

Penjadwalan *Single Machine* Dengan Metode Algoritma *Branch And Bound* Untuk Meminimasi Total *Lateness* Dan Jumlah *Tardy Job*

Muhamad Syafei¹, Evi Febianti², Lely Herlina³
^{1,2,3}Jurusan Teknik Industri Universitas Sultan Ageng Tirtayasa
syafeimuhamad23@gmail.com¹, evi@untirta.ac.id², lely@untirta.ac.id³

ABSTRAK

PT. XYZ adalah salah satu perusahaan manufaktur yang memproduksi pipa baja las. PT. XYZ mempunyai 2 jenis mesin yaitu mesin ERW (*Electric Resistance Welding*) digunakan untuk pembuatan pipa baja dengan las longitudinal dan mesin SPM (*Spiral Pipe Machine*) digunakan untuk pembuatan pipa baja dengan las spiral. Jumlah mesin ERW yang hanya 1 buah membuat pembuatan pipa longitudinal sangat bergantung pada efektifitas penggunaan mesin ERW tersebut. Hal ini yang menyebabkan sering terjadinya keterlambatan dalam proses penyelesaian job. Tujuan dari penelitian ini yaitu meminimasi total *lateness* dan jumlah *tardy job* pada produk pipa longitudinal yang terjadi di PT. XYZ dengan metode algoritma *branch and bound*. Sebelum melakukan perhitungan algoritma *branch and bound* dilakukan penentuan jadwal inisial dengan aturan *priority rule* EDD (*earliest due date*), SPT (*shortest processing time*) dan LDD (*last due date*). Hasil penelitian ini didapatkan nilai total *lateness* pada jadwal existing sebesar 1616 jam dan 6 *tardy job* dengan urutan *sequencing job* (1-2-3-4-5-6-7-8-9-10-11-12-13-14-15), kemudian dilanjutkan dengan pembuatan jadwal inisial EDD, SPT dan LDD. Jadwal inisial EDD menghasilkan total *lateness* 720 jam dan 6 *tardy job* yaitu job ke-2, 5, 8, 11, 12 dan 14 dengan urutan job (10-9-7-13-1-3-4-6-12-8-2-5-11-14-15). Jadwal inisial SPT menghasilkan total *lateness* sebesar 880 jam dan 4 *tardy job* yaitu job ke-1, 2, 3 dan 10 dengan urutan job (14-5-8-9-11-12-13-10-4-7-15-6-3-1-2). Sementara jadwal inisial LDD menghasilkan total *lateness* sebesar 4296 jam dengan 8 *tardy job* yaitu job ke-1, 3, 4, 6, 7, 9, 10 dan 13 dengan urutan job (15-14-11-2-5-8-12-3-4-6-1-13-7-9-10). Pada perhitungan metode algoritma *branch and bound* baik itu menggunakan jadwal inisial EDD, SPT maupun LDD menghasilkan 24 alternatif *sequencing* yang sama dengan nilai total *lateness* 344 jam dan 2 *tardy job* yaitu job ke-1 dan ke-2 akan tetapi alternatif *sequencing* yang dipilih ialah alternatif *sequencing* yang dihasilkan oleh metode algoritma *branch and bound* dengan jadwal inisial EDD dan SPT. Hal ini dikarenakan jadwal inisial EDD terbukti menghasilkan jadwal alternatif dengan total *lateness* yang minimum sedangkan jadwal inisial SPT menghasilkan jadwal alternatif dengan jumlah *tardy job* yang minimum. Oleh karena itu alternatif jadwal *sequencing* dengan urutan job (9-8-5-10-7-4-11-13-12-14-15-6-3-1-2) dipilih sebagai alternatif jadwal *sequencing* terbaik.

Kata kunci: Algoritma *Branch and Bound*, *Lateness*, *Tardy Job*, EDD (*Earliest Due Date*), SPT (*Short Processing Time*), LDD (*Last Due Date*).

ABSTRACT

PT. XYZ is a manufacturing company that produces steel pipe welding. PT. XYZ has two types of machines are machines ERW (*Electric Resistance Welding*) is used for the manufacture of steel pipes with longitudinal welding and machine SPM (*Spiral Pipe Machine*) is used for the manufacture of steel pipes with spiral weldin . The number of machines that only 1 piece ERW pipe manufacture longitudinal create highly dependent on the effective use of the ERW machine. This led to frequent delays in the completion of the job. The purpose of this study is to minimize the total *lateness* and number of *tardy jobs* in longitudinal pipe products that occur in the PT. XYZ with a *branch and bound* algorithm method. Before performing the calculation of *branch and bound* algorithm is the determination of the initial schedule with *priority rules* EDD (*earliest due date*), SPT (*Shortest processing time*) and LDD (*last due date*). The results of this study, the total *lateness* on the existing schedule is 1616 hours and 6 of *tardy jobs* with job *sequencing* sequence (1-2-3-4-5-6-7-8-9-10-11-12-13-14-15), then proceed with the making the initial schedule EDD, SPT and LDD. The results of EDD initial schedule is 720 hours of total *lateness* and 6 of *tardy jobs* are jobs number 2, 5, 8, 11, 12 and 14 with a job *sequencing* (10-9-7-13-1-3-4-6-12-8-2-5-11-14-15). The results of SPT initial schedule is 880 hours of total *lateness* and 4 of *tardy jobs* are jobs number 1, 2, 3 and 10 with a job *sequencing* (14-5-8-9-11-12-13-10-4-7-15-6-3-1-2). While the results of LDD initial schedule is 4296 hours of total *lateness* with 8 of *tardy jobs* are jobs number 1, 3, 4, 6, 7, 9, 10 and 13 with a job *sequencing* (15-14-11-2-5-8-12-3-4-6-1-13-7-9-10). In the calculation method *branch and bound* algorithm that uses the initials schedule EDD, SPT and LDD produce 24 alternative

sequencing with the same value of 344 hours of total lateness and 2 of tardy jobs are jobs 1st and 2nd but the selected sequencing alternative is the result of alternative sequencing by the method of branch and bound algorithm with the initial schedule EDD and SPT. Because the initial schedule EDD shown to produce an alternative schedule with minimum total lateness and the initial schedule SPT while generating alternative schedule with the minimum number of tardy jobs. Therefore, an alternative schedule job sequencing is (9-8-5-10-7-4-11-13-12-14-15-6-3-1-2) was chosen as the best alternative sequencing schedule.

Keywords: Branch and Bound Algorithm, Lateness, Tardy Job, EDD (Earliest Due Date,) SPT (Short Processing Time), LDD (Last Due Date).

PENDAHULUAN

Perencanaan dan pengendalian produksi berfungsi sebagai perencanaan aktivitas untuk melaporkan hasil operasi dan meninjau kembali rencana yang diperlukan agar keinginan yang dijadikan tujuan tercapai. Salah satu elemen dalam perencanaan dan pengendalian produksi adalah penjadwalan (Arifin dan Rudyanto, 2010).

Penjadwalan adalah kegiatan pengalokasian sumber daya untuk mengerjakan suatu *job* pada suatu waktu (Baker, 1974). Menurut Arifin dan Rudyanto (2010) penjadwalan adalah salah satu hal yang penting dalam perusahaan manufaktur. Penjadwalan menghasilkan berbagai kriteria yang dapat digunakan oleh perusahaan. Dimana kriteria tersebut adalah ketepatan dalam penyelesaian *job* terhadap *due date* dan meminimasi lamanya pengerjaan *job* di lantai produksi. dengan banyaknya metode yang digunakan dalam melakukan penjadwalan membuat perusahaan dapat memilih metode penjadwalan sesuai dengan kriteria yang perusahaan inginkan. Oleh karena itu, masalah penjadwalan menjadi perhatian yang serius di perusahaan.

PT. XYZ adalah salah satu perusahaan manufaktur yang memproduksi pipa baja las. Pipa baja las ini terbagi menjadi pipa baja las spiral dan longitudinal. Pipa yang dihasilkan berupa pipa minyak, pipa gas, pipa air, dan pipa pancang. Bahan baku yang digunakan dalam pembuatan pipa baja adalah *hot rolling coil* (HRC). *Hot rolling coil* adalah material dasar berbentuk pelat yang digulung sehingga membentuk *coil*. Bahan baku ini didatangkan langsung dari PT. KS yang membuat *hot strip mill* (HSM), kemudian menjadi *hot rolling coil* (HRC). Dalam proses pembuatan pipa baja, PT. XYZ mempunyai dua jenis mesin produksi pembuat pipa, yaitu mesin ERW (*Electric Resistance Welding*) digunakan untuk pembuatan pipa baja dengan las longitudinal dan mesin SPM (*Spiral Pipe Machine*).

Tipe aliran produksi PT. XYZ adalah aliran *flow shop* dengan tipe produksi adalah *make to order*, karena PT. XYZ menerima pesanan berdasarkan permintaan konsumen dengan standarisasi produk seperti ASTM, AWWA, API spec 5L dan lain sebagainya. PT. XYZ

masih menggunakan metode konvensional dalam menjadwalkan produknya dengan menganut sistem FCFS (*First Come First Serve*) dimana *job* yg dikerjakan sesuai dengan urutan kedatangan *order job* tersebut tanpa memperhatikan waktu *due date* dari *job* tersebut sehingga menimbulkan total *lateness* yang cukup besar dan jumlah *tardy job* yang cukup banyak. Masalah ini sering terjadi pada mesin ERW yang memiliki jadwal produksi lebih padat dibandingkan dengan mesin SPM, karena produk pipa *longitudinal* yang dihasilkan mesin ERW lebih banyak diminati oleh konsumen dibandingkan dengan pipa *spiral* yang dihasilkan mesin SPM. Hal ini juga disebabkan karena saat ini PT. XYZ hanya memiliki 1 mesin ERW sehingga pada pembuatan pipa *longitudinal* sangat bergantung pada efektifitas penggunaan mesin ERW tersebut. Sehingga peneliti memutuskan untuk memilih mesin ERW sebagai objek penelitiannya.

Penelitian Kurniati (2012) menjelaskan pengembangan model penjadwalan menggunakan algoritma *branch and bound* dapat meminimasi *mean flowtime*. Akan tetapi, pada kenyataan algoritma *branch and bound* menghasilkan solusi yang optimal dengan mencoba semua solusi yang ada. Maka pada penelitian ini metode algoritma *branch and bound* menggunakan *priority rule* EDD, SPT dan LDD akan digunakan dalam penjadwalan pola aliran *flowshop* untuk meminimasi total *lateness* dan jumlah *tardy job*.

METODE PENELITIAN

Tahapan penelitian yang akan dilakukan untuk membuat variasi jadwal yang baru dengan menggunakan algoritma *Branch and Bound* untuk meminimasi total *lateness* dan jumlah *tardy job*, data yang diperlukan dalam penelitian ini adalah data primer dan data sekunder. Data primer seperti wawancara yang dilakukan penulis yaitu permasalahan yang dialami oleh PT. XYZ. Kemudian untuk data sekunder yang digunakan adalah data waktu proses untuk setiap *job* nya pada periode juli 2014 sebanyak 15 *job*.

Tahap pertama yang dilakukan adalah perhitungan penjadwalan eksisting pada perusahaan. Dimulai dari menentukan metode yang digunakan oleh perusahaan. Kemudian

dilakukan perhitungan untuk mencari nilai total *lateness* dan jumlah *tardy job*.

Tahap kedua adalah melakukan perhitungan jadwal inisial sebagai penentu kriteria yang dicari seperti meminimasi total *lateness* dan jumlah *tardy job*, yang bertujuan untuk mengetahui variasi jadwal yang memiliki total *lateness* dan jumlah *tardy job* minimum dan sebagai jadwal inisial untuk perhitungan algoritma *Branch and Bound*.

Tahap ketiga adalah perhitungan penjadwalan dengan algoritma *Branch and Bound* dengan jadwal inisial EDD, SPT dan LDD Kemudian dilakukan perhitungan dengan algoritma *Branch and Bound* yang pertama dengan melakukan percabangan di tiap simpul nya kemudian dari percabangan tersebut, dipilih node dengan total waktu proses yang terkecil yang dijadikan sebagai *lower bound* untuk melakukan percabangan sampai semua *job* sudah terjadwalkan.

Tahap keempat adalah perbandingan total *lateness* dan jumlah *tardy job* pada kondisi eksisting dengan penjadwalan menggunakan algoritma *Branch & Bound* dengan *priority rule* EDD, SPT dan LDD.

HASIL DAN PEMBAHASAN

Pengambilan data pada penelitian ini bersifat sekunder dan primer, data sekunder yaitu belum dapat memenuhi semua *order* konsumen secara tepat waktu. Kemudian data primer yang digunakan adalah data data produksi, data *due date job* dan data waktu proses pada periode bulan juli 2014.

Tabel 1 Data Job Mesin ERW pada bulan Juli 2014

No	Job	Spesifikasi			Jumlah (Pcs)	Process (Jam)	Due date (Jam)
		D(mm)	T(mm)	P(m)			
1	PT. Chevron Pasific Indonesia	219.1	6.35	12	1746	176	672
2	PT. Duta Hita Jaya	219.1	6	6	2429	248	960
3	PT. Catur Prima Perkasa	219.1	4.8	6	1173	120	720
4	PT. Chevron Pasific Indonesia	219.1	6.35	12	262	24	720
5	PT. Chevron Pasific Indonesia	219.1	8.18	12	43	8	960
6	PT. Chevron Pasific Indonesia	219.1	6.35	12	854	88	720
7	PT. Puteramusa Antania	219.1	4.8	6	241	24	576
8	PT. Chevron Pasific Indonesia	219.1	9.53	12	80	8	840
9	PT. Barata Indonesia	219.1	8.18	12	19	8	360
10	PT. Bumi Kaya Steel Industries	219.1	9.53	12	85	16	240
11	PT. Chevron Pasific Indonesia	323.9	9.53	12	45	8	1008
12	PT. Chevron Pasific Indonesia	323.9	12.7	12	164	16	744
13	PT. Puteramusa Antania	323.9	6	6	100	16	600
14	PT. Steel Pipe Industry of Indonesia	168.3	7.11	12	16	8	1080
15	PT. Adiguna Shipbuilding	168.3	6.35	12	521	56	1440

Pada perhitungan jadwal *existing* akan menggunakan susunan jadwal yang dimiliki oleh perusahaan yang menganut aturan FCFS(*first come first serve*) dimana *job* yang datang lebih awal yang akan dikerjakan. Setelah itu, melakukan perhitungan terhadap nilai total *lateness*. Untuk susunan *job* dapat dilihat pada tabel 1.

Dari perhitungan yang telah dilakukan maka diperoleh jumlah *job* yang terlambat terdapat 6 *job* dengan nilai total *lateness* sebesar 1616 jam dengan urutan *job* 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15.

Tabel 4.2 Hasil Perhitungan *lateness* Jadwal *Existing*

Job	Waktu Proses (Jam)	Due date (Jam)	Saat Mulai (Jam)	Saat Selesai (Jam)	Lateness (Jam)
Setup	72		0	72	
1	176	672	72	248	0
2	248	960	248	496	0
3	120	720	496	616	0
4	24	720	616	640	0
5	8	960	640	648	0
6	88	720	648	736	16
7	24	576	736	760	184
8	8	840	760	768	0
9	8	360	768	776	416
10	16	240	776	792	552
Setup	72		792	864	
11	8	1008	864	872	0
12	16	744	872	888	144
13	16	600	888	904	304
Setup	72		904	976	
14	8	1080	976	984	0
15	56	1440	984	1040	0
Total Lateness					1616

Contoh perhitungan :

$$\text{Saat Selesai} = \text{Saat Mulai} + \text{Waktu Proses}$$

$$\text{Saat Selesai job 10} = 776 + 16 = 792 \text{ jam}$$

$$\text{Lateness} = \text{Saat Selesai} - \text{Due date}$$

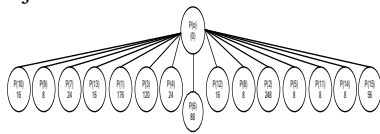
$$\text{Lateness job 10} = 792 - 240 = 552 \text{ jam}$$

1. Algoritma *Branch and Bound* dengan Jadwal Inisial EDD

Langkah-langkah penjadwalan menggunakan algoritma *branch and bound priority rule* EDD, antara lain :

- Mencari Jadwal Inisial
Sebelumnya jadwal inisial telah dihitung dan didapatkan total *lateness* sebesar 720 jam dan total *tardy job* sebanyak 6 *job* yaitu *job* ke-2, 5, 8, 11, 12 dan 14 dengan urutan *job* (10-9-7-13-1-3-4-6-12-8-2-5-11-14-15).
- Inialisasi Parameter
Tempatkan P(0) pada list yang aktif. Nilai yang terkait pada node ini adalah $V_0 = 0$ dan $p(\Phi) = \sum_{j=1}^n P_j$. Dikarenakan nilai pada algoritma *branch and bound* menggunakan integer programming yaitu 1 dan 0. Yang artinya untuk mesin dalam posisi diam diberi tanda 0 dan ketika mesin mengerjakan suatu *job* diberi tanda 1.
 $P(0) =$ Waktu proses pada *job* pada mesin dalam keadaan diam, belum berproses.
 $V_0 =$ Value pada P(0) dalam keadaan mesin diam.
- Membuat Percabangan
Membuat percabangan $X_1=1$ untuk masing-masing *job* i. Ketika mesin mengerjakan suatu *job* diberi tanda 1, P(s), dari list yang aktif dengan node baru untuk masing-masing *job* yang belum terjadwalkan. Kemudian notasikan K sebagai nomor *job* dalam urutan parsial s. Jika $K = n$, maka stop, yang berarti urutan penjadwalan sudah optimal. Jika belum optimal,

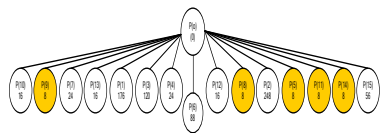
dilanjutkan dengan langkah 4 dan langkah 5.
 $K = \text{level}$
 $n = \text{banyaknya job yang akan dijadwalkan}$.



Gambar 1 Struktur Tree Algoritma Branch and Bound Priority Rule EDD Level 1

Contoh perhitungan :
 $P(s') = P(10) = \text{Job ke } 10 = 16$
 $V_{js} = V_s + P(s')$; $V_{10s} = 0 + 16 = 16$
 Kemudian, dilanjutkan ke langkah selanjutnya, dikarenakan level K tidak sama dengan n, maka belum dikatakan optimal.

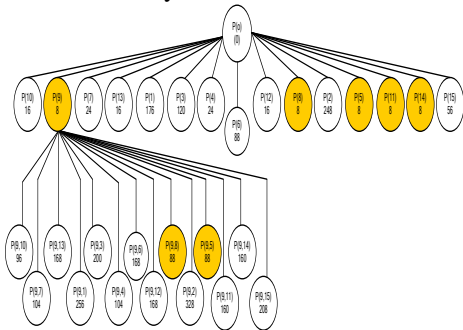
- d. Menentukan Lower Bound
 Dilanjutkan dengan memilih lower bound (batas bawah). Node yang akan dipilih, yang mempunyai nilai V_s nya lebih kecil diantara node-node yang lain.



Gambar 2 Penentuan Lower Bound Algoritma Branch and Bound Priority Rule EDD Level 1

Dan didapatkan node yang dipilih sebagai lower bound (batas bawah) yaitu node dengan P(9), P(8), P(5), P(11) dan P(14) dengan masing-masing job ke 9, 8, 5, 11 dan 14. Yang kemudian dilanjutkan dengan pembuatan node baru yang dicabangkan pada node yang telah dipilih.

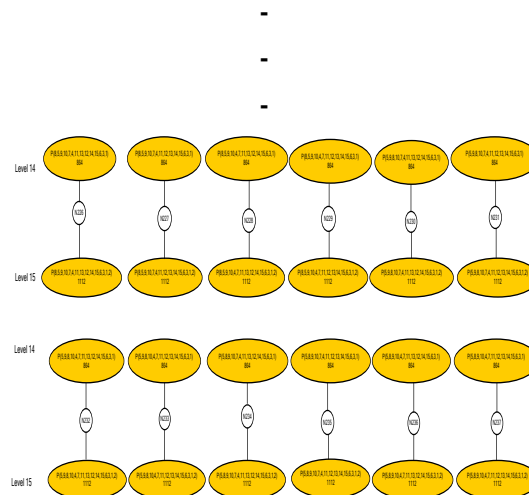
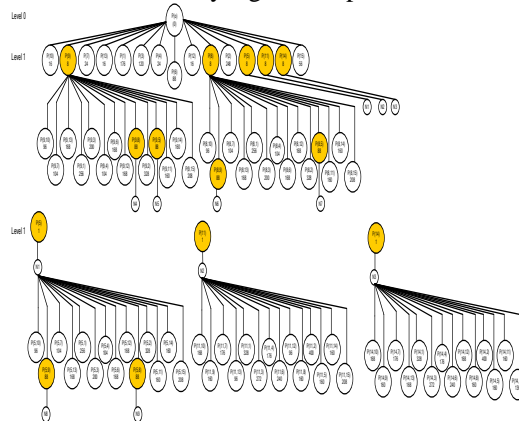
- e. Membuat Node Selanjutnya
 Pada node baru ini, P(s) yang akan dihitung, akan menggabungkan job yang belum terjadwalkan di akhir urutan parsial job dan atribut lainnya yang digunakan untuk membuat node baru dan memasukkan mereka sesuai urutan listnya.



Gambar 3 Penentuan Titik Aktif Algoritma Branch and Bound Priority Rule EDD

Contoh perhitungan
 $P(s') = P(9,8) = \text{job ke } 9 = 8, \text{ job ke } 8 = 8$
 $V_{js} = V_s + P(s')$; $V_{9,8s} = 0 + 72 + 8 + 8 = 88$
 Pada awal pekerjaan ditambahkan nilai 72 karena proses tersebut membutuhkan

waktu setup mesin sebesar 72 jam. Dari hasil perhitungan tersebut didapatkan node aktif selanjutnya yang kemudian akan dihasilkan variasi jadwal dengan total total lateness dan tardy job yang minimum yang akan dipilih.



Gambar 4 Struktur Tree Algoritma Branch and Bound dengan Jadwal Inisial EDD

berikut adalah variasi jadwal yang dihasilkan dari perhitungan algoritma branch and bound dengan priority rule EDD:

Tabel 3 Variasi Jadwal Keseluruhan Dengan Algoritma Branch and Bound dengan priority rule EDD

No	Variasi Jadwal	Urutan Job	Total Lateness (Jam)
1	S1	Job 9-8-5-10-7-4-11-12-13-14-15-6-3-1-2	344
2	S2	Job 9-8-5-10-7-4-11-12-13-14-15-6-3-1-2	344
3	S3	Job 9-8-5-10-4-7-11-12-13-14-15-6-3-1-2	344
4	S4	Job 9-8-5-10-4-7-11-12-13-14-15-6-3-1-2	344
5	S5	Job 9-5-8-10-7-4-11-12-13-14-15-6-3-1-2	344
6	S6	Job 9-5-8-10-7-4-11-12-13-14-15-6-3-1-2	344
7	S7	Job 9-5-8-10-4-7-11-12-13-14-15-6-3-1-2	344
8	S8	Job 9-5-8-10-4-7-11-12-13-14-15-6-3-1-2	344
9	S9	Job 8-9-5-10-7-4-11-12-13-14-15-6-3-1-2	344
10	S10	Job 8-9-5-10-7-4-11-12-13-14-15-6-3-1-2	344
11	S11	Job 8-9-5-10-4-7-11-12-13-14-15-6-3-1-2	344
12	S12	Job 8-9-5-10-4-7-11-12-13-14-15-6-3-1-2	344
13	S13	Job 8-5-9-10-7-4-11-12-13-14-15-6-3-1-2	344
14	S14	Job 8-5-9-10-7-4-11-12-13-14-15-6-3-1-2	344
15	S15	Job 8-5-9-10-4-7-11-12-13-14-15-6-3-1-2	344
16	S16	Job 8-5-9-10-4-7-11-12-13-14-15-6-3-1-2	344
17	S17	Job 5-9-8-10-7-4-11-12-13-14-15-6-3-1-2	344
18	S18	Job 5-9-8-10-7-4-11-12-13-14-15-6-3-1-2	344
19	S19	Job 5-9-8-10-4-7-11-12-13-14-15-6-3-1-2	344
20	S20	Job 5-9-8-10-4-7-11-12-13-14-15-6-3-1-2	344
21	S21	Job 5-8-9-10-7-4-11-12-13-14-15-6-3-1-2	344
22	S22	Job 5-8-9-10-7-4-11-12-13-14-15-6-3-1-2	344
23	S23	Job 5-8-9-10-4-7-11-12-13-14-15-6-3-1-2	344
24	S24	Job 5-8-9-10-4-7-11-12-13-14-15-6-3-1-2	344

Dan didapatkan nilai total lateness terkecil adalah untuk semua sequence baik sequence ke-1 sampai sequence ke-24 yaitu dengan nilai 344 jam dan jumlah tardy job adalah 2 dari setiap sequence tersebut yakni job ke-1 dan job

ke-2. Urutan penjadwalan *job* yang dihasilkan ialah *job* (9-8-5-10-7-4-11-13-12-14-15-6-3-1-2) karena urutan ini paling mendekati dengan aturan EDD.

2. Algoritma *Branch and Bound* dengan Jadwal Inisial SPT

Langkah-langkah penjadwalan menggunakan algoritma *branch and bound priority rule* SPT, antara lain :

- a. Mencari Jadwal Inisial
Sebelumnya jadwal inisial telah dihitung dan didapatkan total *lateness* sebesar 880 jam dan total *tardy job* sebanyak 4 *job* yaitu *job* ke-1, 2, 3 dan 10 dengan urutan *job* (14-5-8-9-11-12-13-10-4-7-15-6-3-1-2).
- b. Inialisasi Parameter
Tempatkan P(0) pada list yang aktif. Nilai yang terkait pada node ini adalah $V_0 = 0$ dan $p(\Phi) = \sum_{j=1}^n P_j$. Dikarenakan nilai pada algoritma *branch and bound* menggunakan integer programming yaitu 1 dan 0. Yang artinya untuk mesin dalam posisi diam diberi tanda 0 dan ketika mesin mengerjakan suatu *job* diberi tanda 1. $P(0) =$ Waktu proses pada *job* pada mesin dalam keadaan diam, belum berproses.
 $V_0 =$ Value pada P(0) dalam keadaan mesin diam.
- c. Membuat Percabangan
Membuat percabangan $X_1=1$ untuk masing-masing *job* i. Ketika mesin mengerjakan suatu *job* diberi tanda 1, P(s), dari list yang aktif dengan node baru untuk masing-masing *job* yang belum terjadwalkan. Kemudian notasikan K sebagai nomor *job* dalam urutan parsial s. Jika $K=n$, maka stop, yang berarti urutan penjadwalan sudah optimal. Jika belum optimal, dilanjutkan dengan langkah 3 dan langkah 4.
K= level
n= banyaknya *job* yang akan dijadwalkan.



Gambar 5 Struktur Tree Algoritma Branch and Bound Priority Rule SPT Level 1

Contoh perhitungan :

$$P(s') = P(14) = \text{Job ke } 14 = 8$$

$$V_{js} = V_s + P(s'); V_{14s} = 0 + 8 = 8$$

Kemudian, dilanjutkan ke langkah selanjutnya, dikarenakan level K tidak sama dengan n, maka belum dikatakan optimal.

- d. Menentukan Lower Bound
Dilanjutkan dengan memilih lower bound (batas bawah). Node yang akan

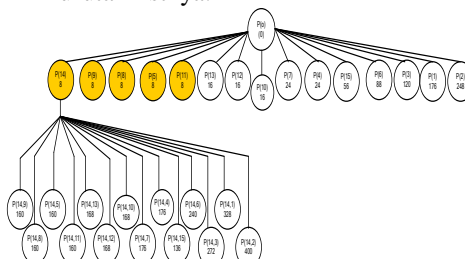
dipilih, yang mempunyai nilai V_s nya lebih kecil diantara node-node yang lain.



Gambar 6 Penentuan Lower Bound Algoritma Branch and Bound Priority Rule SPT Level 1

Dan didapatkan node yang dipilih sebagai lower bound (batas bawah) yaitu node dengan P(14), P(9), P(8), P(5) dan P(11) dengan masing-masing *job* ke 14, 9, 8, 5 dan 11. Yang kemudian dilanjutkan dengan pembuatan node baru yang dicabangkan pada node yang telah dipilih.

- e. Membuat Node Selanjutnya
Pada node baru ini, P(s) yang akan dihitung, akan menggabungkan *job* yang belum terjadwalkan di akhir urutan parsial *job* dan atribut lainnya yang digunakan untuk membuat node baru dan memasukkan mereka sesuai urutan listnya.



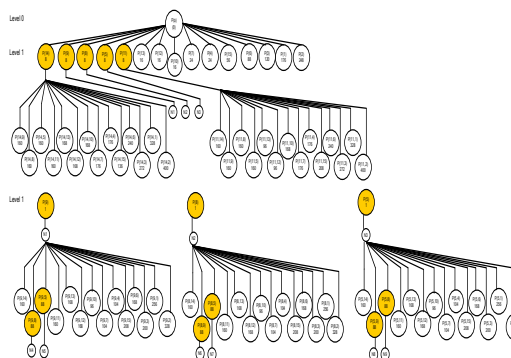
Gambar 7 Penentuan Titik Aktif Algoritma Branch and Bound Priority Rule SPT

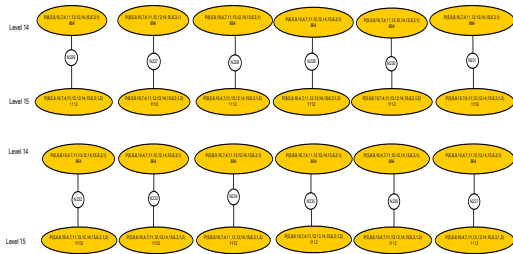
Contoh perhitungan

$$P(s') = P(14,9) = \text{job ke } 14 = 8, \text{ job ke } 9 = 8$$

$$V_{js} = V_s + P(s'); V_{14,9s} = 0 + 72 + 8 + 72 + 8 = 160$$

Pada awal pekerjaan ditambahkan nilai 72 karena proses tersebut membutuhkan waktu *setup* mesin sebesar 72 jam, kemudian antara *job* ke 14 dan *job* ke 9 ditambahkan waktu *setup* kembali sebesar 72 jam karena *job* ke 14 dan *job* ke 9 memiliki diameter pipa yang berbeda sehingga dibutuhkan waktu *setup* mesin sebesar 72 jam.





Gambar 8 Struktur Tree Algoritma Branch and Bound dengan Jadwal Inisial SPT

berikut adalah variasi jadwal yang dihasilkan dari perhitungan algoritma *branch and bound* dengan *priority rule* SPT:

Tabel 4 Variasi Jadwal Keseluruhan Dengan Algoritma Branch and Bound dengan *priority rule* SPT

No	Variasi Jadwal	Urutan Job	Total Lateness (Jam)
1	S1	Job 9-8-5-10-7-4-11-13-12-14-15-6-3-1-2	344
2	S2	Job 9-8-5-10-7-4-11-12-13-14-15-6-3-1-2	344
3	S3	Job 9-8-5-10-4-7-11-13-12-14-15-6-3-1-2	344
4	S4	Job 9-8-5-10-4-7-11-12-13-14-15-6-3-1-2	344
5	S5	Job 9-8-10-7-4-11-13-12-14-15-6-3-1-2	344
6	S6	Job 9-8-10-7-4-11-12-13-14-15-6-3-1-2	344
7	S7	Job 9-8-10-4-7-11-13-12-14-15-6-3-1-2	344
8	S8	Job 9-8-10-4-7-11-12-13-14-15-6-3-1-2	344
9	S9	Job 8-9-5-10-7-4-11-13-12-14-15-6-3-1-2	344
10	S10	Job 8-9-5-10-7-4-11-12-13-14-15-6-3-1-2	344
11	S11	Job 8-9-5-10-4-7-11-13-12-14-15-6-3-1-2	344
12	S12	Job 8-9-5-10-4-7-11-12-13-14-15-6-3-1-2	344
13	S13	Job 8-9-10-7-4-11-13-12-14-15-6-3-1-2	344
14	S14	Job 8-9-10-7-4-11-12-13-14-15-6-3-1-2	344
15	S15	Job 8-9-10-4-7-11-13-12-14-15-6-3-1-2	344
16	S16	Job 8-9-10-4-7-11-12-13-14-15-6-3-1-2	344
17	S17	Job 5-8-9-10-7-4-11-13-12-14-15-6-3-1-2	344
18	S18	Job 5-8-9-10-7-4-11-12-13-14-15-6-3-1-2	344
19	S19	Job 5-8-9-10-4-7-11-13-12-14-15-6-3-1-2	344
20	S20	Job 5-8-9-10-4-7-11-12-13-14-15-6-3-1-2	344
21	S21	Job 5-8-9-10-7-4-11-13-12-14-15-6-3-1-2	344
22	S22	Job 5-8-9-10-7-4-11-12-13-14-15-6-3-1-2	344
23	S23	Job 5-8-9-10-4-7-11-13-12-14-15-6-3-1-2	344
24	S24	Job 5-8-9-10-4-7-11-12-13-14-15-6-3-1-2	344

Didapatkan nilai total *lateness* terkecil adalah untuk semua *sequence* baik *sequence* ke-1 sampai *sequence* ke-24 yaitu dengan nilai 344 jam dan jumlah *tardy job* adalah 2 dari setiap *sequence* tersebut yakni *job* ke-1 dan *job* ke-2. Urutan penjadwalan *job* yang dihasilkan ialah *Job* (9-8-5-10-7-4-11-13-12-14-15-6-3-1-2) karena urutan ini paling mendekati dengan aturan SPT.

3. Algoritma Branch and Bound dengan Jadwal Inisial LDD

Langkah-langkah penjadwalan menggunakan algoritma *branch and bound priority rule* LDD, antara lain :

- Mencari Jadwal Inisial
Sebelumnya jadwal inisial telah dihitung dan didapatkan total *lateness* sebesar 4296 jam dan total *tardy job* sebanyak 8 *job* yaitu *job* ke-1, 3, 4, 6, 7, 9, 10 dan 13 dengan urutan *job* (15-14-11-2-5-8-12-3-4-6-1-13-7-9-10).
- Inisialisasi Parameter
Tempatkan $P(0)$ pada list yang aktif. Nilai yang terkait pada node ini adalah $V_0 = 0$ dan $p(\Phi) = \sum_{j=1}^n P_j$. Dikarenakan nilai pada algoritma *branch and bound* menggunakan integer programming yaitu 1 dan 0. Yang artinya untuk mesin dalam posisi diam diberi tanda 0 dan ketika mesin mengerjakan suatu *job* diberi tanda 1.

$P(0)$ = Waktu proses pada *job* pada mesin dalam keadaan diam, belum berproses.

V_0 = Value pada $P(0)$ dalam keadaan mesin diam.

- Membuat Percabangan
Membuat percabangan $X1=1$ untuk masing-masing *job* i. Ketika mesin mengerjakan suatu *job* diberi tanda 1, $P(s)$, dari list yang aktif dengan node baru untuk masing-masing *job* yang belum terjadwalkan. Kemudian notasikan K sebagai nomor *job* dalam urutan parsial s. Jika $K=n$, maka stop, yang berarti urutan penjadwalan sudah optimal. Jika belum optimal, dilanjutkan dengan langkah 3 dan langkah 4.
 K = level
 n = banyaknya *job* yang akan dijadwalkan.



Gambar 9 Struktur Tree Algoritma Branch and Bound Priority Rule LDD Level 1

Contoh perhitungan :

$P(s') = P(15) = \text{Job ke } 15 = 56$
 $V_{js} = V_s + P(s')$; $V_{15s} = 0 + 56 = 56$
 Kemudian, dilanjutkan ke langkah selanjutnya, dikarenakan level K tidak sama dengan n, maka belum dikatakan optimal.

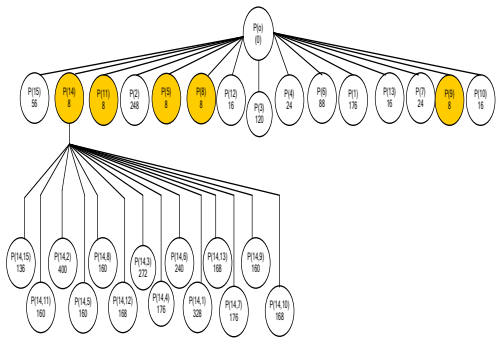
- Menentukan Lower Bound
Dilanjutkan dengan memilih *lower bound* (batas bawah). Node yang akan dipilih, yang mempunyai nilai V_s nya lebih kecil diantara node-node yang lain.



Gambar 10 Penentuan Lower Bound Algoritma Branch and Bound Priority Rule LDD Level 1

Dan didapatkan node yang dipilih sebagai *lower bound* (batas bawah) yaitu node dengan $P(14)$, $P(11)$, $P(5)$, $P(8)$ dan $P(9)$ dengan masing-masing *job* ke 14, 11, 5, 8 dan 9. Yang kemudian dilanjutkan dengan pembuatan node baru yang dicabangkan pada node yang telah dipilih.

- Membuat Node Selanjutnya
Pada node baru ini, $P(s)$ yang akan dihitung, akan menggabungkan *job* yang belum terjadwalkan di akhir urutan parsial *job* dan atribut lainnya yang digunakan untuk membuat node baru dan memasukkan mereka sesuai urutan listnya.



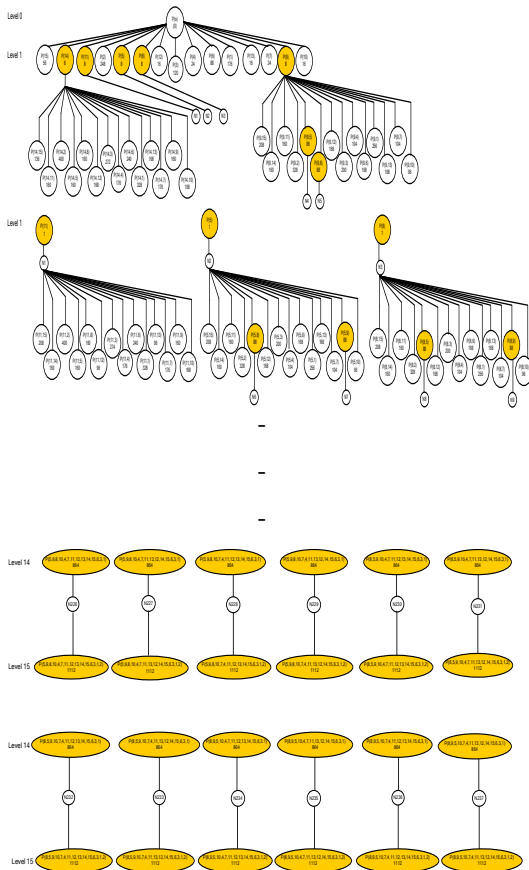
Gambar 11 Penentuan Titik Aktif Algoritma Branch and Bound Priority Rule LDD

Contoh perhitungan

$$P(s^*) = P(14,15) = \text{job ke } 14 = 8, \text{ job ke } 15 = 56$$

$$V_{js} = V_s + P(s^*); V_{14,15s} = 0 + 72 + 8 + 56 = 136$$

Pada awal pekerjaan ditambahkan nilai 72 karena proses tersebut membutuhkan waktu *setup* mesin sebesar 72 jam, kemudian karena *job* ke 14 dan *job* ke 15 memiliki diameter pipa yang sama, maka tidak diperlukan penambahan waktu *setup*. Dari hasil perhitungan tersebut didapatkan node aktif selanjutnya yang kemudian akan dihasilkan variasi jadwal dengan total *lateness* dan *tardy job* yang minimum yang akan dipilih.



Gambar 12 Struktur Tree Algoritma Branch and Bound dengan Jadwal Inisial LDD

berikut adalah variasi jadwal yang dihasilkan dari perhitungan algoritma *branch and bound* dengan *priority rule* LDD:

Tabel 5 Variasi Jadwal Keseluruhan Dengan Algoritma Branch and Bound dengan *priority rule* LDD

No	Variasi Jadwal	Urutan Job	Total <i>Lateness</i> (Jam)
1	S1	Job 9-5-8-10-4-7-11-12-13-14-15-6-3-1-2	344
2	S2	Job 9-5-8-10-4-7-11-12-13-14-15-6-3-1-2	344
3	S3	Job 9-5-8-10-7-4-11-12-13-14-15-6-3-1-2	344
4	S4	Job 9-5-8-10-7-4-11-12-13-14-15-6-3-1-2	344
5	S5	Job 9-5-8-10-4-7-11-12-13-14-15-6-3-1-2	344
6	S6	Job 9-5-8-10-4-7-11-12-13-14-15-6-3-1-2	344
7	S7	Job 9-5-8-10-7-4-11-12-13-14-15-6-3-1-2	344
8	S8	Job 9-5-8-10-7-4-11-12-13-14-15-6-3-1-2	344
9	S9	Job 5-8-9-10-4-7-11-12-13-14-15-6-3-1-2	344
10	S10	Job 5-8-9-10-4-7-11-12-13-14-15-6-3-1-2	344
11	S11	Job 5-8-9-10-7-4-11-12-13-14-15-6-3-1-2	344
12	S12	Job 5-8-9-10-7-4-11-12-13-14-15-6-3-1-2	344
13	S13	Job 9-8-9-10-4-7-11-12-13-14-15-6-3-1-2	344
14	S14	Job 9-8-9-10-4-7-11-12-13-14-15-6-3-1-2	344
15	S15	Job 5-8-9-10-7-4-11-12-13-14-15-6-3-1-2	344
16	S16	Job 5-8-9-10-7-4-11-12-13-14-15-6-3-1-2	344
17	S17	Job 8-5-9-10-4-7-11-12-13-14-15-6-3-1-2	344
18	S18	Job 8-5-9-10-4-7-11-12-13-14-15-6-3-1-2	344
19	S19	Job 8-5-9-10-7-4-11-12-13-14-15-6-3-1-2	344
20	S20	Job 8-5-9-10-7-4-11-12-13-14-15-6-3-1-2	344
21	S21	Job 8-5-9-10-4-7-11-12-13-14-15-6-3-1-2	344
22	S22	Job 8-5-9-10-4-7-11-12-13-14-15-6-3-1-2	344
23	S23	Job 8-5-9-10-7-4-11-12-13-14-15-6-3-1-2	344
24	S24	Job 8-5-9-10-7-4-11-12-13-14-15-6-3-1-2	344

Didapatkan nilai total *lateness* terkecil adalah untuk semua *sequence* baik *sequence* ke-1 sampai *sequence* ke-24 yaitu dengan nilai 344 jam dan jumlah *tardy job* adalah 2 dari setiap *sequence* tersebut yakni *job* ke-1 dan *job* ke-2. Urutan penjadwalan *job* yang dihasilkan ialah *job* (5-8-9-10-4-7-11-12-13-14-15-6-3-1-2) karena urutan ini paling mendekati dengan aturan LDD.

Dari hasil perhitungan yang telah dilakukan, maka dapat diperoleh nilai total *lateness* dan jumlah *tardy job* dari masing-masing metode serta jadwal *job* terbaik, dan jadwal *job* terbaik dapat menjadi usulan bagi perusahaan. Berikut ini adalah nilai kriteria dari masing-masing metode :

Tabel 6 Perbandingan Nilai Kriteria Jadwal Inisial

Jadwal Inisial	Nilai Kriteria			
	Total <i>Lateness</i>	Jumlah <i>Tardy Job</i>	Urutan Job	
Existing	1616 jam	6	6,7,9,10,12,13	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15
EDD	720 jam	6	2,5,8,11,12,14	10-9-7-13-1-3-4-6-12-8-2-5-11-14-15
SPT	880 jam	4	1,2,3,10	14-5-8-9-11-12-13-10-4-7-15-6-3-1-2
LDD	4296 jam	8	1,3,4,6,7,9,10,13	15-14-11-2-5-8-12-3-4-6-1-13-7-9-10

Tabel 7 Perbandingan Nilai Kriteria Metode Penjadwalan

Metode	Nilai Kriteria			
	Total <i>Lateness</i>	Jumlah <i>Tardy Job</i>	Urutan Job	
Existing	1616 jam	6	6,7,9,10,12,13	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15
Algoritma Branch and Bound dengan jadwal inisial EDD	344 jam	2	1,2	9-8-5-10-7-4-11-13-12-14-15-6-3-1-2
Algoritma Branch and Bound dengan jadwal inisial SPT	344 jam	2	1,2	9-8-5-10-7-4-11-13-12-14-15-6-3-1-2
Algoritma Branch and Bound dengan jadwal inisial LDD	344 jam	2	1,2	5-8-9-10-4-7-11-12-13-14-15-6-3-1-2

Dari hasil perhitungan dengan metode algoritma *branch and bound* menggunakan *priority rule* EDD, SPT dan LDD didapatkan hasil yaitu 24 alternatif *sequencing* yang sama dengan total *lateness* sebesar 344 jam dan 2 *tardy job* yaitu *job* ke-1 dan 2 pada tiap-tiap alternatif *sequencing*. Hal ini disebabkan karena metode algoritma *branch and bound* mencari dan menghasilkan solusi yang paling optimal dari solusi-solusi optimal yang ada. Hal ini dibuktikan dengan hasil *sequencing*, total *lateness* dan *tardy job* yang sama meskipun dengan jadwal inisial yang berbeda. Dari 24 alternatif *sequencing* yang diperoleh di pilih alternatif *sequencing* yang dihasilkan oleh metode algoritma *branch and bound* dengan jadwal inisial EDD dan SPT. Hal ini dikarenakan jadwal inisial EDD terbukti menghasilkan jadwal alternatif dengan total *lateness* yang minimum sedangkan jadwal

inisial SPT menghasilkan jadwal alternatif dengan jumlah *tardy job* yang minimum. Oleh karena itu alternatif jadwal *sequencing* dengan urutan *job* (9-8-5-10-7-4-11-13-12-14-15-6-3-1-2) dipilih sebagai alternatif jadwal *sequencing* terbaik.

KESIMPULAN

Kesimpulan yang didapatkan dari hasil penelitian yang telah dilakukan ini adalah sebagai berikut: penjadwalan menggunakan algoritma *branch and bound* dengan *priority rule* EDD memberikan nilai total *lateness* 344 jam dan 2 *tardy job* yaitu *job* ke-1 dan ke-2 dengan urutan *job* (9-8-5-10-7-4-11-13-12-14-15-6-3-1-2). Penjadwalan algoritma *branch and bound* dengan *priority rule* SPT memberikan nilai total *lateness* 344 jam dan 2 *tardy job* yaitu *job* ke-1 dan ke-2 dengan urutan *job* (9-8-5-10-7-4-11-13-12-14-15-6-3-1-2) dan penjadwalan algoritma *branch and bound* dengan *priority rule* LDD memberikan nilai total *lateness* 344 jam dan 2 *tardy job* yaitu *job* ke-1 dan ke-2 dengan urutan *job* (5-8-9-10-4-7-11-12-13-14-15-6-3-1-2). Dan untuk kriteria minimasi total *lateness* dan *tardy job* metode algoritma *branch and bound* dengan *priority rule* EDD atau SPT atau LDD memberikan nilai total *lateness* dan *tardy job* yang lebih kecil yaitu 344 jam dan 2 *tardy job* (*job* ke-1 dan *job* ke-2) dibandingkan dengan jadwal *existing* yang memiliki nilai total *lateness* 1616 jam dan 6 *tardy job*. Untuk kriteria minimasi total *lateness* jadwal inisial EDD terbukti menghasilkan jadwal dengan total *lateness* yang minimum, sedangkan untuk kriteria minimasi jumlah *tardy job* jadwal inisial SPT terbukti menghasilkan jadwal dengan jumlah *tardy job* yang minimum. Sehingga jadwal yang terpilih sebagai jadwal terbaik merupakan jadwal dari algoritma *branch and bound* dengan jadwal inisial EDD dan algoritma *branch and bound* dengan jadwal inisial SPT dengan urutan *job* (9-8-5-10-7-4-11-13-12-14-15-6-3-1-2).

DAFTAR PUSTAKA

- Arifin, M., dan Rudyanto, A. 2010. *Perancangan Sistem Informasi Penjadwalan Produksi Paving Block Pada CV. Eko Joyo. Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010)*. Yogyakarta, 19 Juni 2010.
- Baker, K. R. 1974. *Introduction to Sequencing dan Scheduling*. John Wiley dan Sons Inc. New York.
- Brucker, P. 2007. *Scheduling Algorithms Fifth Edition*. Springer-Verlag Berlin Heidelberg, Germany.
- Conway, R. W., et. al., 1967. *Theory of Scheduling*. MA: Addison-Wesley.
- Ginting, R. 2007. *Sistem Produksi*. Graha Ilmu. Yogyakarta.
- Kurniati, A. 2013. *Penjadwalan Produk Painted di PT. ABC dengan Algoritma Branch and*

Bound & Neighborhood Search Untuk Meminimasi Mean Flow Time. Skripsi. Jurusan Teknik Industri. Universitas Sultan Ageng Tirtayasa. Cilegon.

- Lamabelawa, M. 2006. *Implementasi Program Pascal Untuk Optimisasi Penjadwalan Pekerjaan Secara Kelompok Pada Mesin Tunggal Dengan Algoritma Lawler*. Kupang.
- Nasution, A., et. al. 2003. *Perencanaan & Pengendalian Produksi*. Guna Widya. Surabaya.
- Pinedo, M., 2004. *Planning dan Scheduling in Manufacturing dan Services*. Springer. New York.
- Riyanti, E. 2004. *Penerapan Algoritma Branch And Bound Untuk Penentuan Obyek Wisata*. Skripsi. Jurusan Teknik Informatika. Universitas Komputer Indonesia. Bandung.
- Sutanto, G. 2000. *Algoritma Branch And Bound Dan Algoritma Genetika Untuk Penjadwalan Flowshop Dengan Fungsi Tujuan Ganda*. Skripsi Jurusan Teknik Industri Fakultas Teknik Universitas Kristen Petra : Surabaya.
- Sutanto, J., et. al. 2004. *Algoritma Branch and Bound untuk Masalah Penjadwalan pada Mesin Paralel*. Jurnal Teknik Informatika. Laboratorium Ilmu dan Rekayasa Komputasi. Departemen Teknik Informatika : ITB. Bandung.
- Suyanto. 2010. *Algoritma Optimasi Deterministik atau Probabilistik*. Graha Ilmu. Yogyakarta.
- Zai, R. N., 2010. *Flow Shop Scheduling Problem Dengan Menggunakan Hybrid Heuristic*. Skripsi. Jurusan Teknik Industri. Universitas Sultan Ageng Tirtayasa. Cilegon.