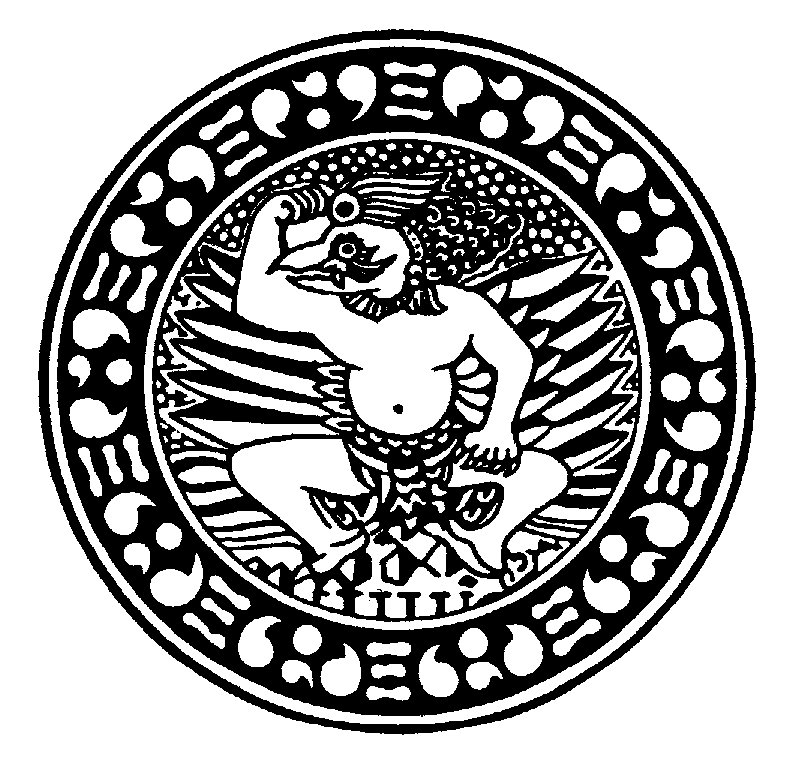
DRAFT PROPOSAL SKRIPSI

PENYELESAIAN MASALAH *FLOW SHOP* DENGAN METODE *PARTICLE SWARM OPTIMIZATION* TERMODIFIKASI DENGAN PENDEKATAN *GRAVITATIONAL INTERACTIONS*



## SHOF RIJAL AHLAN ROBBANI

**NIM : 081411631038**

**PROGRAM STUDI S1 SISTEM INFORMASI**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS AIRLANGGA**

**SURABAYA**

**2017**

**DAFTAR ISI**

[**BAB I 1**](#_Toc483898361)

[**PENDAHULUAN 1**](#_Toc483898362)

[**1.1. Latar Belakang 1**](#_Toc483898363)

[**1.2. Rumusan Masalah 4**](#_Toc483898364)

[**1.3. Tujuan 4**](#_Toc483898365)

[**1.4. Manfaat 5**](#_Toc483898366)

[**1.5. Batasan Masalah 5**](#_Toc483898367)

[**BAB II 6**](#_Toc483898368)

[**TINJAUAN PUSTAKA 6**](#_Toc483898369)

[**2.1. Penjadwalan 6**](#_Toc483898372)

[**2.1.1. Elemen Penjadwalan 7**](#_Toc483898376)

[**2.1.2. Gantt Chart 7**](#_Toc483898377)

[**2.2. Flow Shop 8**](#_Toc483898378)

[**2.3. Particle Swarm Optimization 11**](#_Toc483898379)

[**2.3.1. Algoritma Particle Swarm Optimization 14**](#_Toc483898380)

[**2.3.2. Inialisasi populasi 15**](#_Toc483898381)

[**2.3.3. Personal Best (Pti) 15**](#_Toc483898382)

[**2.3.4. Global Best () 15**](#_Toc483898383)

[**2.3.5. Update Velocity () 16**](#_Toc483898384)

[**2.4. Particle Swarm optimization based on Gravitational Field Interactions 18**](#_Toc483898385)

[**2.5. PHP 20**](#_Toc483898386)

[**2.5.1. Keunggulan PHP 21**](#_Toc483898387)

[**2.5.2. Script Dasar PHP 21**](#_Toc483898388)

[**BAB III 23**](#_Toc483898389)

[**METODE PENELITIAN 23**](#_Toc483898390)

# **BAB I**

# **PENDAHULUAN**

## Latar Belakang

Dalam bidang Industri, terutama bagian manufaktur. Proses produksi suatu produk berjalan setiap saat untuk memenuhi semua kebutuhan dan permintaan perusahaan. Supaya dapat menjalankan segala proses produksi dengan lancar maka pengaturan meliputi segla hal tentang produksi harus dipersiapkan salah satunya adalah penjadwalan. Penjadwalan adalah untuk mempersiapkan dan mengatur segala alur produksi suatu produk agar dapat berjalan sesuai yang diinginkan. Penjadwalan yang dilakukan antara lain membuat urutan produksi pekerjaan, urutan mesin yang digunakan. Penjadwalan merupakan salah satu aspek yang utama dalam proses produksi. Karena dengan adanya penjadwalan pekerjaan, dapat ditentukan urutan proses yang akan dikerjakan terlebih dahulu. Sehingga dapat membuat biaya dan waktu menjadi lebih optimal. Oleh karena itu untuk melakukan penyusunan penjadwalan diperlukan untuk mempertimbangkan beberapa aspek yang dapat mempengaruhi urutan penjadwalan.

Menyusun penjadwalan yang baik dapat dilakukan dengan cara membuat pengkodean pada mesin yang akan digunakan, produk yang akan diproduksi dan lain sebagainya. Selain itu dapat juga mendefinisikan seluruh spesifikasi dari produk dan mesin agar dapat mengetahui hal apa saja yang dapat berpengaruh pada penjadwalan. Mesin dapat dikodekan menjadi beberapa bagian dan dapat diketahui kemampuan produksi yang dilakukan. Pekerjaan juga dapat dikodekan dan diketahui berapa lama waktu yang dibutuhkan untuk memproduksi. Hal tersebut dapat mempermudah penjadwalan dalam menentukan urutan yang sesuai.

Pada lantai produksi suatu Industri, banyak masalah yang timbul pada penjadwalan. Salah satu permasalahan yang terjadi meliputi penentuan urutan pekerjaan yang akan dilakukan pada setiap mesin dengan urutan yang sama. Masalah tersebut dikenal sebagai *flow shop. Flow shop* adalah sebuah sistem pemrosesan yang mana urutan tugas dari masing-masing pekerjaan ditentukan sepenuhnya dan semua pekerjaan melewati mesin dalam urutan yang sama (Emmons & Vairaktarakis, 2013). Masalah penjadwalan *flow shop* yang paling sederhana adalah penjadwalan *flow shop* pada lini produksi tunggal (hanya ada satu mesin). Dikatakan sederhana karena mesin hampir tidak memiliki waktu luang untuk berhenti produksi hal ini membuat minimnya masalah penjadwalan dalam satu mesin. Sedangkan masalah yang sering timbul di perusahaan adalah penjadwalan *flow shop* pada lini produksi paralel (ada satu operasi atau lebih yang memiliki lebih dari satu mesin). Karena kondisi mesin yang saling menunggu mengakibatkan mesin memiliki waktu luang yang banyak untuk tidak berproduksi hal itu membuat masalah baru dalam penjadwalan karena kurangnya mengoptimalkan mesin yang ada. Masalah utama dari *flow shop* adalah penentuan urutan pekerjaan sehingga dapat meminimalkan waktu penyelesaian pekerjaan (*make-span*).

*Make-span* adalah waktu yang ditempuh proses produksi untuk memproduksi sebuah produk. Jika terdapat jadwal dengan waktu penyelesaian pekerjaan (*make-span*) yang minimum, maka dapat meminimalkan waktu produksi. Untuk mendapatkan waktu penyelesaian pekerjaan (*make-span*) yang optimal harus d icoba beberapa kombinasi antara pekerjaan dan mesin, hal ini akan memerlukan waktu dan tenaga yang besar. Sehingga problem yang ada dapat dikatakan sebagai masalah kombinatorik. Masalah kombinatorik tersebut dapat diselesaikan dengan menggunakan metode meta-heuristik.

Metode Meta-heuristik adalah suatu metode yang digunakan untuk mencari nilai optimal dengan memberikan beberapa kemungkinan untuk menciptakan suatu solusi global. Metode meta-heuristik biasanya berupa penerapan prosedur umum dalam sebuah masalah. Metode meta-heuristik memiliki karakteristik antara lain merupaakan strategi panduan proses pencarian, melakukan pencarian pada search space agar menemukan solusi optimal terdekat, teknik yang digunakan mulai dari prosedur pencarian lokal yang sederhana hingga proses belajar yang kompleks, termasuk perkiraan dan biasanya non-deterministik, menggabungkan mekanisme untuk menghindari terperangkap dalam area sear*ch space* yang terbatas, bukan masalah spesifik , menggunakan pengalaman untuk panduan pencarian (Blum & Roli, 2003). Metode meta-heuristik, antara lain Algoritma Genetika (GA), Algoritma Simulated Annealing (SA), Algoritma Tabu Search, Algoritma Particle Swarm Optimization (PSO).

Particle Swarm Optimization adalah optimasi stokastik berbasis teknik populasi yang dikembangkan oleh Dr. Eberhart dan Dr. Kenedy pada tahun 1995, terinpirasi dari perilaku social sekelompok burung atau ikan (Lazinica, 2009). Particle Swarm Optimization adalah salah satu metode metaheuristik yang bisa digunakan untuk menyelesaikan problem kombinatorik. Algoritma ini berawal dari sebuah populasi yang dibentuk secara random. Anggota dari populasi dinamakan dengan partikel. Tiap partikel digunakan sebagai prosedur pencarian dengan saling bertukar informasi antara satu dengan yang lain tentang posisi terbaik yang ada pada *search space.* Tiap partikel akan dievaluasi untuk memperbaiki posisinya pada tiap iterasi berdasarkan dari pengalaman sebelumnya maupun pengalaman dari seluruh partikel. Selanjutnya semua partikel akan bergerak menuju posisi terbaik yang telah didapatkan sebelumnya.

Untuk menyelesaikan masalah penjadwalan *Flow Shop*, Particle Swarm Optimization membentuk suatu populasi yang mempresentasikan urutan pekerjaanyang membentuk jadwal *Flow Shop* pada tiap partikel. Jadwal-jadwal tersebut akan dievaluasi dengan menggunakan kriteria *make-span* untuk mendapatkan jadwal terbaik dengan waktu minimal pengerjaan. Seiring pekermbangan, terdapat pengembangan dari Algoritma Particle Swarm Optimization yaitu algortima Particle Swarm Optimization based on Gravitational Field Interactions yang mana algoritma tersebut memiliki kesamaan dengan algortima Particle Swarm Optimization, namun terdapat parameter yang diganti berdasarkan posisi *gravitational interactions* antara partikel. Algoritma Particle Swarm Optimization based on Gravitational Field Interactions bergantung pada posisi partikel saat ini. Oleh karena itu hasil yang dihasilkan oleh algortima Particle Swarm Optimization based on Gravitational Field Interactions lebih stabil daripada Particle Swarm Optimization standar.

Berdasarkan kendala dan masalah yang diuraikan diatas. Penulis ingin membuat solusi baru untuk penyelesaian masalah *Flow Shop* dengan menggunakan metode Particle Swarm Optimization based on Gravitational Field Interactions. Pengunaan pendekatan Gravitational Field Interactions diharapkan memberikan hasil yang lebih optimal. Pada perancangan sistem yang akan di bangun, penyelesaian masalah akan dibangun dengan membuat sistem berbasis web dengan bahasa pemograman PHP. Dengan menggunakan aplikasi berbasis web diharapakan dapat mempermudah pengguna dalam melakukan peneyelesaian masalah *flow shop* yang ada dengan mudah dan fleksibel.

## Rumusan Masalah

1. Bagaimana cara yang tepat untuk penerapan algoritma PSO-GI dalam *Flow Shop Scheduling & Sequencing Problems*?
2. Bagaimana perbandingan hasil penerapan algortima PSO dan Modified PSO GI dalam *Flow Shop Scheduling & Sequencing Problems*?
3. Bagaimana cara mengimplementasikan algortiama Particle Swarm Optimization based on Gravitational Field Interactions untuk masalah Flow Shop dengan bahas pemograman PHP?

## Tujuan

1. Menerapkan penyelesaian Flow Shop Scheduling & Sequencing Problem menggunakan metode Particle Swarm Optimization based on Gravitational Field Interactions
2. Membandingkan hasil penyelesaian masalah Flow Shop antara metode Particle Swarm Optimization dan Particle Swarm Optimization based on Gravitational Field Interactions
3. Mengimplemantasikan algortima Particle Swarm Optimization based on Gravitational Field Interactions untuk penyelesaian masalah Flow Shop dengan Bahasa Pemograman PHP

## Manfaat

Memberikan alternatif penyelsaian masalah Flow Shop dengan algortima Particle Swarm Optimization dan Particle Swarm Optimization based on Gravitational Field Interactions. Sehingga algortima tersebut dapat di implementasikan dalam permasalahan produksi sebeneranya. Dan mengerti kelebihan dan kekurangan algortima yang digunakan. Serta dapat menjadi modul untuk kalanagan yang membutuhkan algortima tersebut untuk meneyelesaikan masalah penjadwalan Flow Shop.

## Batasan Masalah

Batasan masalah dari penulisan ini adalah pengambilan data yang dilakukan secara sekunder bukan data primer. Karena sistem yang akan di buat akan diterapkan pada beberapa kasus.

# **BAB II**

# **TINJAUAN PUSTAKA**



## Penjadwalan

Penjadwalan dapat didefinisikan sebagai suatu petunjuk atau indikasi apa saja yang harus dilakukan, dengan siapa, dan dengan peralatan apa yang digunakan untuk menyelesaikan suatu pekerjaan pada waktu tertentu (Scroeder, 2000).

Penjadwalan adalah proses pengambilan keputusan yang digunakan secara reguler pada beberapa industri manufaktur dan jasa. Hai ini berhubungan dengan alokasi sumber daya untuk suatu pekerjaan selama periode tertentu dan tujuannya adalah mengoptimalkan satu atau lebih tujuan (Pinedo, 2010).

Penjadwalan memiliki pengertian secara khusus sebagai durasi dari waktu kerja yang dibutuhkan untuk melakukan serangkaian aktivitas kerja yang ada dalam bidang konstruksi (Bennatan, 1995)

Penjadwalan juga merupakan proses penyusunan daftar pekerjaan yang akan dilakukan untuk mencapai atau mewujudkan suatu tujuan tertentu yang juga memuat tabel waktu pelaksanaannya (Gould, 1997).

Menurut Everett dan Robert (1999), penjadwalan produksi merupakan bagian dari shop floor control, yang mencakup tahapan loading, sequencing, dan detailed scheduling. Loading ialah kegiatan dimana setiap job ditentukan rute prosesnya. Lalu, beban (load) yang harus diselesaikan oleh setiap mesin ditentukan dengan melihat job apa saja yang akan diproses oleh mesin tersebut. Pada tahap sequencing, ditentukan urutan pengerjaan setiap job baik secara keseluruhan atau untuk setiap mesin. Tahap terakhir yaitu detailed scheduling, yaitu tahap penentuan waktu mulai dan waktu selesai dari setiap operasi.



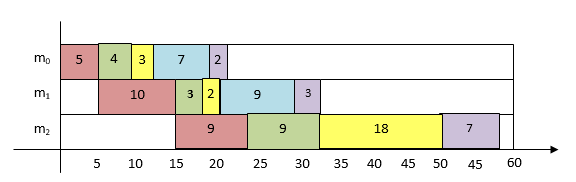
### **Elemen Penjadwalan**

Menurut Pinedo (2010), elemen – elemen penjadwalan antara lain sebagai berikut:

1. Jumlah job (n) yang akan dijadwalkan.
2. Jumlah mesin (m) yang akan dilalui dalam menyelesaikan proses operasi.
3. Processing Time (pij), yaitu waktu proses yang diperlukan untuk sebuah mesin j menyelesaikan operasi dari suatu job i.
4. Completion Time (Ci), yaitu waktu penyelesaian seluruh operasi untuk suatu job i.
5. Makespan, yaitu waktu yang digunakan untuk penyelesaian seluruh job yang akan dijadwalkan.

### **Gantt Chart**

Gantt chart merupakan sebuah alat sederhana yang digunakan untuk merepresentasikan waktu dimana digambarkan dengan balok atau garis yang terdapat pada sebuah tabel atau chart (Basu, 2008). *Gantt Chart* dikembangkan oleh Henri l Gantt pada masa PD 1 *Gantt Chart* merupakan representasi grapis dari pekerjaan-pekerjaan  yang harus diselesaikan, digambarkan dalam bentuk-bentuk susunan blok-blok batang analog dengan waktu penyelesaian pekerjaan-pekerjaan tersebut. Salah satu contoh gantt chart untuk permasalahan penjadwalan 3 mesin, 5 job ditunjukkan pada Gambar 2.1 seperti berikut



**Gambar 2.1. Gantt chart**

## Flow Shop

*Flow shop* adalah sebuah sistem pemrosesan yang mana urutan tugas dari masing-masing pekerjaan ditentukan sepenuhnya dan semua pekerjaan melewati mesin dalam urutan yang sama (Emmons & Vairaktarakis, 2013). Mesin-mesin pada model ini disusun secara seri, sehingga ketika suatu *job* telah selesai diproses pada satu mesin maka *job* tersebut akan langsung masuk antrian mesin selanjutnya, sehingga dapat dikatakan operator *job* hanya bergerak satu arah dari proses awal sampai akhir.

Masalah penjadwalan *Flow Shop* adalah masalah optimasi kombinatorial dan hal itu merupakan masalah Non polimonial lengkap (Garey, Johnson, & Sethi, 1976).

Permasalahan penjadwalan flowshop pada dasarnya adalah untuk menemukan sebuah urutan job pada setiap mesin yang sesuai dengan ketentuan yang ada. Menurut T. E. Morton (1993), ada beberapa pola penjadwalan flowshop antara lain :

1. Pure Flow Shop

Pure flow shop adalah flow shop yang semua jobnya memerlukan satu operasi pada tiap mesin. Pure Flow Shop dapat juga dikatakan sebagai simple Flow Shop karena tiap job memerlukan jumlah mesin dan urutan yang sama. Untuk memperjelas pola Pure Flow Shop dapat dilihat pada Gambar 2.2 seperti berikut :



**Gambar 2.2** **Pure Flow Shop**

1. Skip Flow Shop

Skip Flow Shop adalah Flow Shop yang semua jobnya mungkin membutuhkan kurang dari m operasi dan operasi-operasi tersebut mungkin tidak memerlukan pasangan mesin sesuai urutan nomornya dan mungkin tidak selalu berawal atau berakhir pada mesin yang sama. Bagi job yang jumlah operasinya kurang dari m, waktu pemrosesan untuk operasi yang bersangkutan diberi nilai nol. Pola ini disebut juga skip flow shop atau pola melompat, dimana mesin-mesin tertentu dapat dilompati oleh job-job tertentu.

1. Compound Flow Shop

Compound Flow Shop adalah Flow Shop yang satu mesin dalam set mesin dapat digantikan oleh sekelompok mesin. Kelompok mesin tersebut umumnya adalah mesin-mesin paralel atau jalur batch yang diikuti oleh mesin-mesin paralel. Contohnya, pabrik kertas, besi, pembotolan, makanan, dan lain-lain.

1. Reentrant Flow Shop

Reentrant flow shop adalah flow shop yang beberapa mesin dapat memproses sebuah job lebih dari sekali. Hal ini sulit dimodelkan secara analitis, tetapi relatif mudah bila digunakan pendekatan heuristik. Contohnya, suatu pipa diekstruksi secara berulang-ulang untuk mendapatkan diameter yang lebih kecil. pada tiap ekstruksi, pipa tersebut kembali pada tungku yang sama untuk ditempa.

1. *Finite Queue Flow Shop*

Finite Queue Flow Shop adalah Flow Shop yang memiliki tempat penyimpanan terbatas di depan tiap mesin. Dalam kasus khusus dapat juga terjadi bahwa tidak boleh ada antrian selain pada mesin pertama, misalnya pada industri pemrosesan logam.

1. *Permutation Flow Shop*

Permutation flowshop termasuk dalam pure flowshop dimana processing times pada semua job telah diketahui sebelumnya, dengan sebuah urutan pada n job telah ditentukan. Job-job menjalankan operasi pada semua mesin yang ada tanpa mengubah urutannya.

Adapun asumsi-asumsi yang dimiliki oleh flowshop ini adalah :

1. Setiap *job* diproses pada semua mesin dengan urutan mesin 1,2,…,*m*.
2. Setiap mesin hanya memproses sebuah job pada saat yang sama.
3. Setiap *job* hanya diproses di satu mesin pada saat yang sama.
4. Operasi tidak bersifat *pre-emptable* atau sebuah *job* harus diselesaikan dulu prosesnya secara keseluruhan di sebuah mesin sebelum diproses di mesin selanjutnya.
5. Set-up time dari operasi termasuk pada processing time dan tidak bergantung pada urutan.

Tujuan dari masalah penjadwalan Flow Shop adalah mencari jadwal permutasi yang dapat meminimalkan waktu penyelesaian maksimum. Setiap pekerjaan dikerjakan pada setiap mesin sesuai urutan (M1,M2,…Mm). Setiap mesin hanya dapat mngerjakan pekerjaan pada saat yang sama.

Banyak metode heuristik yang dikembangkan untuk menyelesaikan masalah penjadwalan Flow Shop, seperti aturan Johnson (Johnson, 1954), Palmer’s Slope Index (Palmer, 1965), CDS (Campbell, Dudek, & Smith, 1970) dan NEH (Nawaz, Enscore, & Ham, 1963).

Metode meta-heuristik untuk menyelesaikan masalah pnejadwalan Flow Shop, seperti Tabu Search Method (Tailard, 1990), Ogbu and Smith Simulated Annealing Algorithms (Ogbu & Smith, 1990), Reeves’s genetic Algorithms (Reeves, 1995) dan Lian Particle Swarm Optimization (Lian, Gu, & Jiao, 2008)

## Particle Swarm Optimization

Particle Swarm Optimization adalah optimasi stokastik berbasis teknik populasi yang dikembangkan oleh Dr. Eberhart dan Dr. Kenedy pada tahun 1995, terinpirasi dari perilaku sosial sekelompok burung atau ikan (Lazinica, 2009). Ide awal dari particle swarm yang dikemukakan oleh Kennedy dan Eberhart adalah mengarah pada pembuatan kecerdasan komputasi dengan memanfaatkan analog sederhana dari interaksi sosial, dibandingkan dengan hanya kemampuan kognitif dari individu. Selanjutnya mengembangkannya ke dalam metode optimisasi yang dinamakan dengan Particle Swarm Optimization (PSO) (Poli dkk., 2007).

Sejak diperkenalkan pertama kali, algoritma PSO berkembang cukup pesat, baik dari sisi aplikasi maupun dari sisi pengembangan metode yang digunakan pada algoritma tersebut (Haupt, R.L. & Haupt, S.E. 2004). Oleh sebab hal tersebut, mereka mengategorikan algoritma sebagai bagian dari kehidupan rekayasa/buatan Artificial Life. Algoritma ini juga terhubung dengan komputasi evolusioner, algoritma genetik dan pemrograman evolusionari (Jatmiko, 2010).

Particle Swarm Optimization (PSO) adalah salah satu dari teknik komputasi evolusioner, yang mana populasi pada PSO didasarkan pada penelusuran algoritma dan diawali dengan suatu populasi yang random yang disebut dengan particle. Berbeda dengan teknik komputasi evolusioner lainnya, setiap particle di dalam PSO juga berhubungan dengan suatu velocity. Partikel-partikel tersebut bergerak melalui penelusuran ruang dengan velocity yang dinamis yang disesuaikan menurut perilaku historisnya. Oleh karena itu, partikel-partikel mempunyai kecenderungan untuk bergerak ke area penelusuran yang lebih baik setelah melewati proses penelusuran.

Menurut Chen & Shih (2013) posisi dari tiap partikel dapat dianggap sebagai calon solusi (candidate solution) bagi suatu masalah optimisasi. Tiap-tiap partikel diberi suatu fungsi fitness merancang sesuai dengan menunjuk masalah yang yang bersesuaian. Ketika masing-masing partikel bergerak ke suatu posisi baru didalam ruang pencarian, itu akan mengingat sebagai personal best (Pbest). Sebagai tambahan terhadap ingatan informasi sendiri, masing-masing partikel akan juga menukar informasi dengan partikel yang lain dan mengingat global best (Gbest). Kemudian masing-masing partikel akan meninjau kembali arah dan percepatannya sesuai dengan Pbest dan Gbest untuk bergerak ke arah yang optimal dan menemukan solusi yang optimal.

Menurut Wati (2011), beberapa istilah umum yang biasa digunakan dalam Particle Swarm Optimization dapat didefinisikan sebagai berikut:

1. Swarm : populasi dari suatu algoritma.
2. Particle: anggota (individu) pada suatu swarm. Setiap particle merepresentasikan suatu solusi yang potensial pada permasalahan yang diselesaikan. Posisi dari suatu particle adalah ditentukan oleh representasi solusi saat itu.
3. Pbest (Personal best): posisi Pbest suatu particle yang menunjukkan posisi particle yang dipersiapkan untuk mendapatkan suatu solusi yang terbaik.
4. Gbest (Global best) : posisi terbaik particle pada swarm atau posisi terbaik diantara Pbest yang ada.
5. Velocity (v): vektor yang menggerakkan proses optimisasi yang menentukan arah di mana suatu particle diperlukan untuk berpindah (move) untuk memperbaiki posisinya semula atau kecepatan yang menggerakkan proses optimasi yang menentukan arah dimana particle diperlukan untuk berpindah dan memperbaiki posisinya semula.
6. Inertia weight (θ): inertia weight di simbolkan w, parameter ini digunakan untuk mengontrol dampak dari adanya velocity yang diberikan oleh suatu particle.
7. Learning Rates (c1 dan c2) : suatu konstanta untuk menilai kemampuan particle (c1) dan kemampuan sosial swarm (c2) yang menunjukkan bobot dari particle terhadap memorinya.

Menurut Bai (2010) keuntunggan dari Algoritma PSO adalah:

1. PSO berdasar pada kecerdasan (intelligence). Ini dapat diterapkan ke dalam kedua penggunaan dalam bidang teknik dan riset ilmiah.
2. PSO tidak punya overlap dan kalkulasi mutasi. Pencarian dapat dilakukan oleh kecepatan dari partikel. Selama pengembangan beberapa generasi, kebanyakan hanya partikel yang optimis yang dapat mengirim informasi kepartikel yang lain, dan kecepatan dari pencarian adalah sangat cepat.
3. Perhitungan didalam Algoritma PSO sangat sederhana, menggunakan kemampuan optimisasi yang lebih besar dan dapat diselesaikan dengan mudah.
4. PSO memakai kode/jumlah yang riil, dan itu diputuskan langsung dengan solusi, dan jumlah dimensi tetap sama dengan solusi yang ada.

Lebih lanjut Bai (2010) menjelaskan beberapa kerugian dari Algoritma PSO adalah:

1. Metode mudah mendapatkan optimal parsial (sebagian), yang mana menyebabkan semakin sedikit ketepatannya untuk peraturan tentang arah dan kecepatan.
2. Metode tidak bisa berkembang dari permasalahan sistem yang tidak terkoordinir, seperti solusi dalam bidang energi dan peraturan yang tidak menentu didalam bidang energy.

### **Algoritma Particle Swarm Optimization**

Langkah – langkah prosedur penerapan algoritma PSO adalah sebagai berikut :

1. Inialisasi populasi dari n-particles secara acak
2. Menghitung *fitness function*
3. Memilih *fitness value* terbaik
4. Menghitung kecepatan partikel dan posisi
5. Mengupdate kecepatan partikel dan posisi
6. Penghentian PSO

(Teugeh dkk., 2009)

### **Inialisasi populasi**

PSO dimulai dengan membangkitkan populasi secara random. Populasi berisi kandidat-kandidat solusi yang digunakan untuk menyelesaikan sebuah permasalahan yang akan terus dievaluasi pada tiap iterasinya. Pada PSO, populasi yang diambil umumnya tidak terlalu besar, antara 20 sampai 50.

### **Personal Best (Pti)**

*Personal best* (Pti) merupakan vektor yang menggambarkan posisi terbaik untuk *particle i* berdasarkan nilai *fitness* terbaik hingga iterasi *t*. Fungsi objektif yang digunakan untuk meminimalkan *makespan* adalah dimana menunjukkan permutasi job pada partikel , sedangkan personal best pada partikel I ditunjukkan seperti . Secara sederhana, fungsi fitness pada personal best dapat ditulis . Personal best dapat ditulis untuk tiap partikel I, dimana posisi terbaik (Ptij) direpresentasikan sebagai position value untuk personal best ke-I terhadap job j pada iterasi t. pada setiap iterasinya, personal best akan diperbaiki apabila , untuk i = 1, 2, …, (Tasgetiren dkk*.*, 2004).

### **Global Best ()**

*Global best* () merupakan *particle* terbaik, yang berisi kumpulan posisi terbaik, dari seluruh *personal best* pada iterasi *t*. Berdasarkan pengertian tersebut, *global best* dapat diperoleh dengan *Global best* sendiri dapat ditulis dimana merupakan position value untuk particle yang terpilih sebagai *global best* terhadap *job j* pada iterasi *t*. Pada setiap iterasinya, *global best* akan diperbaiki apabila , untuk i = 1, 2, …, (Tasgetiren dkk*.*, 2004).

### **Update Velocity ()**

Particle dapat bergerak menelusuri ruang solusi dengan velocity. Pada dasarnya *velocity* digunakan untuk menentukan arah dimana suatu *particle* diperlukan untuk berpindah dan memperbaiki posisi sebelumnya sehingga *particle* dapat menuju ke ruang solusi yang lebih baik. *Velocity* akan diperbaiki pada tiap iterasi dengan memperhatikan beberapa hal, antara lain *velocity* sebelumnya, pengaruh *personal best* dan *global best* pada iterasi sebelumnya. *Velocity* diperbaiki dengan menggunakan persamaan berikut:

Selanjutnya akan dibentuk populasi baru dengan memperbaiki position value pada tiap *particle* dengan persamaan berikut :

, dimana i = 1, 2, .., ;j = 1, 2, .., n (2.2)

Dengan

: *velocity* yang terletak pada *particle i*, *job j* untuk iterasi *t*-1

: *inertia weight* pada iterasi *t*-1

: *position value* yang terletak pada *particle i*, *job j* untuk iterasi *t*

: *position value* yang terletak pada *particle i*, *job j* untuk iterasi t-1

: *personal best* yang terletak pada *particle i*, *job j* untuk iterasi *t*-1

: *global best* pada *job j* untuk iterasi *t*-1

: *cognitive* dan *social parameter*

: random uniform [0,1]

(Tasgetiren dkk*.*, 2004)

#### **Inertia Weight**

*Inertia weight* merupakan salah satu parameter yang ada pada PSO yang berfungsi sebagai pengontrol pengaruh dari *velocity* sebelumnya untuk *velocity* yang sekarang. Pada dasarnya, *inertia weight* diperkenalkan untuk keseimbangan antara kemampuan penelusuran *global* dan *local*. *Inertia weight* akan diperbaiki dengan menggunakan persamaan:

Dengan

: *inertia weight* pada iterasi *t*

: *inertia weight* pada iterasi *t-1*

: faktor pengurangan (*decrement factor*)

Nilai awal pada parameter ini biasanya berkisar antara 0.4 sampai dengan 0.9 dan nilai yang biasa digunakan 0.975 (Uysal dan Bulkan, 2008).

#### **Cognitive dan Social Parameter**

*Cognitive parameter* merupakan parameter yang digunakan untuk mengontrol pengaruh dari *personal best* terhadap *position value* pada iterasi sebelumnya*. Social parameter* merupakan parameter yang digunakan untuk mengontrol pengaruh dari *global best* terhadap *position value* pada iterasi sebelumnya. *Cognitve* dan *social parameter* berisi konstanta-konstanta yang umumnya bernilai 1.5 – 2.0 dan 2.0 – 2.5 untuk masing-masing parameter (Hasan, 2004).

## Particle Swarm optimization based on Gravitational Field Interactions

M. Spichakova (2016) melakukan modifikasi terhadap Algoritma Particle Swarm Optimization. Ia membuktikan bahwa Modifikasi PSO+GI lebih optimal dibandingkan dengan PSO standar. Karena nilai learning coeffiecients berdasarkan pada posisi *gravitational interactions* antara partikel yang mana jauh lebih stabil daripada nilai inputan dari user.

Terdapat empat kekuatan uatama yang ada di alam semesta, yaitu : gravitasi, elektromagnetik, nuklir lemah, dan nuklir kuat. Kekuatan yang paling lemah adalah gravitasi. Hal itu terjadi karena benda bergerak tergantung pada massa masing masing. Gaya gravitasi antara dua benda i dan j berbanding lurus dengan massa mereka dan berbanding terbalik dengan jarak kuadrat di antara keduanya. Sehingga

(2.4)

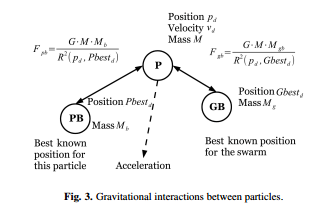
Untuk mengetahui gaya yang bekerja,kita dapat menghitung akselarasi sebagai berikut∶

(2.5)

Untuk membuat algoritma search berdasarkan gravitasi, kita dapat menggunakan ide berikut :

* + 1. Setiap objek memiliki massa dan posisi
    2. Terdapat interaksi antara objek
    3. Objek yang lebih besar dengan massa yang lebih berat membuat ruang gravitasi yang besar dan menarik yang lebih kecil.

Saat ini, algoritma PSO+GI mirip dengan algoritma PSO standar. Partikel saat ini menarik ke kedua posisi Pbest dan Gbest. Dalam algoritma PSO kekuatan atraksi dana arah gerakan ini dikontrol oleh learning coefficients. Pada kasus ini, kita melakukan penghitungan ulang terhdap kekutan antara ketiga titik tersebut dan menghitung akselerasinya. Seperti yang ada pada Gambar 2.3 sebagai berikut :



**Gambar 2.3 Interasksi Gravitasi antar Partikel**

Perubahan pada algoritma ini adalah merubah nilai c1 dan c2 menjadi agb dan apb. Sehingga algoritma berubah menjadi :

Dengan

= Massa Inersia

= Akselerasi terhadap Pbest

= Akselerasi terhadap Gbest

## PHP

Menurut Agus Saputra (2011) PHP atau yang memiliki kepanjangan PHP *Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi.

PHP/F1 merupakan nama awal dari PHP (Personal Home Page / Form Interface). Dibuat pertama kali oleh Rasmus Lerdoff. PHP awalnya merupakan program CGI yang dikhususkan untuk menerima input melalui form yang ditampilkan dalam browser web. Dengan menggunakan PHP maka maintenance suatu situs web menjadi lebih mudah. Proses update dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan script PHP.

PHP berjalan pada sisi server sehingga PHP disebut juga sebagai Bahasa *Server Side Scripting*. Artinya bahwa dalam setiap/untuk menjalankan PHP, wajib adanya web server.

PHP ini bersifat open source sehingga dapat dipakai secara cuma-cuma dan mampu lintas platform, yaitu dapat berjalan pada sistem operasi Windows maupun Linux.

### **Keunggulan PHP**

Ada beberapa alasan yang menjadi dasar pertimbangan mengapa menggunakan PHP sebagai berikut :

* + 1. Mudah dipelajari.
    2. Mampu Lintas *Platform*.
    3. Bersifat *Open Source*.
    4. PHP memiliki tingkat akses yang cepat.
    5. Didukung oleh beberapa macam web server
    6. Mendukung database

### **Script Dasar PHP**

Setiap program PHP disebut dengan script. Script berupa file teks, yang dapat dibuat dengan menggunakan program editor file teks biasa seperti notepad, edit, (dalam lingkungan Unix/linux), atau lainnya. Editor teks yang digunakan sebaiknya editor teks yang memungkinkan membuat program PHP lebih mudah.

Script PHP diawali dengan tag (<?) dan diakhiri dengan tag (?>). Setiap baris perintah / statement harus diakhiri dengan menggunakan tanda titik koma (;). Umumnya setiap statement dituliskan dalam satu baris. Script PHPmerupakan script yang digunakan untuk menghasilkan halaman-halaman web. Cara penulisannnya dibedakan menjadi embedded dan non embedded script.

* + - * 1. Embedded Script

Script yang dimaksud dari embedded script adalah script PHP yang disisipkan diantara tag-tag HTML. Script PHP digunakan apabila isi dari suatu dokumen HTML diinginkan dari hasil eksekusi suatu script PHP, selama informasi masih tidak membutuhkan program maka pemrogram umumnya tidak akan menggunakan program. Contoh embedded Script sebagai berikut :

<html>

<head>

<title>

Contoh

</title>

</head>

<body>

<?php

echo “hai, saya dari script php!”;

?>

</body>

</html>

* + - * 1. Non Embedded Script

Script PHP pada cara ini digunakan sebagai murni pembuatan program PHP, tag HTML yang dihasilkan untuk membuat dokumen merupakan bagian dari script PHP. Contoh non embedded script sebagai berikut :

<?php

echo “<html>”;

echo “<head>”;

echo “<title>”;

echo “contoh 02-Pure On the Fly”;

echo “</title>;

echo ”</head>”;

echo “<body>”;

echo “<p>Teks dokumen yang dihasilkan dengan

menggunakan script PHP </p>”;

echo “</body>”;

echo “</html>”;

?>

# **BAB III**

# **METODE PENELITIAN**

Langkah-langkah penyelesaian permasalahan penjadwalan flowshop dengan algoritma Particle Swarm Optimization (PSO) based on Gravitational Field Interactions adalah sebagai berikut :

1. Mencari data, problem, atau studi literatur mengenai Flow Shop Sequencing, Particle Swarm Optimization, Particle Swarm Optimization based on Gravititational Field Interactions pada Buku, Jurnal, Website, dsb.
2. Menentukan Encoding dan Decoding dari problem Flow Shop

Encoding dilakukan dengan cara menentukan pengkodean untuk tiap pekerjaan yang dilakukan. Tiap pekerjaan dilakukan pengkodean untuk menginialisasi pekerjaan. Contohnya, jika terdapat 5-pekerjaan maka dilakukan pengkodean sebagai berikut :

* Pekerjaan 1 = 0
* Pekerjaan 2 = 1
* Pekerjaan 3 = 2
* Pekerjaan 4 = 3
* Pekerjaan 5 = 4

Output yang dihasilkan adalah dapat menginialisasi urutan pekerjaan setelah dilakukan pengolahan data dengan metode Particle Swarm Optimization (PSO) based on Gravitational Field Interactions

1. Menyusun Algortima Particle Swarm Optimization (PSO) based on Gravitational Field Interactions sebagai berikut :
   1. Inialisasi populasi dari partikel partikel dengan posisi dan *velocity*secara random.

Mendeklarasikan setiap populasi atau data yang akan di olah dari beberapa solusi yang ada secara acak. Dimana Partikel dilambangkan Xi yang mana merupakan i partikel pada kawanan iterasi t dan diwakili oleh n jumlah dimensi yang diberikan persamaan sebagai berikut :

[Xi]t = [(Xi1)t, (Xi2)t,…, (Xin)t] (3.1)

Yang mana (Xij)t, adalah nilai posisi dari i partikel yang berhubungan dengan j dimensi *(j= 1,2,..., n)*.

Populasi adalah himpunan partikel p pada kawanan iterasi t yang dilambangkan dalam persmaan sebagai berikut :

popt = [(X1)t, (X2)t, …, (Xp)t] (3.2)

* 1. Menghitung *fitness function*

Menghitung fitness function, jika fitness value lebih baik daripada best fitness value dalam sejarah (Pijt-1). Maka tentukan nilai saat ini sebagai nilai baru dari Pbest. Dimana Pbest dilambangkan sebagai berikut :

[Pi]t = [(Pi1)t, (Pi2)t,…, (Pin)t] (3.3)

Yang mana (Pij)t, adalah nilai posisi dari i personal best yang berhubungan dengan j dimensi *(j= 1,2,..., n)*.

* 1. Memilih *fitness value* terbaik

Memilih partikel dengan fitness value terbaik daripada partikel lainnya sebagai *Gbest*. (Gj)t-1. Yang mana Gbest dilambangkan sebagai berikut :

[G]t = [(G1)t,(G2)t…….(Gn)t] (3.4)

Yang mana (Gj)t adalah nilai posisi dari global best yang berhubngan dengan j dimensi (j = 1,2,..., n).

* 1. Menghitung kecepatan partikel dan posisi

Setiap partikel dihitung kecepatan dan posisi dengan menggunakan persamaan sebagai berikut :

[Vij]t= [(Vij)t-1 + apbr1{(Pij)t-1- (Xij)t-1} + agbr2 {(Gj)t-1- (Xij)t-1}] (3.5)

[Xij]t= (Xij)t-1 + (Vij)t  (3.6)

Yang mana,

(Vij)t-1 = Kecepatan partikel i pada iterasi t-1

(Vij)t = Kecepatan paartikel i pada iterasi t

(Xij)t-1= Posisi partikel i pada iterasi t-1

(Xij)t = Posisi partikel i pada iterasi t

r1 = Angka acak antara 0 dan 1

r2 = Angka acak antara 0 dan 1

(Gj)t-1= Posisi global best dari populasi

(Pij)t-1= Posisi local best dari partikel

apb = Akselerasi faktor terhadap Pbest

(3.7)

agb = Akselerasi faktor terhadap Gbest

(3.8)

G = Gaya Gravitasi

M = Massa

R2 = Koefisien Determinasi

P = Posisi

* 1. Mengupdate kecepatan partikel dan posisi

Setiap kecepatan dan partikel diperbarui sesaui dengan dimensi.

* 1. Penghentian PSO

Algoritma PSO berhenti ketika iterasi telah mencapai 750. Jika tidak maka kembali ke langkah 2.

1. Membuat implementasi algortima Aprticle Swarm Optimization based on Gravitational Field Intercations untuk penyelesaian maslah flowshop pada komputer dengan menggunakan Bahasa pemograman PHP.
2. Menguji coba program dengan berbagai macam ukuran data (kecil, sedang, besar) tetang maslah flowshop.
3. Membandingkan data yang diperoleh dari hasil metode Particle Swarm Optimization (PSO) dan Modified Particle Swarm Optimization (PSO) based on Gravitational Field Interactions.