

Model Penjadwalan *Batch* Multi Item dengan *Dependent Processing Time*

Sukoyo¹, TMA Ari Samadhi², Bermawi P. Iskandar³, Abdul Hakim Halim⁴

Abstract: This paper investigates a development of single machine batch scheduling for multi items with dependent processing time. The batch scheduling problem is to determine simultaneously number of batch (N), which item and its size allocated for each batch, and processing sequences of resulting batches. We use total actual flow time as the objective of schedule performance. The multi item batch scheduling problem could be formulated into a biner-integer nonlinear programming model because the number of batch should be in integer value, the allocation of items to resulting batch need binary values, and also there are some non-linearity on objective function and constraint due to the dependent processing time. By applying relaxation on the decision variable of number of batch (N) as parameter, a heuristic procedure could be applied to find solution of the single machine batch scheduling problem for multi items.

Keywords: Batch scheduling, multi items, dependent processing time.

Pendahuluan

Permasalahan penjadwalan ditemukan ketika ada n buah *job* yang harus dikerjakan menggunakan satu mesin secara bergantian. Hal ini dikarenakan hanya sebuah *job* yang dapat dikerjakan setiap saat, untuk itu perlu ditentukan urutan pengerjaan *job* yang sesuai dengan kriteria performansi penjadwalan yang ditetapkan. Jika waktu proses setiap *job* sudah diketahui dan tidak tergantung pada urutan pengerjaan, terdapat beberapa aturan penjadwalan sudah tersedia untuk mesin tunggal (Nahmias [8]; Sipper dan Bullfin, Jr. [9]). Misalnya: aturan SPT (*shortest processing time*) akan meminimumkan rata-rata waktu tinggal (*mean flowtime*) penyelesaian semua *job*, atau aturan EDD (*earliest due date*) jika ingin meminimumkan maksimum waktu keterlambatan (*tardiness*). Permasalahan penjadwalan *batch* mempunyai karakteristik yang berbeda dengan penjadwalan *job*, yaitu pengerjaan setiap *job* dapat dilakukan dalam beberapa bagian, yang didefinisikan sebagai *batch*. Akibatnya jumlah *job* dan waktu pengerjaan *job* berubah mengikuti langkah pembagian *job* menjadi *batch*. Ini artinya persoalan penjadwalan menjadi lebih kompleks, yaitu mencari jumlah pembagian *job* menjadi *batch*, ukuran setiap *batch*, dan mencari urutan pengerjaan dari *batch* yang dihasilkan. Perbedaan karakteristik yang mendasar ini menyebabkan aturan penjadwalan *job*

tidak dapat digunakan langsung untuk persoalan penjadwalan *batch*.

Penelitian tentang penjadwalan *batch* untuk item tunggal yang diproses dengan mesin tunggal dapat ditemukan pada Dobson [3]. Kriteria waktu tinggal (*flowtime*) *part* menjadi ukuran performansi penjadwalan. Penjadwalan *batch* disusun dengan pendekatan penjadwalan maju dengan semua *job* diasumsikan sudah siap dikerjakan pada saat awal penjadwalan tanpa mempertimbangkan saat penyerahan (*due date*) yang diinginkan oleh konsumen. Penggunaan waktu tinggal ini tidak sepenuhnya sesuai dengan kenyataannya karena saat kedatangan *batch* untuk dikerjakan dapat dikendalikan menyesuaikan dengan saat penyerahannya. Mengatasi kelemahan kriteria waktu tinggal pada penjadwalan *batch* ini, Halim [4] mengembangkan kriteria waktu tinggal aktual, yaitu interval waktu mulai saat kedatangan sampai dengan saat penyerahan. Halim [4] juga mengembangkan model penjadwalan *batch* untuk multi item yang diproses pada mesin tunggal. Pada penelitian Dobson [3] dan Halim [4], waktu proses juga diasumsikan konstan.

Waktu proses kadangkala tidak selalu konstan. Hal ini bisa ditemukan misalnya pada proses pengolahan baja proses pengerolan *slab* baja untuk memproduksi plat baja (*coil*) (Chen *et al.* [2]). Proses pengerolan diawali dengan proses pemanasan awal (*preheating*). Waktu pemanasan awal bergantung pada temperatur awal dari *slab*. Apabila proses pengerolan dilakukan segera setelah *slab* dihasilkan, proses pengerolan tidak memerlukan proses pemanasan awal karena temperatur *slab* sudah sesuai persyaratan. Namun jika setelah *slab*

^{1,2,3,4} Fakultas Teknologi Industri, KK Sistem Manufaktur, Institut Teknologi Bandung. Gd. Labtek III, Jl. Ganesha 10. Bandung 40132. Email: sukoyo@mail.ti.itb.ac.id

Naskah masuk 26 Agustus 2010; revisi1 07 Oktober 2010; revisi2 29 Oktober 2010; diterima untuk dipublikasikan 9 November 2010.

dihasilkan tidak langsung masuk ke proses pengerolan, maka temperatur *slab* secara alami turun. Semakin lama waktu menunggu *slab* untuk masuk ke proses pengerolan, temperaturnya semakin rendah. Ketika proses pengerolan dimulai, *slab* tersebut membutuhkan proses pemanasan awal sehingga waktu pengerolan menjadi lebih lama. Penjelasan ini menunjukkan bahwa waktu proses dipengaruhi oleh waktu tunggu. Dalam penelitian ini, fenomena waktu proses tersebut didefinisikan sebagai waktu proses bersyarat (*dependent processing time*). Pada penelitian lain fenomena ini didefinisikan sebagai *deteriorating jobs* (Wang dan Xia [11]; Chen et al. [2]; Jeng dan Lin [5]; Musheiov [7]).

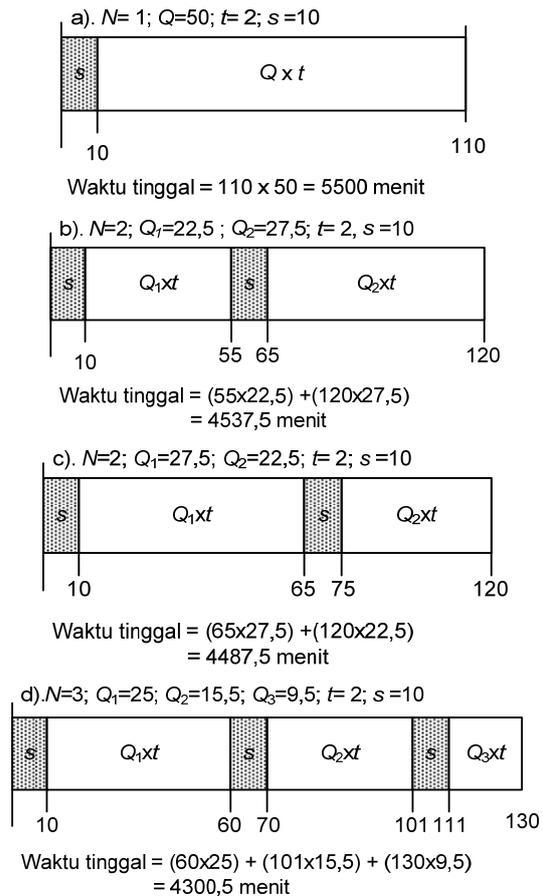
Metode Penelitian

Penjadwalan *Batch*

Untuk mengerjakan satu *job* dapat dibagi dalam beberapa bagian, yang disebut dengan *batch*. Misalnya ada satu *job* item A sebanyak 50 unit yang dikerjakan dengan satu mesin. Waktu proses (t) adalah 2 menit/unit dan waktu *setup* (s) adalah 20 menit. Ada beberapa alternatif pembagian *job* menjadi beberapa *batch* yang mungkin dilakukan, seperti yang ditunjukkan pada Gambar 1. Setiap alternatif memberikan dampak pada waktu tinggal, yaitu perkalian antara interval mulai dari saat kedatangan *batch* sampai saat penyerahan (*due date*) dengan ukuran *batch* (Q). Pada ilustrasi Gambar 1 diasumsikan bahwa saat kedatangan semua *batch* adalah saat nol dan saat penyerahan (*due date*) sama dengan saat selesai pengerjaan *batch*. Dari contoh ilustrasi tersebut terlihat bahwa jumlah *batch* (N), ukuran *batch* (Q), dan urutan pengerjaan *batch* adalah variabel keputusan dalam penjadwalan *batch*.

Untuk pembahasan penjadwalan *batch* multi item ini, notasi berikut digunakan dalam makalah ini:

- d : waktu penyerahan (*due date*)
- i, j, k : indeks urutan *batch* : 1,2,..., N
- g : indeks item: 1,2,..., G
- G : jumlah item
- N : jumlah *batch*
- $B_{[i]}$: saat mulai pengerjaan *batch* ke- i
- $c_{[i]}$: saat selesai *batch* ke- i
- D_g : jumlah permintaan item g (unit)
- F^a : waktu tinggal aktual total
- $F_{[i]}^a$: waktu tinggal aktual *batch* ke- i
- $L_{[i]g}$: nilai prioritas *batch* ke- i berisi item g
- $Q_{[i]}$: ukuran *batch* untuk *batch* ke- i

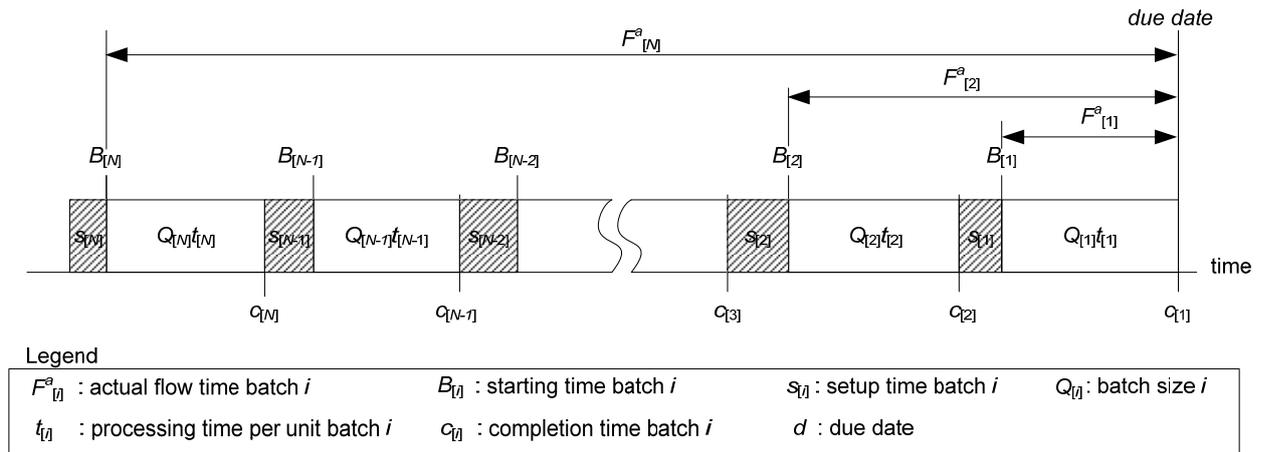


Gambar 1. Ilustrasi penjadwalan *batch*

- $s_{[i]}$: waktu *setup* untuk *batch* ke- i
- S_g : waktu *setup* per *batch* berisi item g (satuan waktu /*batch*)
- $t_{[i]}$: waktu pengerjaan per unit untuk *batch* ke- i
- T_g : waktu proses standar per unit dari item g (satuan waktu /unit)
- $x_{[i]g}$: variabel biner (0 atau 1) untuk menyatakan *batch* ke- i berisi item g atau item yang lain
- δ_g : laju peningkatan waktu proses setiap unit waktu tunggu *batch* yang berisi item g (satuan waktu /unit-waktu tunggu)

Waktu Tinggal Aktual

Pada penjadwalan *batch*, pengerjaan beberapa item yang harus diproses dibagi dalam sejumlah *batch*. Satu *batch* hanya berisi item yang sejenis. Jika ada N buah *batch*, maka kita dapat menyusun penjadwalan *batch* seperti pada Gambar 2. *Batch* diberi indeks urutan dari arah *due date* (*backward scheduling*), dan saat penyelesaian *batch* yang diproses paling akhir adalah tepat pada saat penyerahan (*due date*) yang diinginkan oleh konsumen.



Gambar 2. Gantt chart penjadwalan *batch*

Halim [4] mengembangkan ukuran performansi penjadwalan *batch* yaitu waktu tinggal aktual yang didefinisikan sebagai interval waktu mulai saat kedatangan (*arrival time*) sampai dengan saat penyerahan (*due date*). Pada kasus penjadwalan mundur maka waktu tinggal aktual *batch* ke- i ($F_{[i]}^a$), $i=1,2,\dots,N$ dapat dinyatakan dalam formula:

$$F_{[i]}^a = (d - B_{[N]})Q_{[N]} - (\sum_{j=1}^i (Q_{[j]}t_{[j]} + s_{[j]}) - s_{[i]})Q_{[i]} \quad (1)$$

Waktu tinggal total dari penjadwalan *batch* adalah penjumlahan waktu tinggal semua *batch*. Jadi waktu tinggal total dari N buah *batch* dapat dinyatakan dalam persamaan sebagai berikut:

$$F^a = \sum_{i=1}^N F_{[i]}^a = \sum_{i=1}^N (d - B_{[N]})Q_{[N]} - (\sum_{j=1}^i (Q_{[j]}t_{[j]} + s_{[j]}) - s_{[i]})Q_{[i]} \quad (2)$$

Persamaan (2) memperlihatkan bahwa waktu tinggal aktual dipengaruhi oleh keputusan tentang jumlah *batch* (N) dan ukuran *batch* ($Q_{[i]}$). Sedangkan keputusan pengaturan urutan pengerjaan *batch* sudah direpresentasikan oleh indeks *batch*.

Pada penjadwalan *batch*, item-item yang harus diproduksi dialokasikan pada N buah *batch* yang ditetapkan. Misalkan jumlah item adalah G dengan permintaan setiap item adalah D_g ($g = 1,2,\dots,G$) dan saat penyerahan (*due date*) semua item sama yaitu d . Jika ditetapkan ada N buah *batch* ($N \geq G$) dan setiap *batch* harus berisi item sejenis, maka dibutuhkan keputusan biner untuk menyatakan jenis item yang dialokasikan pada urutan *batch* ke- i ($x_{[i]g}$), yaitu:

$$x_{[i]g} = \begin{cases} 1, & \text{jika } \textit{batch} \text{ ke-} i \text{ berisi item } g \\ 0, & \text{lainnya} \end{cases} \quad (3)$$

Jenis item yang berada di *batch* ke- i menentukan waktu proses ($t_{[i]}$) dan waktu *setup* ($s_{[i]}$) untuk *batch* tersebut.

Dependent processing time

Kondisi *dependent processing time*, yaitu waktu proses per unit tidak lagi tetap, dapat mempengaruhi waktu tinggal aktual *batch*. Waktu proses per unit dalam satu *batch* dipengaruhi oleh lama waktu tunggu mulai saat penyelesaian *batch* sampai dengan saat penyerahannya (*due date*). Sukoyo et al. [10] mengembangkan model *dependent processing time* dengan fenomena waktu proses per unit yang meningkat pada item tunggal. Pendekatan model *dependent processing time* ini diadopsi untuk situasi multi item.

Dalam makalah ini waktu proses per unit meningkat seiring dengan waktu tunggu dari saat penyelesaian *batch* sampai dengan saat penyerahan. Untuk mencari bentuk umum waktu proses *batch*, berikut ini adalah formula waktu proses per unit untuk *batch* ke-1 ($t_{[1]}$), *batch* ke-2 ($t_{[2]}$), dan *batch* ke-3 ($t_{[3]}$):

$$t_{[1]} = \sum_{g=1}^G x_{[1]g} (T_g + \delta_g(d - c_{[1]})) = \sum_{g=1}^G x_{[1]g} T_g \quad (4)$$

$$t_{[2]} = \sum_{g=1}^G x_{[2]g} (T_g + \delta_g(d - c_{[2]})) = \sum_{g=1}^G x_{[2]g} (T_g + \delta_g(Q_{[1]}t_{[1]} + s_{[1]})) \quad (5)$$

$$t_{[3]} = \sum_{g=1}^G x_{[3]g} (T_g + \delta_g(d - c_{[3]})) = \sum_{g=1}^G x_{[3]g} (T_g + \delta_g(Q_{[1]}t_{[1]} + Q_{[2]}t_{[2]} + s_{[1]} + s_{[2]})) = \sum_{g=1}^G x_{[3]g} (T_g + \delta_g(\sum_{k=1}^2 Q_{[k]}t_{[k]} + s_{[k]})) \quad (6)$$

Berdasarkan formula waktu proses per unit dari 3 (tiga) *batch* yang pertama di atas, maka dapat

diperoleh formulasi umum waktu proses per unit dari *batch* ke-*i*, yaitu:

$$t_{[i]} = \sum_{g=1}^G x_{[i]g} \left(T_g + \delta_g \left(\sum_{k=1}^{i-1} Q_{[k]} t_{[k]} + s_{[k]} \right) \right) \quad (7)$$

dengan $t_{[0]} = 0$, $Q_{[0]} = 0$, $s_{[0]} = 0$, dan $i = 1, 2, \dots, N$

Jika variabel keputusan biner ($x_{[i]g}$) untuk multi item pada persamaan (7) dihilangkan karena itemnya hanya satu, maka persamaan ini menjadi persamaan waktu proses per unit untuk item tunggal.

Formulasi waktu proses yang meningkat yang ditunjukkan pada persamaan (7) akan dipergunakan dalam penelitian penjadwalan *batch* multi item ini. Persamaan (7) ini merupakan fungsi rekursif, yaitu waktu proses per unit pada satu *batch* membutuhkan waktu proses, ukuran *batch*, dan waktu *setup* dari *batch* yang ada pada urutan sebelumnya. Jika persamaan persamaan (7) disubstitusikan ke persamaan (2) dapat diperoleh formula waktu tinggal aktual untuk N buah *batch* dengan waktu proses yang meningkat seperti berikut ini:

$$F^a = \sum_{i=1}^N \left(\sum_{j=1}^i \left(Q_{[j]} \left(\sum_{g=1}^G x_{[i]g} \left(T_g + \delta_g \left(\sum_{k=1}^{i-1} Q_{[k]} t_{[k]} + s_{[k]} \right) \right) \right) + s_{[j]} \right) - s_{[i]} \right) Q_{[j]} \quad (8)$$

Waktu tinggal aktual dengan waktu proses yang meningkat ini yang digunakan sebagai kriteria penjadwalan *batch* pada mesin tunggal yang mengerjakan permintaan multi item dengan saat penyerahan yang sama.

Hasil dan Pembahasan

Formulasi Model

Formulasi permasalahan penjadwalan *batch* multi item dengan waktu proses yang meningkat pada mesin tunggal ini menggunakan asumsi sebagai berikut: a). Saat penyerahan semua item pada waktu yang sama (*common due date*), b). Peningkatan waktu proses per unit pada satu *batch* adalah proporsional dengan lama waktu tunggu mulai dari saat *batch* selesai dikerjakan sampai dengan saat penyerahan (*due date*), c) Saat awal penjadwalan ditetapkan sama dengan nol, d) Semua item dapat dibagi sehingga ukuran *batch* adalah bilangan riil (kontinu) positif, e) Setiap *batch* hanya berisi item sejenis, f) Waktu proses *batch* tidak dapat diinterupsi, g) Saat kedatangan satu *batch* ke mesin dapat dikendalikan, sehingga saat kedatangannya adalah sama dengan saat mulai pengerjaan *batch*.

Permasalahan perjadwalan *batch* pada mesin tunggal untuk mengerjakan multi item dengan

waktu proses yang meningkat dapat diformulasikan dalam persamaan sebagai berikut:

$$\text{Minimum } \sum_{i=1}^N \left(\sum_{j=1}^i (Q_{[j]} t_{[j]} + s_{[j]}) - s_{[i]} \right) Q_{[i]} \quad (9)$$

dengan fungsi pembatas

$$\sum_{i=1}^N (Q_{[i]} t_{[i]} + s_{[i]}) - s_{[N]} \leq d \quad (10)$$

$$B_{[1]} + Q_{[1]} t_{[1]} = d \quad (11)$$

$$\sum_{i=1}^N x_{[i]g} Q_{[i]} = D_g, \text{ untuk } g = 1, 2, \dots, G \quad (12)$$

$$\sum_{g=1}^G x_{[i]g} = 1 \text{ untuk } i = 1, 2, \dots, N \quad (13)$$

$$t_{[i]} = \sum_{g=1}^G x_{[i]g} \left(T_g + \delta_g \left(\sum_{k=1}^{i-1} Q_{[k]} t_{[k]} + s_{[k]} \right) \right), \text{ untuk } i = 1, 2, \dots, N \quad (14)$$

$$s_{[i]} = \sum_{g=1}^G x_{[i]g} S_g, \text{ untuk } i = 1, 2, \dots, N \quad (15)$$

$$Q_{[0]} = 0, t_{[0]} = 0, \text{ dan } s_{[0]} = 0, \quad (16)$$

$$Q_{[i]} \geq 0 \quad (17)$$

$$N \geq G, \text{ dan } i = 1, 2, \dots, N \quad (18)$$

$$x_{[i]g} = \begin{cases} 1, & \text{jika } \textit{batch} \text{ ke-}i \text{ berisi item } g \\ 0, & \text{lainnya} \end{cases} \quad (19)$$

Persamaan (9) adalah menyatakan fungsi tujuan yaitu meminasi waktu tinggal aktual dari N buah *batch* yang dihitung dari perkalian antara waktu tinggal aktual *batch* dengan ukuran *batch*. Persamaan (10) menyatakan bahwa waktu proses total yang dibutuhkan mengerjakan N buah *batch* tidak lebih dari waktu produksi yang tersedia, yaitu saat penyerahan/*due date* (d) dikurangi dengan saat awal penjadwalan. Kendala supaya saat selesai proses *batch* pertama tepat pada saat penyerahan (*due date*) dinyatakan dalam persamaan (11). Persamaan (12) adalah pembatas bahwa jumlah yang diproduksi untuk setiap jenis item dari semua *batch* harus sama dengan jumlah permintaan item tersebut. Persamaan (13) untuk menyatakan hanya satu jenis item yang diperbolehkan dalam satu *batch*. Waktu proses per unit untuk setiap *batch* dan waktu *setup* untuk setiap *batch* yang tergantung pada jenis item yang ada dalam *batch* tersebut diberikan pada persamaan (14) dan (15). Persamaan (16) menyatakan nilai awal ukuran *batch*, waktu proses, dan waktu *setup* untuk *batch* ke nol. Persamaan (17), (18), dan (19) adalah pembatas terkait dengan variabel keputusan, yaitu keputusan ukuran setiap *batch* adalah bilangan riil positif, jumlah *batch* berupa bilangan integer positif lebih besar sama dengan jumlah item, dan variabel biner untuk menyatakan item yang ada pada satu *batch*.

Hasil formulasi di atas menunjukkan bahwa persoalan penjadwalan *batch* multi item dengan waktu proses yang meningkat merupakan persoalan model program biner-integer-nonlinear. Disamping itu, persoalan penjadwalan ini juga melibatkan persamaan rekursif pada penentuan waktu proses. Oleh sebab itu model penjadwalan *batch* multi item dengan waktu proses yang meningkat tidak dapat dipecahkan langsung. Untuk mencari solusi model program biner-integer-non-linear dari persoalan

penjadwalan *batch* multi item, penelitian ini mengusulkan prosedur heuristik. Jadi penelitian ini selain melakukan pengembangan model penjadwalan *batch* multi item itu, juga mengembangkan prosedur heuristik untuk menyelesaikan model yang dihasilkan.

Pencarian Solusi

Ide dasar untuk menyelesaikan model program biner-integer-nonlinear dari persoalan penjadwalan *batch* multi item dengan waktu proses yang meningkat pada mesin tunggal ini adalah merelaksasi variabel integer N menjadi parameter ($N \geq G$). Apabila variabel N diketahui maka keputusan variabel biner ($x_{[i]g}$), yaitu penentuan jenis item yang ada pada setiap *batch*, dapat dicari dengan menggunakan aturan:

$$\frac{t_{[1]}Q_{[1]}+s_{[1]}}{Q_{[1]}} \leq \frac{t_{[2]}Q_{[2]}+s_{[2]}}{Q_{[2]}} \leq \dots \leq \frac{t_{[N]}Q_{[N]}+s_{[N]}}{Q_{[N]}} \quad (20)$$

Detail pembuktian aturan ini dapat dilihat di Halim [4]. Jika waktu proses per unit ($t_{[i]}$) tidak konstan, hal ini disebabkan oleh kondisi *dependent processing time*, maka satu prosedur heuristik perlu disusun. Hal tersebut di atas perlu dilakukan supaya aturan ini dapat digunakan untuk memecahkan persoalan penjadwalan *batch* multi item dengan waktu proses yang meningkat.

Jika variabel N dan variabel biner ($x_{[i]g}$) diketahui, maka persoalan penjadwalan *batch* multi item pada mesin tunggal dengan waktu proses yang meningkat ini dapat disederhanakan menjadi model program nonlinear untuk meminimasi waktu tinggal aktual dengan variabel keputusannya adalah ukuran *batch* yang optimal. Jika persoalan dapat diubah menjadi model program nonlinear, maka alat bantu perangkat lunak optimasi dapat dipergunakan untuk mencari solusinya.

Pencarian solusi terdiri dari 2 (dua) tahapan utama, yaitu tahap penentuan solusi awal untuk $N = G$, dan tahap perbaikan solusi secara bertahap dengan menambah nilai $N = N + 1$ sampai kriteria penghentian dipenuhi. Penambahan nilai N dihentikan jika alternatif solusi pada N terakhir mempunyai selisih antara saat penyerahan (*due date*) dengan waktu proses total kurang dari minimum waktu *setup batch* dan atau mempunyai *batch* yang kosong (*batch* dengan ukuran nol).

Proses pencarian solusi diawali dengan $N = G$. Jumlah *batch* sama dengan jenis item, maka setiap item hanya mengisi salah satu dari G *batch* yang ada. Untuk menentukan alokasi item ke *batch*,

dilakukan secara bertahap. Mulai dari *batch* ke-1, hitung $L_{[1]g} = (t_{[1]}Q_{[1]} + s_{[1]})/Q_{[1]}$ jika *batch* pertama berisi item g ($g = 1, 2, \dots, G$). Persamaan (14) digunakan untuk menghitung $t_{[i]}$, sedangkan $Q_{[i]} = D_g$ dan $s_{[i]} = S_g$. Untuk item k ($g = 1, 2, \dots, k, \dots, G$) dengan nilai $L_{[1]k}$ terkecil ditetapkan $x_{[1]k} = 1$ dan beberapa item yang lain ($g \neq k$) ditetapkan $x_{[i]g} = 0$. Cara ini diulangi untuk *batch* ke-2 sampai ke- N sehingga diperoleh alokasi item ke setiap *batch* ($x_{[i]g}$). Setelah solusi awal $N, Q_{[i]}$, dan $x_{[i]g}$ untuk $N = G$ diketahui, dilakukan pemeriksaan kelayakan fungsi pembatas persamaan (10). Jika fungsi pembatas ini tidak terpenuhi maka pencarian dihentikan karena tidak ada solusi yang layak. Namun jika fungsi pembatas persamaan (10) terpenuhi, maka nilai $N, Q_{[i]}$, dan $x_{[i]g}$ ini ditetapkan sebagai solusi awal terbaik dan total waktu tinggal aktual dihitung dengan persamaan (9) menggunakan nilai solusi awal terbaik ini.

Selanjutnya nilai N ditambah secara bertahap dengan satu ($N = N + 1$) jika kriteria penghentian belum terpenuhi. Berdasarkan kombinasi jumlah *batch* per item yang layak pada N sebelumnya, identifikasi semua alternatif kombinasi jumlah *batch* per item jika jumlah *batch* bertambah menjadi $N = N + 1$. Alternatif kombinasi ini dibangkitkan dengan cara menambah satu *batch* setiap item pada jumlah *batch* sebelumnya. Contoh: misalkan kombinasi jumlah *batch* per item untuk N sebelumnya ($N = 5$) dengan 4 buah item (1,2,3, dan 4) adalah 2 *batch* untuk item 1, dan satu *batch* untuk item 2, 3, dan 4. Kombinasi jumlah *batch* per item $N = 5$ ini dinyatakan dengan 2-1-1-1. Ketika $N = 5$ ini ditambah satu menjadi $N = 6$, maka alternatif kombinasi jumlah *batch* per item akibat jumlah *batch* bertambah satu adalah menambah *batch* item 1 dari 2 *batch* menjadi 3 *batch* sehingga menghasilkan kombinasi 3-1-1-1, atau menambah *batch* item 2 dari 1 *batch* menjadi 2 *batch* sehingga menghasilkan kombinasi 2-2-1-1, atau menambah *batch* item 3 dari 1 *batch* menjadi 2 *batch* sehingga menghasilkan kombinasi 2-1-2-1, atau menambah *batch* item 4 dari 1 *batch* menjadi 2 *batch* sehingga menghasilkan kombinasi 2-1-1-2. Jadi alternatif kombinasi jumlah *batch* per item dari 2-1-1-1 dari $N = 5$ ketika N berubah menjadi $N = 6$ adalah {3-1-1-1; 2-2-1-1; 2-1-2-1; 2-1-1-2}.

Untuk setiap alternatif kombinasi jumlah *batch* per item, tetapkan ukuran sementara semua *batch*. Gunakan ukuran *batch* dari solusi sebelumnya jika jumlah *batch* per item sama. Jika jumlah *batch* per item berbeda, ukuran sementara semua *batch* dicari dengan menggunakan persamaan (9) sampai (19) dengan $G = 1$, dan $N =$ jumlah *batch*, dan $x_{[i]g} = 1$ ($i = 1, \dots, N$), dan $D =$ jumlah permintaan dari item

yang jumlah berbeda tersebut. Mulai dari *batch* ke-1, hitung $L_{[i]g} = (t_{[i]}Q_{[i]} + s_{[i]})/Q_{[i]}$ jika *batch* pertama berisi item g . Persamaan (14) digunakan untuk menghitung $t_{[i]}$. Untuk item k ($g = 1, 2, \dots, k, \dots, G$) dengan nilai $L_{[i]k}$ terkecil tetapkan $x_{[i]k} = 1$ dan beberapa item yang lain ($g \neq k$) tetapkan $x_{[i]g} = 0$. Ulangi proses ini pada *batch* ke-2 sampai ke- N untuk menentukan alokasi item ke setiap *batch* ($x_{[i]g}$). Berdasarkan pada nilai N dan $x_{[i]g}$, cari $Q_{[i]}$ optimal dan waktu tinggal aktual menggunakan persamaan (9) sampai (19). Periksa kembali nilai $L_{[i]g}$ dan $x_{[i]g}$ menggunakan $Q_{[i]}$ optimal ini. Jika ada perubahan $x_{[i]g}$, hitung kembali $Q_{[i]}$ optimal dan waktu tinggal aktual menggunakan persamaan (9) sampai (19). Jika tidak ada perubahan $x_{[i]g}$, catat nilai $x_{[i]g}$ dan $Q_{[i]}$ optimal untuk nilai N yang diberikan. Lakukan penambahan jumlah *batch* ($N = N + 1$) dan ulangi langkah-langkah pencarian solusi di atas sampai kriteria penghentian pencarian solusi tercapai.

Berdasarkan cara pencarian solusi yang dijelaskan di atas, sebuah prosedur heuristik dikembangkan untuk menyelesaikan model program biner-integer-nonlinear dari persoalan penjadwalan *batch* multi item pada mesin tunggal ini. Pencarian solusi dengan prosedur heuristik ini disusun sebagai berikut:

- a. Penentuan solusi awal:
 - a.1. Tetapkan nilai $N = G$ dan lanjutkan ke langkah (a.2),
 - a.2. Mulai dari $i = 1$ dan lanjutkan ke langkah (a.3),
 - a.3. Hitung $L_{[i]g} = (t_{[i]}Q_{[i]} + s_{[i]})/Q_{[i]}$ untuk $g = 1, 2, \dots, G$ jika *batch* ini berisi item g dengan $t_{[i]}$ dihitung dengan persamaan (14), $Q_{[i]} = D_g$ dan $s_{[i]} = S_g$. Lanjutkan ke langkah (a.4),
 - a.4. Tentukan nilai $x_{[i]g}$ untuk $g = 1, 2, \dots, G$. $x_{[i]g} = 1$ jika item g mempunyai nilai $L_{[i]g}$ paling kecil atau $x_{[i]g} = 0$ jika nilai $L_{[i]g}$ bukan yang paling kecil. Lanjutkan ke langkah (a.5),
 - a.5. Periksa nilai i , jika $i = N$ maka lanjutkan ke langkah (a.6). Jika $i < N$ maka tetapkan $i = i + 1$ dan kembali ke langkah (a.3)
 - a.6. Tentukan nilai $Q_{[i]}$ untuk $i = 1, 2, \dots, N$ dan waktu tinggal aktual dengan persamaan (9) sampai (19) menggunakan nilai $x_{[i]g}$ yang diperoleh dari langkah (a.1 sampai a.4). Lanjutkan ke langkah (a.7),
 - a.7. Jika langkah (a.6) ada solusi layak maka tetapkan N , $Q_{[i]}$ dan $x_{[i]g}$ sebagai solusi awal dan teruskan ke tahap (b). Jika tidak ada solusi layak lanjutkan ke tahap (e).
- b. Pemeriksaan kriteria penghentian iterasi
 - b.1. Periksa setiap alternatif kombinasi jumlah *batch* per item untuk N terakhir, apakah $d - \sum_{i=1}^N (Q_{[i]}t_{[i]} + s_{[i]}) - s_{[N]} \geq$ minimum $\{S_g\}$ untuk $g = 1, 2, \dots, G$. Jika ya, berikan keterangan “Lanjut” dan jika tidak berikan keterangan “Berhenti”. Lanjutkan ke langkah (b.2).
 - b.2. Periksa setiap alternatif kombinasi jumlah *batch* per item dengan keterangan “Lanjut”, apakah alternatif kombinasi ini berisi *batch* dengan ukuran nol ($Q_{[i]} = 0$, $i = 1, 2, \dots, N$). Berikan keterangan “Berhenti” hanya pada alternatif kombinasi item yang mempunyai *batch* dengan ukuran nol. Lanjutkan ke langkah (b.3).
 - b.3. Periksa, apakah semua alternatif kombinasi jumlah *batch* per item pada N terakhir mempunyai keterangan “Berhenti”. Jika ya, maka lanjutkan tahap (d). Jika tidak, maka alternatif kombinasi jumlah *batch* per item dengan keterangan “Lanjut” diberikan status *parent* dan lanjutkan ke tahap (c).
- c. Perbaiki solusi
 - c.1. Tingkatkan $N = N + 1$ dan lanjutkan ke langkah (c.2),
 - c.2. Tentukan himpunan yang berisi semua alternatif kombinasi jumlah *batch* per item untuk mengisi N buah *batch* dengan status *child* berdasarkan kombinasi jumlah *batch* per item dengan status *parent* dari N sebelumnya. Pilih salah satu alternatif dengan status *child* untuk dievaluasi. Lanjutkan ke langkah (c.3),
 - c.3. Dari alternatif kombinasi jumlah item dengan status *child* yang dipilih, tentukan ukuran *batch* sementara dari setiap item. Jika jumlah *batch* dari satu item tidak berubah maka gunakan ukuran *batch* dari solusi sebelumnya. Jika jumlah *batch* dari satu item bertambah, ukuran *batch* ditentukan dengan menggunakan persamaan (9) sampai (19) sebagai persoalan item tunggal dengan $G = 1$, $N =$ jumlah *batch* dari item tersebut, $x_{[i]g} = 1$ ($i = 1, \dots, N$), dan $D =$ jumlah permintaan dari item tersebut. Lanjutkan ke langkah (c.4),
 - c.4. Mulai dari $i = 1$ dan lanjutkan ke langkah (c.5),
 - c.5. Hitung $L_{[i]g} = (t_{[i]}Q_{[i]} + s_{[i]})/Q_{[i]}$ untuk $g = 1, 2, \dots, G$ jika *batch* ini berisi item g dengan $t_{[i]}$ dihitung dengan persamaan (14), $Q_{[i]}$ adalah ukuran *batch* sementara dari langkah (c.3) dan $s_{[i]} = S_g$. Apabila untuk satu item terdapat beberapa *batch* yang tersedia, pilih *batch* dari item tersebut yang

- mempunyai ukuran paling besar dan yang belum dialokasikan ke satu urutan *batch*. Lanjutkan ke langkah (c.6),
- c.6. Berdasarkan dengan nilai $L_{[i]g}$, tentukan nilai $x_{[i]g} = 1$ jika item tersebut mempunyai $L_{[i]g}$ terkecil dan $x_{[i]g} = 0$ untuk item lainnya. Lanjutkan ke langkah (c.7),
 - c.7. Periksa nilai i , jika $i < N$ maka ubah $i = i + 1$ dan kembali ke langkah (c.5). Jika $i = N$ maka lanjutkan ke langkah (c.8),
 - c.8. Hitung $Q_{[i]}$ dan waktu tinggal aktual menggunakan persamaan (9) sampai (19) berdasarkan nilai N dan $x_{[i]g}$. Jika nilai $Q_{[i]}$ tidak sama dengan $Q_{[i]}$ sementara, jadikan $Q_{[i]}$ menjadi $Q_{[i]}$ sementara dan kembali ke langkah (c.4). Jika $Q_{[i]}$ sama dengan $Q_{[i]}$ sementara maka tetapkan $Q_{[i]}$ sebagai $Q_{[i]}$ optimal serta catat $x_{[i]g}$ dan waktu tinggal aktual sebagai solusi terbaik dari alternatif kombinasi jumlah *batch* per item yang dipilih dan lanjutkan ke langkah (c.9),
 - c.9. Periksa apakah semua alternatif kombinasi dengan status *child* pada N terakhir sudah dievaluasi. Jika masih ada alternatif kombinasi dengan status *child* yang belum dievaluasi, maka pilih salah satu alternatif dan kembali ke langkah (c.3). Jika semua alternatif kombinasi dengan status *child* sudah dievaluasi, maka kembali ke tahap (b).
- d. Penentuan solusi terbaik
 - d.1. Kumpulkan semua alternatif kombinasi jumlah *batch* per item untuk mengisi N buah *batch* dengan status *parent*. Lanjutkan ke langkah (d.2),
 - d.2. Dari kumpulan semua alternatif dari langkah (d.1), cari alternatif kombinasi jumlah *batch* per item dengan waktu tinggal terkecil, dan tetapkan nilai N , $Q_{[i]}$, dan $x_{[i]g}$ sebagai solusi terbaik dan pencarian solusi berhenti.
 - e. Tidak ada solusi layak karena waktu yang tersedia tidak cukup untuk memproses permintaan semua item dan pencarian solusi berhenti.

Eksperimen

Berikut ini diberikan satu contoh persoalan penjadwalan multi item pada satu mesin dengan $G = 4$, saat penyerahan/*due date* (d) sebesar 400 satuan waktu. Data permintaan (D_g), waktu proses standar per unit (T_g), laju peningkatan waktu proses (δ_g), dan waktu *setup* setiap *batch* (S_g) diberikan pada Tabel 1.

Tabel 1. Data permintaan, waktu proses dan *setup*

Item	Permintaan (unit)	Waktu proses (menit/unit)	Laju peningkatan waktu proses	Waktu <i>setup batch</i>
1	60	1,1	0,001	20
2	45	1,0	0,002	20
3	50	1,2	0,001	10
4	60	1,0	0,002	10

Hasil enumerasi pencarian solusi dengan menggunakan pendekatan heuristik yang dikembangkan di atas dapat ditunjukkan pada Appendix. Pencarian dimulai dari solusi awal dengan jumlah *batch* setiap item=1 atau kombinasi *batch* per item =1-1-1-1, dan diperoleh waktu tinggal selama 37065,9 dengan urutan item pada *batch* adalah 4,3,1, dan 2 dan ukuran *batch* $Q_{[1]}=60$; $Q_{[2]}=50$; $Q_{[3]}=60$; dan $Q_{[4]}=45$. Terlihat ukuran *batch* terakhir tidak nol dan sisa waktu proses lebih besar dari minimum waktu *setup batch* terkecil, maka iterasi dilanjutkan dan kombinasi jumlah *batch* per item 1-1-1-1 menjadi induk (*parent*) untuk $N=5$. Berdasarkan induk dari $N=4$, maka alternatif kombinasi jumlah *batch* per item untuk $N=5$ adalah 2-1-1-1, 1-2-1-1, 1-1-2-1, dan 1-1-1-2. Alternatif ini kemudian diberi status sebagai anak (*child*). Setiap anak pada $N=5$, kemudian dicari alokasi item pada *batch*, ukuran *batch* yang memberikan waktu tinggal aktual yang minimum. Sama seperti pada $N=4$, jika alternatif anak pada $N=5$ tidak memenuhi kriteria berhenti, maka alternatif-alternatif anak pada $N=5$ berubah menjadi induk (*parent*) untuk $N=6$. Cara ini dilanjutkan sampai kriteria berhenti tercapai.

Dari hasil enumerasi pencarian solusi diketahui bahwa pada $N=8$ semua alternatif kombinasi memenuhi kriteria berhenti, sehingga iterasi pencarian solusi dihentikan. Solusi terbaik dicari dari kombinasi jumlah item dengan status *parent* yang mempunyai waktu tinggal aktual paling kecil. Solusi terbaik dari contoh kasus ini adalah $N=7$ dengan *batch* pertama berisi item 4, *batch* ke-2 berisi item 2, *batch* ke-3 berisi item 1, *batch* ke-4 berisi item 3, *batch* ke-5 berisi item 3, *batch* ke-6 berisi item 1, dan *batch* ke-7 berisi item 3 dan ukuran *batch* adalah $Q_{[1]}= 60$; $Q_{[2]}= 45$; $Q_{[3]}=50,327$; $Q_{[4]}=28,785$; $Q_{[5]}=20,574$; $Q_{[6]}=9,673$; dan $Q_{[7]}=0,641$. Waktu tinggal aktual solusi ini adalah 36615,6 satuan waktu. Gambar 3 memperlihatkan *Gantt chart* penjadwalan dari solusi terbaik ini.

Jika diperhatikan hasil solusi terbaik ini, ada item yang sama yang dikerjakan pada 2 (dua) *batch* yang berurutan, yaitu item 3 dikerjakan pada *batch* ke-4 dan ke-5. Jika kedua *batch* yang berisi item 3 ini disatukan, maka total jumlah *batch* dari 7 berkurang menjadi 6 dengan komposisi: item 1 dibagi menjadi 2 *batch*, item 2 dalam 1 *batch*, item 3 dibagi menjadi 2 *batch*, dan item 4 dalam 1 *batch*.

Alternatif kombinasi jumlah *batch* per item ini adalah 2-1-2-1. Terlihat dari hasil enumerasi pada Appendix, alternatif kombinasi jumlah *batch* per item 2-1-2-1 menghasilkan waktu tinggal aktual terbaik sebesar 36738,3. Ini berarti penggabungan *batch* ke-4 dan ke-5 yang berisi item 3 menghasilkan waktu tinggal lebih besar dari solusi terbaik kombinasi 2-1-3-1 yaitu 36615,6. Apabila jumlah *batch* untuk item ke 3 ditambah satu, maka kombinasi jumlah *batch* per item menjadi 2-1-4-1. Waktu tinggal aktual untuk kombinasi 2-1-4-1 ini adalah 36615,6 atau sama dengan solusi terbaik, namun *batch* yang terakhir (ke-8) yang berisi item 3 adalah *batch* kosong (*batch* dengan ukuran nol). Pemecahan *batch* berisi item 3 tidak dapat memperbaiki waktu tinggal aktual dari solusi terbaik yang ditemukan.

Mekanisme imbal-balik antara lama waktu keberadaan *batch* mulai dari saat kedatangan *batch* untuk diproses sampai saat penyerahan (*due date*) dengan ukuran *batch* (Q) terjadi ketika ada penyatuan atau pemecahan sebuah *batch*. Penyatuan beberapa *batch* dapat menghilangkan waktu *setup* (s) beberapa *batch* yang bergabung sehingga memperpendek lama waktu keberadaan sebagian unit item namun akan memperbesar ukuran *batch* (Q) urutan paling awal dari item yang bergabung. Perubahan ukuran *batch* ini menyebabkan waktu penyelesaian *batch* paling awal dari item yang bergabung tersebut bertambah sehingga ada peningkatan lama waktu keberadaan sebagian unit item awal sebelum penggabungan. Pada kondisi *dependent processing time*, perubahan jumlah *batch* dan ukuran *batch* juga mempengaruhi waktu proses. Jika dilakukan pemecahan *batch* dari item, maka hal sebaliknya yang terjadi. Perubahan ukuran *batch* (Q) dan saat mulai kedatangan *batch* ini yang mempengaruhi waktu tinggal aktual, karena waktu tinggal aktual dihitung dari perkalian antara keduanya. Mekanisme imbal-balik ini bisa menjelaskan jika ada satu item yang diproses dalam beberapa *batch* yang berurutan.

Kualitas solusi dari heuristik ini diukur dengan membandingkan dengan solusi dari pendekatan sampel acak berdasarkan alternatif solusi kombinasi urutan pengerjaan *batch* yang mungkin (Baker [1]). Pada penelitian ini diambil 25 sampel acak solusi layak penjadwalan *batch* untuk contoh kasus yang digunakan. Sampel acak kombinasi urutan pengerjaan *batch* dan hasil perhitungan waktu tinggal aktual dari diberikan pada Tabel 2. Pendekatan sampel acak ini memberikan waktu tinggal aktual minimum sebesar 36639,5. Waktu tinggal aktual dari pendekatan sampel acak ini lebih lama jika dibandingkan dengan solusi pendekatan heuristik yang dikembangkan. Ini berarti solusi hasil pendekatan heuristik yang dikembangkan adalah solusi yang terbaik dari alternatif yang ada.

Simpulan

Karakteristik waktu proses yang meningkat sesuai dengan waktu tunggu *batch* dari saat penyelesaian sampai saat penyerahan (*due-date*) mempengaruhi penentuan jumlah *batch* (N), alokasi item ke *batch* ($x_{[i]g}$), dan ukuran setiap *batch* ($Q_{[i]}$) pada penjadwalan *batch* multi item pada mesin tunggal. Permasalahan penjadwalan ini dapat diformulasikan dalam model program nonlinear yang melibatkan variabel keputusan yang bersifat integer (N) dan variabel biner ($x_{[i]g}$). Persoalan penjadwalan *batch* multi item pada mesin tunggal yang disederhanakan menjadi program nonlinear, dengan melakukan relaksasi pada variabel keputusan N . Hal ini memungkinkan model program biner-integer nonlinear yang dihasilkan dapat dipecahkan melalui pendekatan iterasi dengan prosedur heuristik. Hasil eksperimen menunjukkan bahwa prosedur heuristik yang dikembangkan mampu menemukan solusi terbaik dari persoalan penjadwalan *batch* multi item dengan waktu proses bersyarat pada mesin tunggal.

Daftar Pustaka

1. Baker, K. R., *Introduction to Sequencing and Scheduling*, John Wiley and Sons, Inc., 1974.
2. Chen, T. C. E., Kang, L., and Ng, C. T., Due-Date Assignment and Single Machine Scheduling with Deteriorating Jobs, *Journal of the Operational Research Society*, 55, 2004, pp. 198-203.
3. Dobson, G., Karmakar, U. S., and Rummel, J., Batching to Minimize Flow Times on One Machines, *Management Science*, 33(6), 1989, pp.784-799.
4. Halim, A. H., *Batch Scheduling for Production Under Just In Time Environment*, Desertation, University of Osaka Prefectur, 1993.
5. Jeng, A. A. K., and Lin, B. M. T., Makespan Minimization in Single-Machine Scheduling with Step-Deterioration of Processing Times, *Journal of the Operational Research Society*, 55, 2004, pp.247-256.
6. Morton, T. E., and Pentico, D. W., *Heuristic Scheduling System with Application to Production System and Project Management*, John Wiley and Sons, Inc., Hoboken, NJ, 1993.
7. Mosheiov, G., Multi-Machine Scheduling with Linear Deterioration. *Inform*, 36(4), 1998, pp. 205-214.
8. Nahmias, S., *Production And Operation Analysis*, 6th ed., McGraw-Hill International, 2009.
9. Sipper, D., and Bullfin, Jr., R. L., *Production Planning, Control, and Integration*, McGraw-Hill Companies, Inc., New York, NY, 1997.
10. Sukoyo, Samadhi, T. M. A., Iskandar, B. P., and Halim, A. H., Batch Scheduling for A Single

Machine Processing Parts of A Single Item with Increasing Processing Time to Minimize Total Actual Flow Time, *Proceeding of Asia Pacific Industrial Engineering and Management Systems (APIEMS) Conference*, Bali, Indonesia, 2008.

11. Wang, J. B., and Xia, Z. Q., Flow Shop Scheduling Problems with Deteriorating Jobs under Dominating Machines, *Journal of the Operation Research Society*, 57, 2006. pp. 220-226.

Appendix. Enumerasi pencarian solusi

Status	N	Jml <i>batch</i> per item				AF	Alokasi item pada <i>batch</i>	Ukuran <i>batch</i>	Keterangan
		1	2	3	4				
Awal	4	1	1	1	1	37065,9	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=1; g_{[4]}=2$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=60; Q_{[4]}=45$	Lanjut
P	4	1	1	1	1				
C	5	2	1	1	1	36749,6	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=2; g_{[4]}=1; g_{[5]}=1$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=45; Q_{[4]}=38,49; Q_{[5]}=21,51$	Lanjut
C	5	1	2	1	1	36730,3	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=1; g_{[4]}=2; g_{[5]}=2$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=60; Q_{[4]}=30,346; Q_{[5]}=14,654$	Lanjut
C	5	1	1	2	1	36866,2	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2; g_{[5]}=3$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,033; Q_{[4]}=45; Q_{[5]}=13,967$	Lanjut
C	5	1	1	1	2	37065,9	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=1; g_{[4]}=2; g_{[5]}=4$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=60; Q_{[4]}=45; Q_{[5]}=0$	Berhenti
P	5	2	1	1	1	36749,6			
C	6	3	1	1	1	36726,9	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=2; g_{[4]}=1; g_{[5]}=1; g_{[6]}=1$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=45; Q_{[4]}=36,152; Q_{[5]}=19,268; Q_{[6]}=4,579$	Lanjut
C	6	2	2	1	1	36749,6	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=2; g_{[4]}=1; g_{[5]}=1; g_{[6]}=2$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=45; Q_{[4]}=38,477; Q_{[5]}=21,523; Q_{[6]}=0$	Berhenti
C	6	2	1	2	1	36738,3	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=2; g_{[4]}=1; g_{[5]}=1; g_{[6]}=3$	$Q_{[1]}=60; Q_{[2]}=48,565; Q_{[3]}=45; Q_{[4]}=38,528; Q_{[5]}=21,472; Q_{[6]}=1,435$	Lanjut
C	6	2	1	1	2	36749,6	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=2; g_{[4]}=1; g_{[5]}=1; g_{[6]}=4$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=45; Q_{[4]}=38,49; Q_{[5]}=21,51; Q_{[6]}=0$	Lanjut
P	5	1	2	1	1	36730,3			
C	6	2	2	1	1	36749,6	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=2; g_{[4]}=1; g_{[5]}=1; g_{[6]}=2$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=45; Q_{[4]}=38,477; Q_{[5]}=21,523; Q_{[6]}=0$	Berhenti
C	6	1	3	1	1	36727,9	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=1; g_{[4]}=2; g_{[5]}=2; g_{[6]}=2$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=60; Q_{[4]}=29,639; Q_{[5]}=13,986; Q_{[6]}=1,375$	Lanjut
C	6	1	2	2	1	36685,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2; g_{[5]}=3; g_{[6]}=2$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,642; Q_{[4]}=34,322; Q_{[5]}=13,358; Q_{[6]}=10,678$	Lanjut
C	6	1	2	1	2	36709,3	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=1; g_{[4]}=2; g_{[5]}=2; g_{[6]}=4$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=60; Q_{[4]}=30,351; Q_{[5]}=14,649; Q_{[6]}=0$	Berhenti
P	5	1	1	2	1	36866,2			
C	6	2	1	2	1	36738,3	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=2; g_{[4]}=1; g_{[5]}=1; g_{[6]}=3$	$Q_{[1]}=60; Q_{[2]}=48,565; Q_{[3]}=45; Q_{[4]}=38,528; Q_{[5]}=21,472; Q_{[6]}=1,435$	Lanjut
C	6	1	2	2	1	36685,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2; g_{[5]}=3; g_{[6]}=2$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,642; Q_{[4]}=34,322; Q_{[5]}=13,358; Q_{[6]}=10,678$	Lanjut
C	6	1	1	3	1	36731,9	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=2; g_{[4]}=3; g_{[5]}=3; g_{[6]}=3$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=45; Q_{[4]}=24,254; Q_{[5]}=16,413; Q_{[6]}=9,333$	Lanjut
C	6	1	1	2	2	36866,2	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2; g_{[5]}=3; g_{[6]}=4$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,033; Q_{[4]}=45; Q_{[5]}=13,967; Q_{[6]}=0$	Berhenti
P	6	3	1	1	1	36726,9			
C	7	4	1	1	1	36726,9	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=2; g_{[4]}=1; g_{[5]}=1; g_{[6]}=1; g_{[7]}=1$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=45; Q_{[4]}=36,152; Q_{[5]}=19,266; Q_{[6]}=4,583; Q_{[7]}=0$	Berhenti
C	7	3	2	1	1	36726,9	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=2; g_{[4]}=1; g_{[5]}=1; g_{[6]}=1; g_{[7]}=2$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=45; Q_{[4]}=36,138; Q_{[5]}=19,281; Q_{[6]}=4,581; Q_{[7]}=0$	Berhenti
C	7	3	1	2	1	36621,7	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=3; g_{[4]}=1; g_{[5]}=3; g_{[6]}=1; g_{[7]}=1$	$g_{[1]}=60; g_{[2]}=45; g_{[3]}=35,976; g_{[4]}=42,735; g_{[5]}=14,024; g_{[6]}=15,795; g_{[7]}=1,47$	Lanjut
C	7	3	1	1	2	36700,6	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=2; g_{[4]}=1; g_{[5]}=1; g_{[6]}=1; g_{[7]}=4$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=45; Q_{[4]}=36,188; Q_{[5]}=19,279; Q_{[6]}=4,533; Q_{[7]}=0$	Berhenti

Appendix. Enumerasi pencarian solusi (continued)

Status	N	Jml <i>batch</i> per item				AF	Alokasi item pada <i>batch</i>	Ukuran <i>batch</i>	Keterangan
		1	2	3	4				
P	6	2	1	2	1	36738,3	cek		
C	7	3	1	2	1	36621,7	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=3; g_{[4]}=1;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=1$	$Q_{[1]}=60; Q_{[2]}=45; Q_{[3]}=35,976;$ $Q_{[4]}=42,735; Q_{[5]}=14,024;$ $Q_{[6]}=15,795; Q_{[7]}=1,47$	Lanjut
C	7	2	2	2	1	36722,5	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=3;$ $g_{[5]}=2; g_{[6]}=2; g_{[7]}=1$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=29,63;$ $Q_{[4]}=20,37; Q_{[5]}=30,229;$ $Q_{[6]}=14,771; Q_{[7]}=0$	Berhenti
C	7	2	1	3	1	36615,6	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=1; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=3$	$Q_{[1]}=60; Q_{[2]}=45; Q_{[3]}=50,327;$ $Q_{[4]}=28,785; Q_{[5]}=20,574;$ $Q_{[6]}=9,673; Q_{[7]}=0,641$	Lanjut
C	7	2	1	2	2	36738,3	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=2; g_{[4]}=1;$ $g_{[5]}=1; g_{[6]}=3; g_{[7]}=4$	$Q_{[1]}=60; Q_{[2]}=48,565; Q_{[3]}=45;$ $Q_{[4]}=38,528; Q_{[5]}=21,472;$ $Q_{[6]}=1,435; Q_{[7]}=0$	Berhenti
P	6	1	3	1	1	36727,9			
C	7	2	3	1	1	36722,0	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=1; g_{[4]}=2;$ $g_{[5]}=2; g_{[6]}=1; g_{[7]}=2$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=57,646;$ $Q_{[4]}=30,409; Q_{[5]}=14,591;$ $Q_{[6]}=2,354; Q_{[7]}=0$	Berhenti
C	7	1	4	1	1	36727,9	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=1; g_{[4]}=2;$ $g_{[5]}=2; g_{[6]}=2; g_{[7]}=2$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=60;$ $Q_{[4]}=29,639; Q_{[5]}=13,986;$ $Q_{[6]}=1,375; Q_{[7]}=0$	Berhenti
C	7	1	3	2	1	36685,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2;$ $g_{[5]}=3; g_{[6]}=2; g_{[7]}=2$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,641;$ $Q_{[4]}=34,32; Q_{[5]}=13,359;$ $Q_{[6]}=10,68; Q_{[7]}=0$	Berhenti
C	7	1	3	1	2	36727,9	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=1; g_{[4]}=2;$ $g_{[5]}=2; g_{[6]}=2; g_{[7]}=4$	$Q_{[1]}=60; Q_{[2]}=50; Q_{[3]}=60;$ $Q_{[4]}=29,673; Q_{[5]}=14,007;$ $Q_{[6]}=1,32; Q_{[7]}=0$	Berhenti
P	6	1	2	2	1	36685,7			
C	7	2	2	2	1	36685,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2;$ $g_{[5]}=3; g_{[6]}=2; g_{[7]}=1$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,642;$ $Q_{[4]}=34,322; Q_{[5]}=13,358;$ $Q_{[6]}=10,678; Q_{[7]}=0$	Berhenti
C	7	1	3	2	1	36685,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2;$ $g_{[5]}=3; g_{[6]}=2; g_{[7]}=2$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,641;$ $Q_{[4]}=34,32; Q_{[5]}=13,359;$ $Q_{[6]}=10,68; Q_{[7]}=0$	Berhenti
C	7	1	2	3	1	36685,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2;$ $g_{[5]}=3; g_{[6]}=2; g_{[7]}=3$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,642;$ $Q_{[4]}=34,322; Q_{[5]}=13,358;$ $Q_{[6]}=10,678; Q_{[7]}=0$	Berhenti
C	7	1	2	2	2	36685,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2;$ $g_{[5]}=3; g_{[6]}=2; g_{[7]}=4$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,642;$ $Q_{[4]}=34,322; Q_{[5]}=13,358;$ $Q_{[6]}=10,678; Q_{[7]}=0$	Berhenti
P	6	2	1	2	1	36738,3			
C	7	3	1	2	1	36621,7	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=3; g_{[4]}=1;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=1$	$Q_{[1]}=60; Q_{[2]}=45; Q_{[3]}=35,976;$ $Q_{[4]}=42,735; Q_{[5]}=14,024;$ $Q_{[6]}=15,795; Q_{[7]}=1,47$	Lanjut
C	7	2	2	2	1	36722,5	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=3;$ $g_{[5]}=2; g_{[6]}=2; g_{[7]}=1$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=29,63;$ $Q_{[4]}=20,37; Q_{[5]}=30,229;$ $Q_{[6]}=14,771; Q_{[7]}=0$	Berhenti
C	7	2	1	3	1	36615,6	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=1; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=3$	$Q_{[1]}=60; Q_{[2]}=45; Q_{[3]}=50,327;$ $Q_{[4]}=28,785; Q_{[5]}=20,574;$ $Q_{[6]}=9,673; Q_{[7]}=0,641$	Lanjut
C	7	2	1	2	2	36738,3	$g_{[1]}=4; g_{[2]}=3; g_{[3]}=2; g_{[4]}=1;$ $g_{[5]}=1; g_{[6]}=3; g_{[7]}=4$	$Q_{[1]}=60; Q_{[2]}=48,565; Q_{[3]}=45;$ $Q_{[4]}=38,528; Q_{[5]}=21,472;$ $Q_{[6]}=1,435; Q_{[7]}=0$	Berhenti
P	6	1	2	2	1	36685,7			
C	7	2	2	2	1	36685,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2;$ $g_{[5]}=3; g_{[6]}=2; g_{[7]}=1$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,642;$ $Q_{[4]}=34,322; Q_{[5]}=13,358;$ $Q_{[6]}=10,678; Q_{[7]}=0$	Berhenti
C	7	1	3	2	1	36685,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2;$ $g_{[5]}=3; g_{[6]}=2; g_{[7]}=2$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,641;$ $Q_{[4]}=34,32; Q_{[5]}=13,359;$ $Q_{[6]}=10,68; Q_{[7]}=0$	Berhenti
C	7	1	2	3	1	36685,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2;$ $g_{[5]}=3; g_{[6]}=2; g_{[7]}=3$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,642;$ $Q_{[4]}=34,322; Q_{[5]}=13,358;$ $Q_{[6]}=10,678; Q_{[7]}=0$	Berhenti
C	7	1	2	2	2	36685,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2;$ $g_{[5]}=3; g_{[6]}=2; g_{[7]}=4$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,642;$ $Q_{[4]}=34,322; Q_{[5]}=13,358;$ $Q_{[6]}=10,678; Q_{[7]}=0$	Berhenti

Appendix.Enumerasi pencarian solusi(continued)

Status	N	Jml <i>batch</i> per item				AF	Alokasi item pada <i>batch</i>	Ukuran <i>batch</i>	Keterangan
		1	2	3	4				
P	6	1	1	3	1	36731,9			
C	7	2	1	3	1	36615,6	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=1; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=3$	$Q_{[1]}=60; Q_{[2]}=45; Q_{[3]}=50,327;$ $Q_{[4]}=28,785; Q_{[5]}=20,574;$ $Q_{[6]}=9,673; Q_{[7]}=0,641$	Lanjut
C	7	1	2	3	1	36685,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=3; g_{[4]}=2;$ $g_{[5]}=3; g_{[6]}=2; g_{[7]}=3$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=36,642;$ $Q_{[4]}=34,322; Q_{[5]}=13,358;$ $Q_{[6]}=10,678; Q_{[7]}=0$	Berhenti
C	7	1	1	4	1	36727,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=2; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=3; g_{[7]}=3$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=45;$ $Q_{[4]}=23,567; Q_{[5]}=15,748;$ $Q_{[6]}=8,667; Q_{[7]}=2,017$	Lanjut
C	7	1	1	3	2	36731,9	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=2; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=3; g_{[7]}=4$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=45;$ $Q_{[4]}=24,254; Q_{[5]}=16,413;$ $Q_{[6]}=9,333; Q_{[7]}=0$	Berhenti
P	7	3	1	2	1	36621,7			
C	8	4	1	2	1	36621,7	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=3; g_{[4]}=1;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=1; g_{[8]}=1$	$Q_{[1]}=60; Q_{[2]}=45; Q_{[3]}=35,976;$ $Q_{[4]}=42,731; Q_{[5]}=14,024;$ $Q_{[6]}=15,8; Q_{[7]}=1,469; Q_{[8]}=0$	Berhenti
C	8	3	2	2	1	36621,7	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=3; g_{[4]}=1;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=1; g_{[8]}=2$	$Q_{[1]}=60; Q_{[2]}=45; Q_{[3]}=35,976;$ $Q_{[4]}=42,731; Q_{[5]}=14,024;$ $Q_{[6]}=15,8; Q_{[7]}=1,469; Q_{[8]}=0$	Berhenti
C	8	3	1	3	1	36615,6	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=1; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=3; g_{[8]}=1$	$Q_{[1]}=60; Q_{[2]}=45; Q_{[3]}=50,327;$ $Q_{[4]}=28,785; Q_{[5]}=20,574;$ $Q_{[6]}=9,673; Q_{[7]}=0,641; Q_{[8]}=0$	Berhenti
C	8	3	1	2	2	36621,7	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=3; g_{[4]}=1;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=1; g_{[8]}=4$	$Q_{[1]}=60; Q_{[2]}=45; Q_{[3]}=35,976;$ $Q_{[4]}=42,731; Q_{[5]}=14,024;$ $Q_{[6]}=15,8; Q_{[7]}=1,469; Q_{[8]}=0$	Berhenti
P	7	2	1	3	1	36615,6			
C	8	3	1	3	1	36615,6	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=1; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=3; g_{[8]}=1$	$g_{[1]}=60; g_{[2]}=45; g_{[3]}=50,327;$ $g_{[4]}=28,785; g_{[5]}=20,574;$ $g_{[6]}=9,673; g_{[7]}=0,641; g_{[8]}=0$	Berhenti
C	8	2	2	3	1	36615,6	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=1; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=3; g_{[8]}=2$	$Q_{[1]}=60; Q_{[2]}=45; Q_{[3]}=50,327;$ $Q_{[4]}=28,785; Q_{[5]}=20,574;$ $Q_{[6]}=9,673; Q_{[7]}=0,641; Q_{[8]}=0$	Berhenti
C	8	2	1	4	1	36615,6	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=1; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=3; g_{[8]}=3$	$Q_{[1]}=45; Q_{[2]}=45; Q_{[3]}=50,327;$ $Q_{[4]}=28,785; Q_{[5]}=20,574;$ $Q_{[6]}=9,673; Q_{[7]}=0,641; Q_{[8]}=0$	Berhenti
C	8	2	1	3	2	36615,6	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=1; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=3; g_{[8]}=4$	$Q_{[1]}=60; Q_{[2]}=45; Q_{[3]}=50,327;$ $Q_{[4]}=28,785; Q_{[5]}=20,574;$ $Q_{[6]}=9,673; Q_{[7]}=0,641; Q_{[8]}=0$	Berhenti
P	7	1	1	4	1	36727,7			
C	8	2	1	4	1	36615,6	$g_{[1]}=4; g_{[2]}=2; g_{[3]}=1; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=1; g_{[7]}=3; g_{[8]}=3$	$Q_{[1]}=45; Q_{[2]}=45; Q_{[3]}=50,327;$ $Q_{[4]}=28,785; Q_{[5]}=20,574;$ $Q_{[6]}=9,673; Q_{[7]}=0,641; Q_{[8]}=0$	Berhenti
C	8	1	2	4	1	36727,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=2; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=3; g_{[7]}=3; g_{[8]}=2$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=45;$ $Q_{[4]}=23,567; Q_{[5]}=15,748;$ $Q_{[6]}=8,667; Q_{[7]}=2,017; Q_{[8]}=0$	Berhenti
C	8	1	1	5	1	36727,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=2; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=3; g_{[7]}=3; g_{[8]}=3$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=45;$ $Q_{[4]}=23,567; Q_{[5]}=15,748;$ $Q_{[6]}=8,667; Q_{[7]}=2,017; Q_{[8]}=0$	Berhenti
C	8	1	1	4	2	36727,7	$g_{[1]}=4; g_{[2]}=1; g_{[3]}=2; g_{[4]}=3;$ $g_{[5]}=3; g_{[6]}=3; g_{[7]}=3; g_{[8]}=4$	$Q_{[1]}=60; Q_{[2]}=60; Q_{[3]}=45;$ $Q_{[4]}=23,567; Q_{[5]}=15,748;$ $Q_{[6]}=8,667; Q_{[7]}=2,017; Q_{[8]}=0$	Berhenti

Keterangan :

Status : P = *parent*, C = *child*, AF = waktu tinggal aktual