleh sistem informasinya.

Model penjadwalan yang diusulkan pada PT STI adalah model penjadwalan *hybird flow-shop* (HFS) yang merupakan pengembangan dari model penjadwalan *flow-shop*. Dalam model ini, semua pekerjaan secara garis besar harus melewati semua urutan mesin. Namun di setiap urutan, dibagi lagi menjadi beberapa mesin yang bekerja secara paralel (Şerifoğlu & Ulusoy, 2004). Penjadwalan pada dasarnya memiliki beberapa indikator atau fungsi objektif dan salah satunya yang dapat mencerminkan optimalisasi sumber daya adalah minimalisasi waktu total pengerjaan proses produksi atau *makespan* (Marichelvam, Prabaharan, & Yang, 2012). Pada kurun tiga tahun terakhir saja, *makespan* telah digunakan sebanyak 60% peneliti di seluruh dunia (Ruiz, Antonio, & Vázquez-Rodríguez, 2009). Digunakan juga indikator maksimal keterlambatan (*lateness*) sebagai indikator kedua untuk menggambarkan performa penjadwalan yang lebih baik.

Dalam mencari solusi terbaik dengan fungsi objektif *makespan*, digunakan metode *meta-heuristik* berupa algoritma *Lévy-Flight Discrete Firefly* (L DFA) yang merupakan algoritma *nature-inspired* terbaik untuk memecahkan masalah optimalisasi (Sayadi, Ramezani, & Ghaffari-Nasab, 2010). Sesuai dengan latar belakang permasalahan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut: (1) Apakah algoritma Levy-flight Discrete Firefly dapat dan cocok diaplikasikan pada sistem penjadwalan? (2) Bagaimana rancangan suatu sistem penjadwalan yang mampu mengendalikan penggunaan sumber daya, dengan algoritma Levy-flight Discrete Firefly? (3) Apakah keuntungan yang dimiliki oleh sistem penjadwalan jika diterapkan pada proses bisnis perusahaan?

METODE

Untuk melaksanakan penelitian ini, sebelumnya dikaji konsep dan model-model yang relevan dengan masalah penelitian dan konsep dasar serta model yang digunakan untuk menjawab dan memecahkan masalah. Konsep dan model dasar yang dianggap penting dalam yaitu yang berhubungan dengan model-model dalam penjadwalan dan praktikal model atau prosedur dalam pembangunan sistem informasi yang mengaplikasikan model-model matematis dalam penjadwalan.

Hybrid Flow-Shop

Merupakan ekstensi dari model penjadwalan *general flow-shop* dan permasalahan penjadwalan paralel di mana semua pekerjaan akan melewati tahap produksi yang sama dengan beberapa mesin di setiap tahap (Righi, 2012). Pada model penjadwalan HFS ini, setiap tahap produksi memiliki satu mesin atau lebih yang bekerja secara paralel (Marichelvam, Prabaharan, & Yang, 2012). Fungsi objektif ini telah digunakan sebanyak 60% pada akhir-akhir ini (Ruiz, Antonio, & Vázquez-Rodríguez, 2009). Rumus dari *makespan* tersebut adalah sebagai berikut:

$$C_{max} = \max(C_j)$$

Dimana:

C_{max} = nilai *makespan*

C_j = waktu total pekerjaan j , dimana $j = 1, 2, 3, \dots, n$

Lévy-Flight Discrete Firefly

Merupakan metode *meta-heuristik* berupa algoritma yang digunakan untuk memecahkan masalah optimalisasi. Algoritma ini terinspirasi oleh metode pergerakan suatu kunang-kunang dalam mencari pasangan (Yang, 2010). Cara ini kemudian dirumuskan dalam tiga prinsip dasar agar dapat digunakan untuk pemecahan masalah optimalisasi, yaitu: (1) Semua kunang-kunang adalah *unisex* sehingga semua kunang-kunang dapat tertarik satu sama lain. (2) Daya tarik kunang-kunang proporsional dengan terang dari kunang-kunang, di mana nilai dari daya tarik berbanding terbalik dengan nilai jarak. Kunang-kunang akan tertarik pada kunang-kunang yang lebih terang. (3) Intensitas cahaya dari kunang-kunang merupakan nilai fungsi objektif dari suatu permasalahan.

Perancangan algoritma LDFA ini menggunakan *pseudo-code* sebagai berikut (Yang, 2010):

Tabel 1 *Pseudo-code* LDFA

```
begin
  Objective function  $f(x)$ ,  $x = (x_1, \dots, x_n)T$ 
  Generate initial population of fireflies  $x_i$  ( $i = 1, 2, \dots, n$ )
  Light intensity  $I_i$  at  $x_i$  is determined by  $f(x_i)$ 
  Define light absorption coefficient  $\gamma$ 

  while ( $t < \text{MaxGeneration}$ )

    for  $i = 1 : n$  all  $n$  fireflies

      for  $j = 1 : i$  all  $n$  fireflies

        if ( $I_j > I_i$ )
          Move firefly  $i$  towards  $j$  in  $d$ -dimension via Lévy flights
        end if
        Attractiveness varies with distance  $r$  via  $\exp[-\gamma r]$ 
        Evaluate new solutions and update light intensity
      end for  $j$ 
    end for  $i$ 
    Rank the fireflies and find the current best

  end while
  Postprocess results and visualization
end
```

Tiga prinsip tersebut dijelaskan pada tiga konsep utama yang digunakan dalam menjalankan algoritma ini, yaitu (Yang, 2010):

Intensitas Cahaya

Intensitas cahaya pada kunang-kunang merupakan indikator yang digunakan untuk mengetahui apakah kunang-kunang tersebut bergerak ke kunang-kunang berikutnya atau tidak. Pada fungsi minimalisasi *makespan* pada penjadwalan ini, nilai dari intensitas cahaya kunang-kunang menggunakan rumus sebagai berikut:

$$I_i = \frac{f}{f(x)} \times 100000$$

Di mana:

I_i = intensitas cahaya pada kunang-kunang i , dimana $i = 1, 2, 3, \dots, n$
 $f(x)$ = fungsi objektif

Jarak

Setiap jarak ini akan direpresentasikan dalam bentuk tabel *Cartesian*. Jarak dari kunang-kunang i dan j dapat dicari dengan menggunakan rumus sebagai berikut:

$$r_{ij} = \sqrt{\sum_{k=1}^n (x_{i,k} - x_{j,k})^2}$$

Di mana:

r_{ij} = jarak antara kunang-kunang i dan j
 $x_{i,k}$ = nilai dimensi k pada kunang-kunang i
 $x_{j,k}$ = nilai dimensi k pada kunang-kunang j

Pergerakan

Pergerakan dilakukan dengan prinsip dimana kunang-kunang yang lebih redup akan bergerak ke kunang-kunang yang lebih terang. Berbeda dengan algoritma kunang-kunang pada dasarnya, LDFA menggunakan distribusi *Lévy* untuk mengatur laju pergerakan. Rumus dari pergerakan tersebut adalah sebagai berikut:

$$x'_i = x_i + \beta_0 e^{-\gamma t} (x_j - x_i) + \left(\alpha \left(\text{rand} - \frac{1}{2} \right) \oplus \text{Lévy} \right)$$

$$\text{Lévy} \sim u = t^{-\lambda}$$

Di mana:

x'_i = posisi kunang-kunang i yang baru
 x_i = posisi kunang-kunang i
 β_0 = daya tarik pada posisi 0
 γ = koefisien penyerapan cahaya, $0.01 < \gamma < 100$
 α = koefisien bilangan acak, $0 < \alpha < 1$
 t = jumlah iterasi, $1 < t < \infty$
 λ = konstanta lambda, $1 < \lambda < 3$
 r_{ij} = jarak antara kunang-kunang i dan j

Object Oriented Analysis & Design

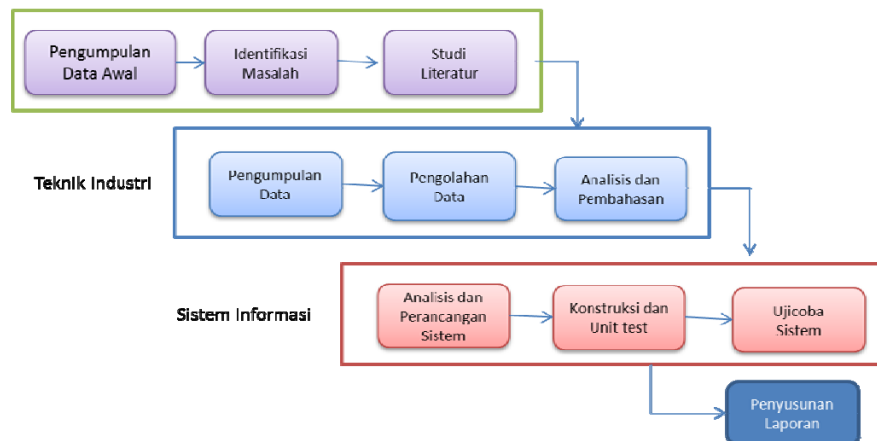
Metode analisis dan pengembangan berbasis objek atau OOAD merupakan salah satu proses pengembangan sistem informasi yang merupakan kumpulan dari beberapa model *Unified Modeling Language* (UML) serta berorientasi pada objek dan digunakan untuk menghubungkan antara kebutuhan sistem dari para pengguna dengan para perancang sistem (Satzinger, Jackson, & Burd, 2008). Dalam mengembangkan konsep OOAD ini, berikut beberapa *tools* yang akan digunakan dalam menganalisis dan merancang sistem informasi (Satzinger, Jackson, & Burd, 2008):

Tabel 2 *Tools* OOAD

<i>Requirement Analysis</i>	<i>Design System</i>
<ul style="list-style-type: none"> • <i>Activity Diagram</i> • <i>Event Table</i> • <i>Use Case Diagram</i> • <i>Use Case Description</i> • <i>Domain Class Diagram</i> • <i>Activity-Data Matrix</i> • <i>State Transition Diagram</i> • <i>System Sequence Diagram</i> • <i>Storyboarding</i> 	<ul style="list-style-type: none"> • <i>First-Cut Class Diagram</i> • <i>Deployment Environment</i> • <i>Software Architecture</i> • <i>Completed Three-Layer Sequence Diagram</i> • <i>Updated Design Class Diagram</i> • <i>Package Diagram</i> • <i>Interface Design Standards</i>

Prosedur Penelitian

Untuk melaksanakan studi kasus ini, berikut adalah diagram alir yang digunakan:



Gambar 1 Metode Penelitian

Aktivitas pertama berhubungan dengan mengumpulkan beberapa data seperti topik permasalahan yang akan dibahas dan masalah optimisasi proses penjadwalan produksi akan dijadikan sebagai permasalahan utama. Setelah pengumpulan data awal dilakukan, salah satu masalah yang dihadapi pada perusahaan adalah bagaimana mengetahui penggunaan sumber daya melalui pengaturan pada proses penjadwalan menjadi lebih baik. Masalah ini akan dijadikan sebagai masalah utama dan beberapa rumusan permasalahan akan ditentukan.

Aktivitas berikutnya adalah mengenali lingkungan operasional perusahaan, mencari literatur mengenai penggunaan algoritma *Lévy-Flight Discrete Firefly* (LDFA) dalam memecahkan masalah penjadwalan serta beberapa buku yang memberikan referensi mengenai bagaimana merancang sistem penjadwalan yang fokus pada peningkatan proses bisnis perusahaan. Pengumpulan data dilaksanakan pada bulan Januari sampai dengan bulan Februari 2013. Dua kategori pengumpulan data secara mendasar, yaitu pengumpulan data pada sejarah dan proses bisnis perusahaan. Kedua adalah pengumpulan data produksi yang merupakan data sekunder seperti waktu standar dari setiap mesin dan data produksi pada setiap barang.

Pengolahan data dilakukan dengan menggunakan algoritma LDFA untuk mengoptimalkan penjadwalan produksi. Hasil akhir dari pengolahan data adalah urutan penjadwalan proses produksi dengan waktu total pengerjaan proses produksi (*makespan*) yang lebih baik. Analisis pertama

dilakukan pada hasil dari algoritma LDFA dalam mengatur urutan penjadwalan proses produksi. Indikator analisisnya berupa maksimum keterlambatan (*lateness*) yang diperoleh dari hasil *makespan* melalui total iterasi yang telah dilakukan. Analisis terakhir dilakukan pada kebutuhan dari sistem informasi penjadwalan yang dirangkum dalam *Requirement Analysis* dan *Design System sistem informasi*.

Sistem yang akan dirancang berupa sistem yang dapat mendukung pemecahan masalah utama dalam proses penjadwalan perusahaan. Sistem ini menggunakan konsep sistem informasi yang telah dikumpulkan pada studi literatur sebelumnya. Beberapa fungsi usulan yang akan ada pada sistem ini dirancang melalui metode *Object Oriented Analysis and Design* (OOAD) sehingga mampu menghasilkan kemampuan-kemampuan sistem yang akan mendukung proses bisnis. Rancangan dimulai dari rancangan Arsitektur hingga rancangan rinci berupa spesifikasi dari komponen sistem yang digunakan untuk membangaun atau konstruksi dari piranti lunak sistem, dan secara individu akan diuji coba sebelum ujicoba sistem secara keseluruhan.

Pengujian sistem akan dilakukan pada akhir konstruksi dan beberapa tolak ukur utamanya seperti keakuratan dari proses dalam sistem, kecepatan proses pengolahan data, dan penggunaan antarmuka yang sesuai. Dari hasil uji coba diperoleh jawaban dari semua pertanyaan dalam penelitian. Dari keseluruhan proses atau aktivitas, dengan semua hasil diperoleh dan selesai maka laporan penelitian atau studi dalam berbagai bentuk untuk publikasi penelitian.

HASIL DAN PEMBAHASAN

Aturan SPV

Dalam mengimplementasikan algoritma LDFA dalam masalah penjadwalan proses produksi, digunakan algoritma kunang-kunang berbentuk diskrit. Bentuk diskrit dari algoritma ini diperoleh melalui implementasi aturan *Smallest Position Value* (SPV) pada representasi urutan permutasi sebagai bentuk dari solusi penjadwalan. Aturan SPV dengan contoh tujuh urutan proses produksi ini dapat diilustrasikan pada tabel berikut:

Tabel 3 Solusi Representasi Untuk Sebuah Kunang-kunang

	Dimensi (<i>j</i>)						
	1	2	3	4	5	6	7
Nilai Posisi (x_{ij})	0,911	0,738	0,224	0,592	0,03	0,633	0,862
Urutan Permutasi	5	3	4	6	2	7	1

Dari tabel 3, setiap dimensi menandakan satu urutan proses produksi. Nilai x_{ij} terkecil pertama berada pada dimensi $j = 5$ sehingga urutan permutasi pertama adalah 5 (lima), dan berikut seterusnya. Proses SPV ini akan dilakukan sebanyak jumlah dimensi sehingga terbentuk satu kunang-kunang yang merepresentasikan satu kandidat solusi. Jika diartikan dalam urutan penjadwalan, maka urutan produksi akan dimulai dari komponen kelima, ketiga dan seterusnya hingga komponen kesatu .

Pergerakan Kunang-Kunang

Pergerakan pada kunang-kunang dilakukan jika intensitas antara satu kunang-kunang lebih besar dibanding kunang-kunang yang berada paling dekat. Pergerakan ini menggunakan rumus dari algoritma LDFA yang akan dilakukan pada setiap dimensi dalam nilai posisi kunang-kunang (x_{ij}). Sebagai ilustrasi, berikut contoh perhitungan dari pergerakan kunang-kunang:

Tabel 4 Perhitungan Pergerakan Dua Kunang-kunang

Kunang-Kunang (x_i)	Dimensi (j)				
x_1	0,203	0,573	0,032	0,134	0,887
x_2	0,293	0,116	0,613	0,673	0,934
x_1'	0,585	0,955	0,414	0,516	1,269

Skenario pada tabel 4 adalah kunang-kunang kedua memiliki intensitas cahaya lebih terang sehingga kunang-kunang pertama yang bergerak ke kunang-kunang kedua. Contoh perhitungan pada x_1' pada dimensi $j = 1$ adalah sebagai berikut (Contoh nilai $rand = 0,734$):

$$x_{1j} = 0,203 + (((0,5 \times e^{0,734 \times 0,203^2}) \times 0,293 - 0,203) + (0,5 \times (rand - 0,5) \times 1^{-1,5}))$$

$$x_{1j} = 0,203 + 0,337 + 0,117$$

$$x_{1j} = 0,677$$

Perhitungan ini akan dilanjutkan sampai dengan dimensi ke- n dan hasil x_{ij} , $j = 1, 2, 3, \dots, k$; $k =$ jumlah dimensi, akan digunakan sebagai nilai dari posisi baru bagi kunang-kunang pertama. Berikutnya akan digunakan aturan SPV untuk mengubah setiap dimensi dalam nilai posisi kunang-kunang (x_{ij}) menjadi bentuk urutan permutasi sehingga terbentuk kandidat solusi yang baru.

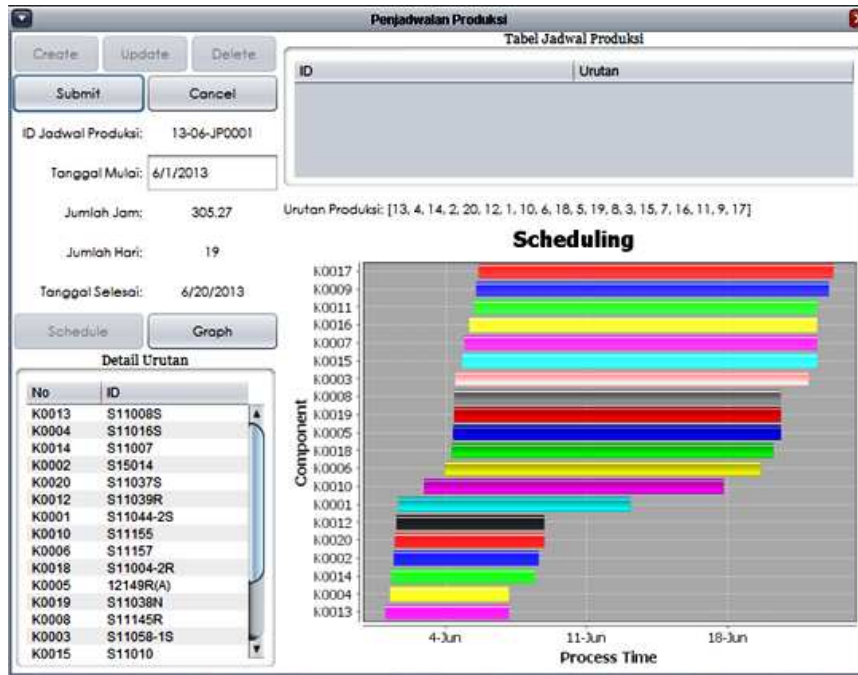
Implementasi Algoritma LDFA

Implementasi algoritma LDFA ini terdiri dari dua proses utama, yaitu penentuan parameter yang digunakan dan tampilan antarmuka sebagai gambaran dari hasil algoritma untuk proses penjadwalan dengan data yang tersedia. Parameter yang digunakan pada algoritma LDFA ini adalah (Marichelvam, Prabaharan, & Yang, 2012):

Tabel 5 Pengaturan Parameter Algoritma LDFA

Simbol	Keterangan	Nilai	Simbol	Keterangan	Nilai
n	Jumlah Populasi	20	β_0	Konstanta Ketertarikan	0,5
t	Jumlah Iterasi	15	γ	Konstanta <i>Gamma</i>	0,75
α	Konstanta <i>Alpha</i>	0,5	λ	Konstanta <i>Lévy</i>	1,5

Semua perhitungan ini dijalankan pada komputer dengan spesifikasi Intel Core i7 3,4 GHz CPU dengan 8 GB DDR3 RAM dan dirancang dengan menggunakan bahasa pemrograman *Java* 1,7 dengan *Netbeans* versi 7,3. Adapun beberapa asumsi yang digunakan pada implementasi algoritma dalam mengatur penjadwalan ini, yaitu sebagai berikut: (1) Semua waktu mulai pengerjaan (*release date*) komponen dilakukan pada awal bulan dan tenggat waktu (*due date*) untuk setiap komponen adalah sama, yaitu akhir bulan dari pengerjaan proses produksi. (2) Satu tahap produksi dapat berisi satu mesin atau lebih yang memiliki karakteristik dan proses yang sama. (3) Untuk setiap tahap produksi, dua komponen tidak bisa diproses dalam waktu yang bersamaan. (4) Waktu pengaturan mesin sudah termasuk pada waktu standar dari proses pengerjaan setiap komponen. (5) Waktu standar setiap komponen dalam setiap proses telah diketahui dan tetap untuk setiap penjadwalan. (6) Waktu pemindahan barang untuk setiap tahap produksi diabaikan. (7) Data produksi yang digunakan adalah data hasil peramalan dan jumlah dari komponen disesuaikan berdasarkan kasus.



Gambar 2 User Interface Penjadwalan Produksi

Gambar 2 memperlihatkan *makespan* dengan nilai sebesar 305,27 jam atau 19,08 hari. Nilai hari ini ditentukan dengan membagi jumlah jam dengan jumlah jam kerja total dari standar waktu kerja di PT STI, yaitu 16 jam tanpa adanya lembur. Karena *makespan* yang diperoleh merupakan hasil dari waktu total pengerjaan proses produksi yang dilihat dari setiap komponen, maka dengan menggunakan rumus (2.16), dapat dihitung nilai keterlambatannya (*lateness*):

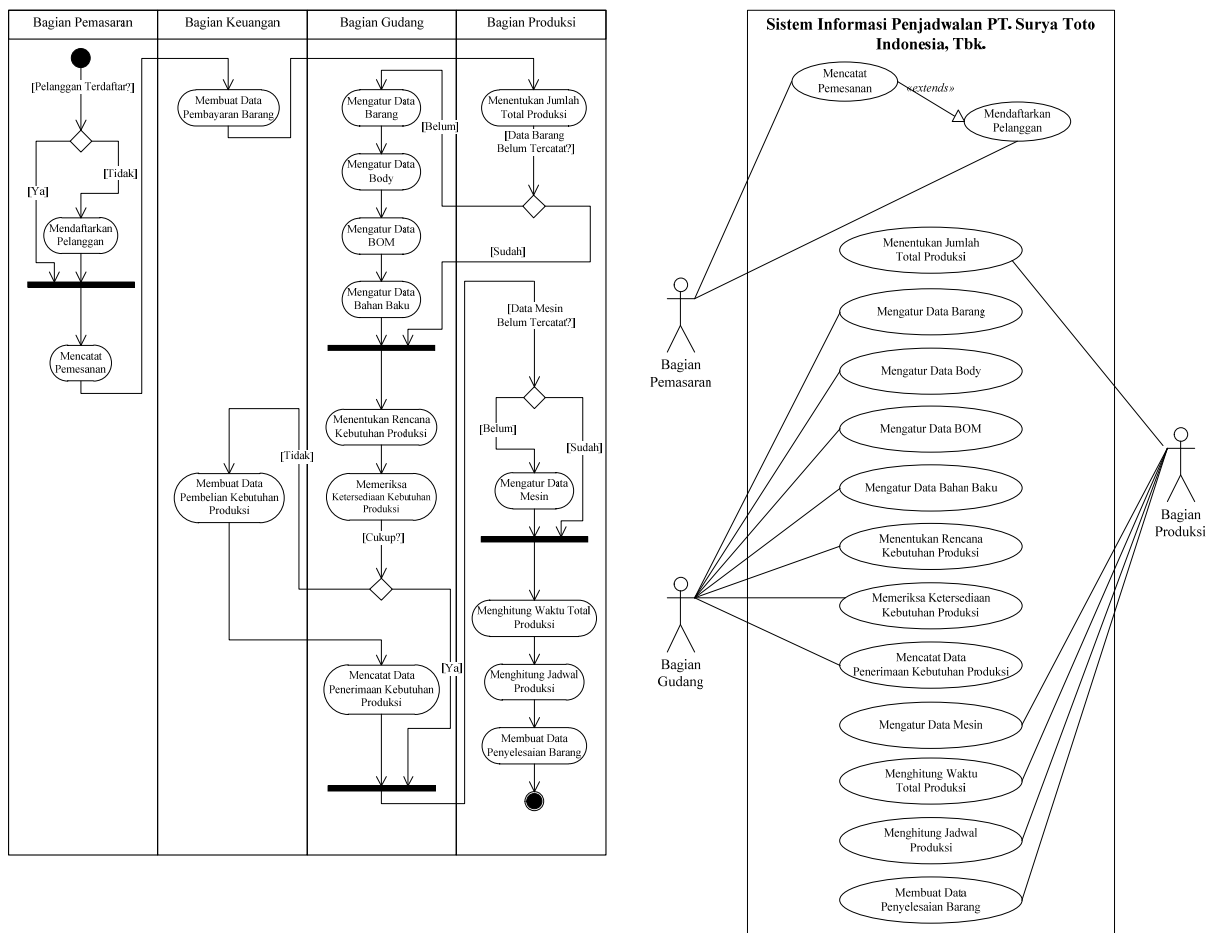
$$L_{max} = \max (C_j - d_j)$$

$$L_{max} = (19,08 - 30)$$

$$L_{max} = -10,92$$

Analisis Sistem Informasi

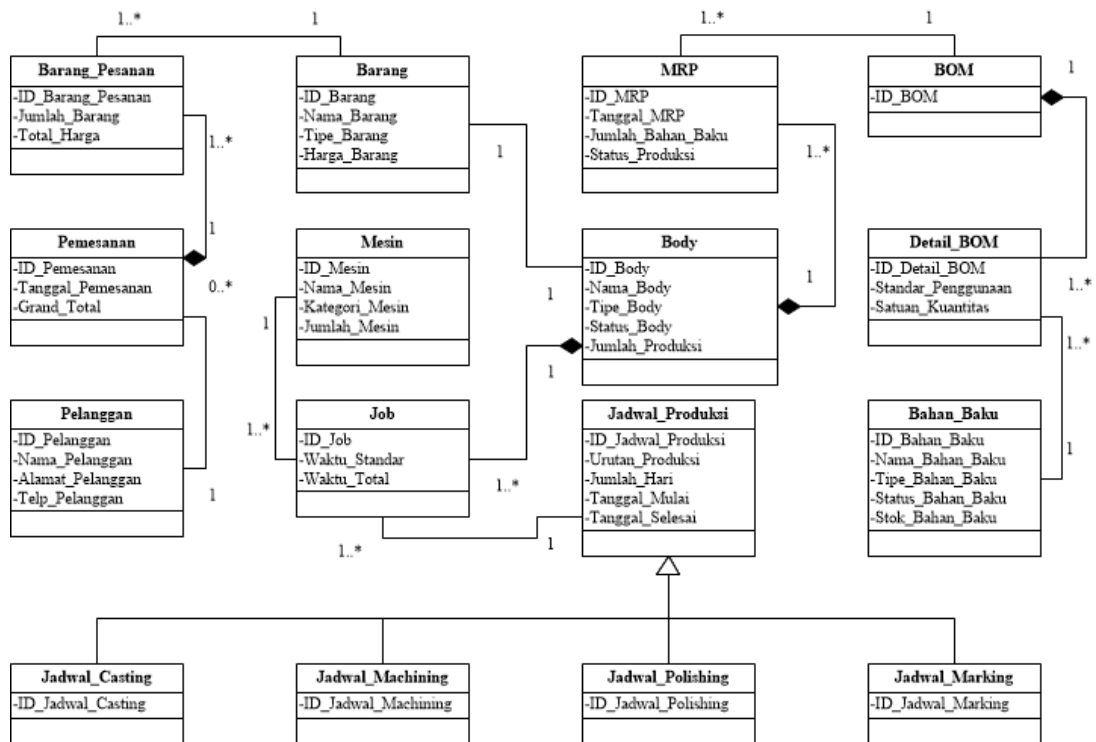
Analisis pada perancangan sistem informasi proses penjadwalan dimulai dengan menentukan kebutuhan dari sistem. Kebutuhan ini berupa komponen-komponen dalam sistem yang akan dianalisa menggunakan model UML. Komponen-komponen ini seperti fungsi, detail dari setiap fungsi, aktor dalam sistem dan arsitektur data. Tahap-tahap analisis dilakukan dengan menentukan fungsi dalam sistem. Kemudian dilakukan analisis terhadap data *input* dan *ouput* dari setiap fungsi. Analisis berikutnya adalah menentukan siapa yang akan bertanggung jawab terhadap fungsi-fungsi ini. Berikutnya adalah menentukan arsitektur data dari kriteria *input* dan *output* dari setiap fungsi. Analisis yang penting berasal dari dua diagram di bawah ini:



Gambar 3 Diagram Aktivitas dan Diagram Use Case

Tujuan diagram aktivitas adalah untuk memberikan gambaran keseluruhan terhadap proses bisnis di dalam perusahaan. Analisis dilakukan dengan mengobservasi beberapa aktivitas utama dalam proses bisnis perusahaan khususnya proses penjadwalan. Tujuan dari diagram use case adalah untuk memetakan fungsi yang berasal dari tabel dan diagram aktivitas menjadi terstruktur sehingga sistem mengetahui secara pasti siapa yang bertanggung jawab untuk melakukan fungsi tersebut.

Diagram berikutnya pada gambar 4 digunakan untuk menjelaskan struktur data dan asosiasi antar data yang akan digunakan untuk perancangan arsitektur sistem sesuai dengan tabel aktivitas yang telah dijelaskan. Diagram ini penting untuk perancangan sistem karena setiap data memiliki atribut yang merupakan informasi untuk sistem.

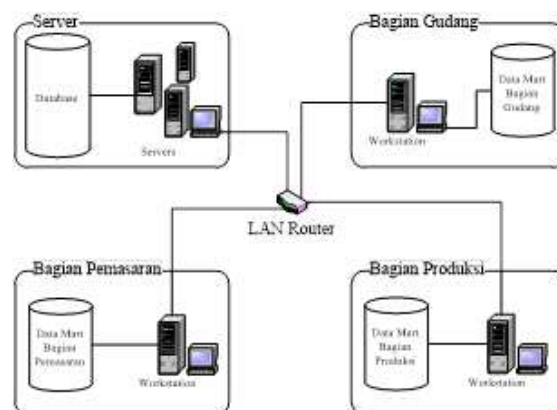


Gambar 4 Diagram *Domain Class*

Desain Sistem Informasi

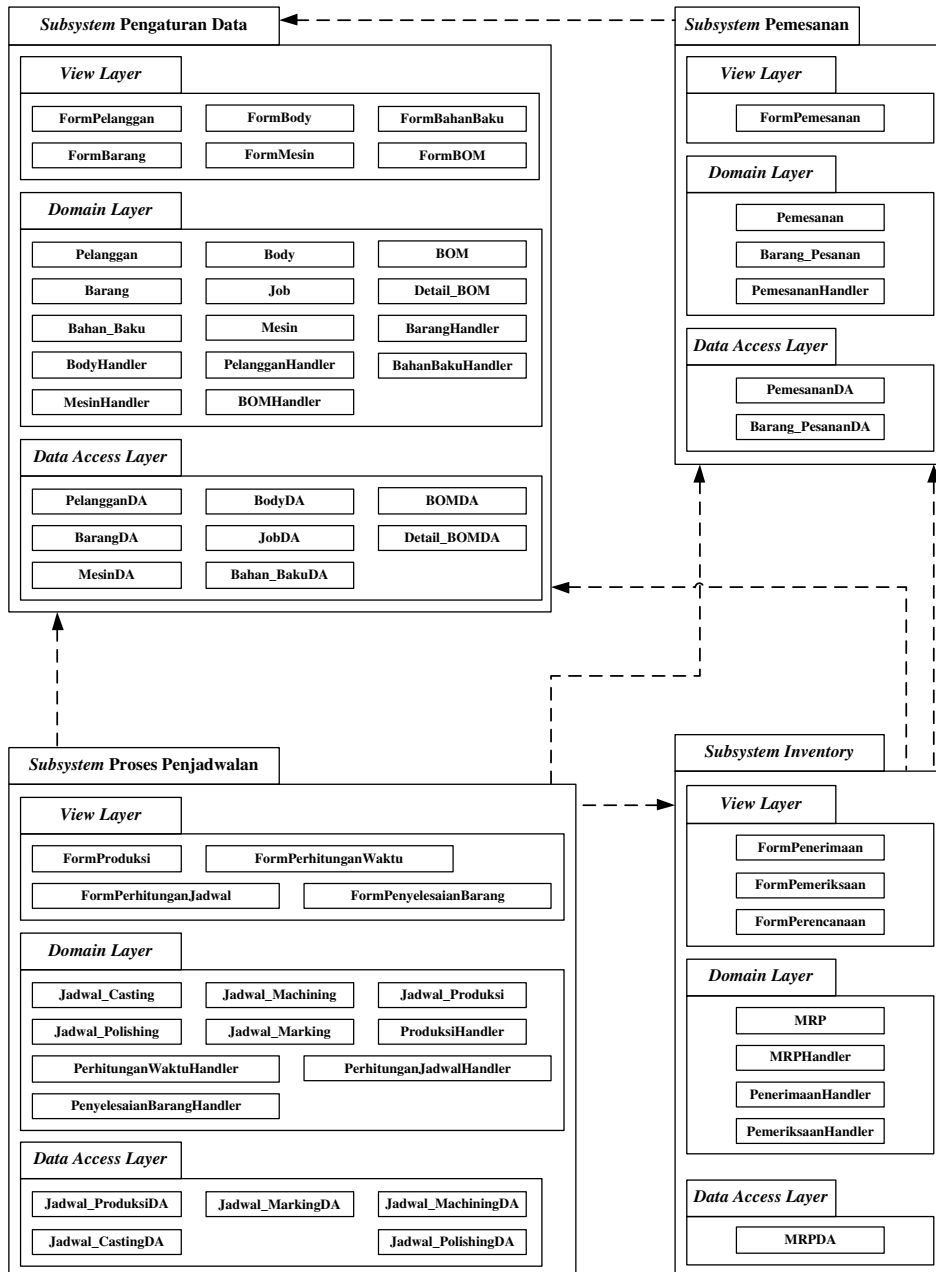
Desain dimulai dengan mengembangkan navigasi data dari setiap arsitektur dalam sistem. Setelah itu, ditentukan standar dari spesifikasi teknologi informasi dan arsitektur jaringan dari sistem. Kemudian, dilakukan perancangan pada urutan proses dari setiap fungsi sistem serta arsitektur data yang telah dilengkapi dengan operasi-operasi untuk setiap objeknya. Terakhir adalah menggambarkan hubungan dari setiap subsistem. Hasil dari perancangan desain ini berupa sistem yang memiliki fungsi-fungsi yang telah ditentukan pada analisis kebutuhan sistem.

Arsitektur dari sistem informasi menggunakan jenis arsitektur *client* dan *server*. *Server* akan digunakan untuk menyimpan *database* utama. Sedangkan untuk setiap *client*-nya, digunakan *data mart* sebagai penyimpanan sementara. Berikut adalah gambar dari struktur hubungan antara *server* dengan *client*:



Gambar 5 Arsitektur Perangkat Lunak

Hasil akhir dari perancangan sistem ini adalah diagram *subsystem package*. Tujuan dari diagram ini adalah menggambarkan hubungan keseluruhan dari semua lapisan atau *layer* di dalam sistem. Secara singkat, *layer* ini dibagi menjadi tiga, yaitu *view*, *domain* dan *data access layer*. Berikut adalah struktur diagram yang telah dibagi menjadi empat subsistem:



Gambar 6 Diagram *Sub-system Package*

Pengujian Sistem Informasi

Setelah dilakukan perancangan pada arsitektur dan konstruksi pada sistem, maka dilakukan pengujian pada sistem informasi proses penjadwalan dengan menggunakan urutan dari proses bisnis dan data dari perusahaan. Hasil dari pengujian sistem tersebut berupa hal-hal yang dapat dicapai dengan adanya sistem dalam mendukung proses bisnis dalam perusahaan.

Hal-hal tersebut dapat dijelaskan pada poin-poin di bawah ini: (1) Manajemen Data yang Terintegrasi – Perubahan yang dilakukan pada data *master* akan mengubah data-data yang digunakan pada subsistem lainnya. Keseluruhan data ini kemudian disimpan pada *database Access* dan sesuai dengan arsitektur *software*, digunakan *data mart* untuk menyimpan data yang akan disinkronisasi untuk memastikan agar data tetap terintegrasi. (2) Manajemen *Inventory* – Dengan subsistem *inventory* yang dirancang dalam sistem ini, perusahaan dapat mengetahui informasi bahan baku berupa penggunaan stok bahan baku pada tanggal tertentu. Dengan ini, perusahaan dapat mengatur lebih lanjut stok bahan baku yang digunakan pada bulan berikutnya. (3) Penjadwalan yang Sistematis dan Akurat – Dalam subsistem proses penjadwalan, kegiatan penjadwalan dilakukan secara sistematis. Kegiatan ini dimulai dari menentukan jumlah produksi yang dibutuhkan untuk setiap komponen dari barang yang akan diproduksi kemudian menentukan waktu total dari setiap proses pengerjaan barang. Selanjutnya, melakukan penjadwalan dengan metode metaheuristik yang diusulkan. Agar penjadwalan yang dilakukan akurat, digunakan indikator yaitu waktu total pengerjaan (*makespan*) dan maksimal keterlambatan (*lateness*). Terakhir adalah menampilkan waktu penyelesaian dari setiap kategori mesin agar bagian produksi dapat memantau hal ini pada kegiatan produksi.

SIMPULAN

Digunakan algoritma *Levy Flight Discrete Firefly* (LDFA) yang berfungsi untuk mencari solusi berupa urutan produksi dengan tujuan agar penjadwalan yang direncanakan dapat mengendalikan sumber daya. Pengendalian ini dilakukan dengan mengatur *makespan* sehingga didapat nilai terbaik sebesar 19 hari. Dengan pengendalian ini, perusahaan memiliki keuntungan manajemen yang dapat digunakan untuk mengendalikan sumber daya lainnya, contohnya seperti tenaga kerja, penggunaan mesin dan sebagainya. Tujuannya adalah untuk memberikan opsi-opsi teknis yang memungkinkan perusahaan untuk membuat keputusan yang lebih baik.

Guna menciptakan proses penjadwalan yang sistematis dan akurat, maka ada dua tahapan yang harus dilakukan. Pertama, menggunakan sistem informasi yang telah dirancang sehingga proses penjadwalan dapat dilakukan secara bertahap. Tahapan ini dimulai dari penentuan jumlah total produksi hingga penentuan urutan berupa jadwal produksi barang. Dengan adanya proses yang sistematis ini, beberapa keuntungan yang dapat diperoleh seperti penghematan dari tenaga kerja itu sendiri dan pastinya waktu yang digunakan untuk merencanakan dan melakukan validasi dari model yang direncanakan secara manual pastinya akan berkurang. Kedua, diberikan sebuah proses penjadwalan yang akurat, di mana ada dua indikator, yaitu *makespan* dan nilai keterlambatan atau *lateness*. Dengan dua indikator ini, pihak perusahaan mampu mengetahui bagaimana performa dari penjadwalan. Dengan pengetahuan ini, maka tindakan *monitoring* pada proses produksi dapat dijelaskan secara fakta sesuai dengan informasi dari sistem penjadwalan ini.

Melalui pengujian sistem informasi yang dilakukan, diperoleh beberapa keuntungan yaitu perusahaan dapat melakukan pengaturan data yang terintegrasi satu sama lain. Hal ini dicapai melalui adanya penggunaan teknologi *database management system* dan sistem jaringan arsitektur *client* dan *server*. Dengan integritas ini, maka pihak perusahaan mampu melihat pergerakan dari data di dalam sistem. Manajemen sederhana pada *inventory* dapat dilakukan seperti mengetahui informasi-informasi terkait bahan baku. Salah satu poin penting dari manajemen ini adalah memberikan sebuah pedoman bagi proses pengaturan inventori dari sisi teoritis kepada perusahaan. Terakhir adalah sistem memungkinkan perusahaan untuk melakukan penjadwalan yang sistematis melalui sistem penjadwalan yang terstruktur dan mengetahui informasi terkait penjadwalan berupa performa dari penjadwalan itu sendiri dengan indikator yang jelas. Dengan adanya keunggulan-keunggulan ini, maka perusahaan memiliki kekuatan dalam melakukan pengaturan pada bagian perencanaan produksi, khususnya di bidang penjadwalan.

Pengembangan lebih lanjut pada sistem penjadwalan dapat dilakukan pada tiga hal. Pertama adalah dengan mengembangkan sistem inventori menjadi lebih bagus dengan menggabungkan proses perencanaan produksi. Kedua dengan mengimplementasikan subsistem penjadwalan dari perakitan setiap komponen menjadi barang jadi (*finished goods*). Ketiga, mengembangkan subsistem pemeliharaan mesin sehingga proses penjadwalan produksi tidak perlu memikirkan apabila satu atau dua mesin menjadi rusak. Keuntungan dari melakukan saran ini adalah untuk menciptakan suatu sistem yang lebih komprehensif dan menyeluruh bagi perusahaan dalam menciptakan proses bisnis yang lebih handal dan cepat tanggap. Jika saran ini dapat diterapkan oleh perusahaan pada masa depan, maka perusahaan mampu mengurangi resiko dalam hal integritas proses, yang di mana pada masa depan, volume produksi dapat saja meningkat. Di sisi lain, dengan pengembangan-pengembangan sistem ini, perusahaan mampu mengurangi resiko adanya *human error* dalam perencanaan yang biasanya dilakukan oleh beberapa orang.

Pengembangan algoritma dapat dilakukan dengan menggabungkan metode meta-heuristik ini dengan metode heuristik sebagai inisialisasi awal untuk populasi dari algoritma LDFA untuk meningkatkan efisiensi waktu dari pencarian solusi dengan algoritma ini. Tidak ada resiko yang berarti bagi perusahaan jika tidak menerapkan saran ini, namun saran ini merupakan opsional karena hal ini tidak mengubah hasil solusi secara signifikan namun pada waktu pemrosesan algoritma yang dimana hal ini merupakan bukan hal yang penting. Namun saran ini merupakan saran bagi para pembaca yang ingin melakukan penelitian atau studi kasus yang membutuhkan kecepatan dalam hal waktu.

Untuk mendukung lebih lanjut performa *output* dari setiap permasalahan yang dipecahkan oleh algoritma LDFA, diperlukan fungsi objektif yang terdiri dari dua jenis atau lebih. Hal ini penting bagi keuntungan perusahaan terutama untuk meningkatkan kualitas dari solusi, namun beberapa hal yang harus dilakukan adalah melakukan analisis terhadap fungsi-fungsi objektif manakah yang menjadi poin-poin penting lainnya pada proses penjadwalan ini. Resiko yang dapat terjadi apabila perusahaan tidak menerapkan saran ini adalah masalah-masalah penjadwalan yang kerap berubah dari waktu ke waktu dan hal ini membutuhkan penggunaan teori-teori yang memadai pula. Contohnya, penggunaan teknik *lot streaming* dapat memberikan perusahaan sebuah pengetahuan dalam bentuk perhitungan persentase produktivitas terhadap bagaimana meningkatkan kinerja proses penjadwalan.

DAFTAR PUSTAKA

- Kaban A., K., Othman, Z., & Rohmah D., Z. (2012). Comparison Of Dispatching Rules In Job-Shop Scheduling Problem Using Simulation: A Case Study. *International Journal of Simulation Modelling*, 11, 129-140.
- Liao, C. Jong *et al.* (2013). Meta-heuristics for Manufacturing Scheduling and Logistics Problems. *International Journal of Production Economics*, 1-3.
- Marichelvam, M. K., Prabaharan, T., & Yang, X. S. (2012). A Discrete Firefly Algorithm for The Multi-Objective Hybrid Flowshop Scheduling Problems. *IEEE Transactions On Evolutionary Computation*, 124.
- Righi, R. d. (2012). *Production Scheduling*. Sao Leopoldo: InTech.
- Ruiz, R., Antonio, J., & Vázquez-Rodríguez. (2009). *The Hybrid Flow Shop Scheduling Problem*. Nottingham: Jubilee Campus .
- Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2008). *Systems Analysis and Design in a Changing World* (5th ed.). Boston: Course Technology, CENGAGE Learning.

- Sayadi, M. K., Ramezani, R., & Ghaffari-Nasab, N. (2010). A Discrete Firefly Meta-heuristic with Local Search for Makespan Minimization in Permutation Flow Shop Scheduling Problem. *International Journal of Industrial Engineering Computations*, 1-10.
- Şerifoğlu, F. S., & Ulusoy, G. (2004). Multiprocessor task scheduling in multistage hybrid flow-shops: a genetic algorithm approach. *Journal of the Operational Research Society*, 55.
- Yang, X. S. (2010). Firefly algorithm, Lévy flights and global optimization. *Research and Development in Intelligent Systems XXVI*, 209-218.
- Yang, X. S. (2010). *Nature-Inspired Metaheuristic Algorithms: Second Edition*. United Kingdom: Luniver Press.