

Penerapan Algoritma *Divide and Conquer* pada *Material Requirements Planning*

Dimas Gilang Saputra
13509038

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
dimasgsaputra@yahoo.com

Abstrak—Makalah ini membahas tentang salah satu aplikasi algoritma komputer *divide and conquer* dalam bidang manajemen inventarisasi yaitu *material requirements planning* (MRP). MRP adalah perencanaan produksi dan sistem pengendalian inventarisasi yang digunakan untuk mengelola proses manufaktur. Masalah MRP yang akan dibahas di makalah ini yaitu menentukan jumlah dan waktu pemesanan barang dengan penentuan ukuran pemesanan secara *lot-for-lot*.

Kata Kunci—*divide and conquer*, inventarisasi, manajemen, *material requirements planning*

I. PENDAHULUAN

Dengan berkembangnya teknologi, sekarang teknologi komputer bisa digunakan di hampir semua bidang dalam kehidupan manusia. Salah satu bidang yang sangat membutuhkan bantuan teknologi komputer adalah bidang bisnis. Bahkan bisa dibilang teknologi komputer adalah suatu pelayanan untuk mencapai keberhasilan bisnis. *Material requirements planning* (MRP) adalah salah satu bidang bisnis yang membutuhkan bantuan komputer.

Untuk menjamin keberjalanan suatu perusahaan produksi, apapun produknya, perusahaan-perusahaan tersebut menghadapi masalah yang sama - bahwa pelanggan menginginkan produk akan tersedia dalam waktu yang lebih pendek dari yang dibutuhkan untuk dibuat oleh pabrik pembuat. Ini berarti bahwa beberapa tingkat perencanaan diperlukan. Maka dari itu terciptalah MRP. Perencanaan ini sebenarnya bisa dibuat tanpa bantuan komputer, tetapi agar perencanaan ini lebih mudah dibuat dan kesalahan dalam perencanaan menjadi kecil maka teknologi komputer dibutuhkan. Selain itu, dengan menggunakan teknologi komputer, waktu untuk pembuatan suatu perencanaan akan semakin singkat.

Untuk membuat perencanaan yang baik maka dibutuhkan algoritma yang baik pula. Dalam kasus MRP ini algoritma *divide and conquer* cocok untuk digunakan karena sifatnya yang sama dengan sifat MRP.

II. LANDASAN TEORI

A. *Material Requirements Planning*

Menurut Joseph Orlicky (salah seorang pencipta MRP) *Material requirements planning* (MRP) merupakan suatu teknik atau prosedur logis untuk menterjemahkan Jadwal Produksi Induk (JPI) dari barang jadi atau *end item* menjadi kebutuhan bersih untuk beberapa komponen yang dibutuhkan untuk mengimplementasikan JPI. MRP ini digunakan untuk menentukan jumlah dari kebutuhan material untuk mendukung Jadwal Produksi Induk dan kapan kebutuhan material tersebut dijadwalkan.

Perencanaan MRP ini mencakup semua kebutuhan akan semua komponen MRP yaitu kebutuhan material, dimana terdapat dua fungsi dengan diterapkannya MRP yaitu pengendalian persediaan dan penjadwalan produksi. Suatu sistem MRP dimaksudkan secara simultan untuk memenuhi tiga tujuan:

1. Memastikan bahan yang tersedia untuk produksi dan produk-produk yang tersedia untuk pengiriman ke pelanggan.
2. Mempertahankan tingkat persediaan serendah mungkin.
3. Rencana kegiatan produksi, jadwal pengiriman dan aktivitas pembelian

MRP menyediakan jawaban untuk beberapa pertanyaan sebagai berikut:

- Barang apa yang dibutuhkan?
- Berapa banyak yang dibutuhkan?
- Kapan barang tersebut dibutuhkan?

Ada empat tahap dalam proses perencanaan kebutuhan material, tahapan tersebut adalah sebagai berikut :

1. *Netting* (Perhitungan kebutuhan bersih)
Netting adalah proses perhitungan kebutuhan bersih yang besarnya merupakan selisih antara kebutuhan kotor dengan keadaan persediaan.
2. *Lotting* (Penentuan ukuran pemesanan)
Lotting adalah menentukan besarnya pesanan setiap individu berdasarkan pada hasil perhitungan *netting*. Pada makalah ini akan dibahas mengenai pengukuran *lot-for-lot* yaitu memproduksi secara

tepat berapa yang diperlukan.

3. *Offsetting* (Penetapan besarnya waktu ancap-ancang)

Offsetting bertujuan untuk menentukan saat yang tepat untuk melaksanakan rencana pemesanan dalam memenuhi kebutuhan bersih yang diinginkan lead time.

4. *Exploding* (Perhitungan selanjutnya untuk level di bawahnya)

Exploding adalah proses perhitungan kebutuhan kotor untuk tingkat level dibawahnya, berdasarkan pada rencana pemesanan.

Dengan MRP ini, kita akan mendapatkan informasi mengenai :

1. Bahan dan komponen apa saja yang akan dipesan serta berapa banyak yang diperlukan.
2. Kapan waktu komponen tersebut akan dipesan.
3. Apakah komponen tersebut pemesanannya dipercepat, diperlambat atau dibatalkan.

Dan yang terakhir adalah keuntungan dari MRP (Heizer, et.al., 1993) yaitu sebagai berikut :

1. Meningkatkan pelayanan dan kepuasan pelanggan
2. Meningkatkan utilitas dari fasilitas dan tenaga kerja
3. Perencanaan persediaan dan penjadwalan menjadi lebih baik
4. Respon terhadap perubahan pasar semakin cepat
5. Mengurangi level persediaan tanpa mengurangi pelayanan pelanggan

MRP dapat diterapkan baik untuk barang-barang yang didapat dari pemasok luar atau yang diproduksi sendiri.

B. Divide and Conquer

Divide and conquer adalah metode pemecahan masalah yang bekerja dengan membagi masalah besar menjadi beberapa sub-masalah yang lebih kecil, kemudian menyelesaikan masing-masing sub-masalah tersebut secara independen, dan akhirnya menggabungkan solusi masing-masing sub-masalah sehingga menjadi solusi dari masalah semula.

Algoritma *divide and conquer* memiliki tiga proses utama yaitu:

- *Divide*, membagi persoalan menjadi beberapa upa-masalah yang memiliki kemiripan dengan persoalan semula namun berukuran lebih kecil (idealnya berukuran hampir sama).
- *Conquer (Solve)*, memecahkan (menyelesaikan) masing-masing upa-masalah secara rekursif.
- *Combine*, menggabungkan solusi masing-masing upa-masalah sehingga membentuk solusi persoalan semula.

Obyek persoalan yang dibagi adalah masukan (input) atau *instances* persoalan yang berukuran n seperti tabel (larik), matriks, eksponen, dll, bergantung persoalannya.

Pada algoritma *divide and conquer*, tiap-tiap sub-masalah mempunyai karakteristik yang sama dengan karakteristik masalah yang akan diselesaikan, sehingga

metode *divide and conquer* lebih natural diungkapkan dalam skema rekursif. Secara umum, algoritma *divide and conquer* memiliki skema seperti Kode 1.

```
procedure DIVIDE_and_CONQUER(input n : integer)
{
  Menyelesaikan masalah dengan algoritma D&C
  Masukan: masukan yang berukuran n
  Keluaran: solusi dari masalah semula
}
```

Deklarasi

r, k : integer

Algoritma

```
if n <= n0 then
{ ukuran masalah sudah cukup kecil }
  SOLVE upa-masalah yang berukuran n ini
else
  Bagi menjadi r upa-masalah,
  masing-masing berukuran n/k
  for masing-masing dari r upa-masalah do
    DIVIDE_and_CONQUER(n/k)
  endfor
  COMBINE solusi dari r upa-masalah
  menjadi solusi masalah semula
endif
```

Kode 1

III. METODE PEMECAHAN MASALAH

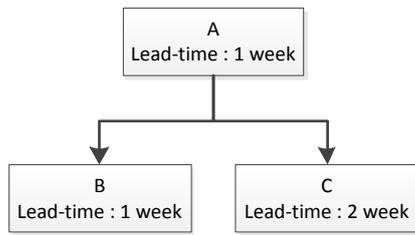
Metode penghitungan dengan cara manual dapat dilakukan dengan cara membuat tabel untuk setiap barang. Hal-hal yang dicatat dalam tabel adalah :

- *gross requirement* (jumlah kebutuhan barang bruto)
- *on-hand* (jumlah barang yang dimiliki)
- *net requirements* (jumlah kebutuhan barang bersih)
- *planned order receipts* (jumlah barang yang diterima)
- *planned order releases* (jumlah barang yang dipesan) tiap satuan waktu.

Jumlah dan waktu setiap barang ditentukan berdasarkan kebergantungannya dengan barang lain dan *lead time* (waktu dari mulai dipesan sampai barang tersebut dapat diterima) dari barang yang bersangkutan. Dan untuk visualisasinya dapat digunakan diagram hierarki.

Misalkan ada konsumen memesan barang A sebanyak 100 dan harus dikirim pada minggu ke-4. Untuk memproduksi satu barang A diperlukan 2 barang B dan 3 barang C. Waktu produksi barang A adalah 1 minggu. *Lead time* barang B adalah 1 minggu. *Lead time* barang C adalah 2 minggu. Di minggu pertama kita sudah memiliki 50 barang B.

Dari soal tersebut diagramnya digambarkan dalam Gambar 1.



Gambar 1

Bentuk tabelnya digambarkan pada Tabel 1.

Week number		1	2	3	4
Quantity					100
A Lt = 1 week	Gross Req				100
	On-Hand	-	-	-	-
	PO receipts				100
	PO releases			100	
B Lt = 1 week	Gross-Req			200	
	On Hand	50	50	50	
	PO receipts			150	
	PO releases		150		
C Lt = 2 week	Gross-Req			300	
	On Hand	-	-	-	
	PO receipts			300	
	PO releases	300			

Tabel 1

Dari tabel akan tergambar perencanaan jumlah dan waktu setiap barang yang diperlukan. Di contoh tersebut kita membutuhkan barang B sebanyak 150 yang harus dipesan pada minggu ke-2 dan barang C sebanyak 300 yang harus dipesan pada minggu ke-1.

Contoh tersebut adalah salah satu kasus dengan jumlah barang yang dibutuhkan sedikit dan kebergantungan pada barang lainnya juga sedikit. Lain halnya jika barang yang dibutuhkan banyak dan/atau barang yang bergantung pada barang lain juga banyak maka akan sangat beresiko karena dapat mengakibatkan kesalahan penghitungan.

Dalam penghitungan MRP ini metode *divide and conquer* dapat dipakai. Penghitungan jumlah dan waktu pemesanan suatu barang dapat ditentukan dengan mengecek setiap barang yang diperlukan untuk memproduksi barang yang dipesan. Setiap barang yang dibutuhkan dicek lagi apakah barang tersebut memerlukan barang lainnya sampai barang terakhir yang tidak memerlukan barang lain lagi. Dari barang terakhir ini dihitung jumlah yang harus dipesan berdasarkan jumlah barang ini untuk memproduksi satu barang yang membutuhkan barang ini lalu dikalikan dengan jumlah barang yang membutuhkan barang ini dikurangi dengan jumlah barang ini yang sudah dimiliki. Waktu untuk memesan barang terakhir ini dihitung berdasarkan lead

time barang ini dan waktu barang yang membutuhkan barang ini untuk ada. Lalu waktu dan jumlah tiap barang disatukan untuk didapatkan suatu bentuk rencana pemesanan setiap barang.

Pseudocode algoritma *divide and conquer* untuk menyelesaikan MRP ditampilkan pada Kode 2.

```

procedure MRP (input B: barang, K: integer, W:
integer, output L: list)
{
  Menghitung jumlah dan waktu pemesanan suatu
barang dalam MRP.
  Masukan : barang (B), kuantitas (K), waktu
(W)
  Keluaran : list of waktu dan kuantitas semua
barang yang dibutuhkan (L)
}

Deklarasi :
  kuantitas, waktu : integer
  list_of_barang : list
  function is_butuh_barang (input B: barang) ->
boolean
  { mengembalikan nilai true jika barang B
membutuhkan barang lain }
  function waktu_proses (input B: barang) ->
waktu
  { mengembalikan waktu produksi atau lead time
barang B }
  function semua_barang (input B: barang) ->
list
  { mengembalikan list semua barang yang
dibutuhkan oleh barang B }
  function qty (input B: barang, R: barang) ->
integer
  { mengembalikan kuantitas barang R untuk
memproduksi satu barang B }

```

```

Algoritma :
  if (not is_butuh_barang(B)) then
    waktu <- W - waktu_proses(B)
    L.add(B,K,waktu)
  else
    list_of_barang <- semua_barang(B)
    for each list_of_barang as R do
      kuantitas <- K * qty(B,R)
      waktu <- W - waktu_proses(B)
      MRP(R,kuantitas,waktu)
    endfor
  COMBINE_list(L)
endif

```

Kode 2

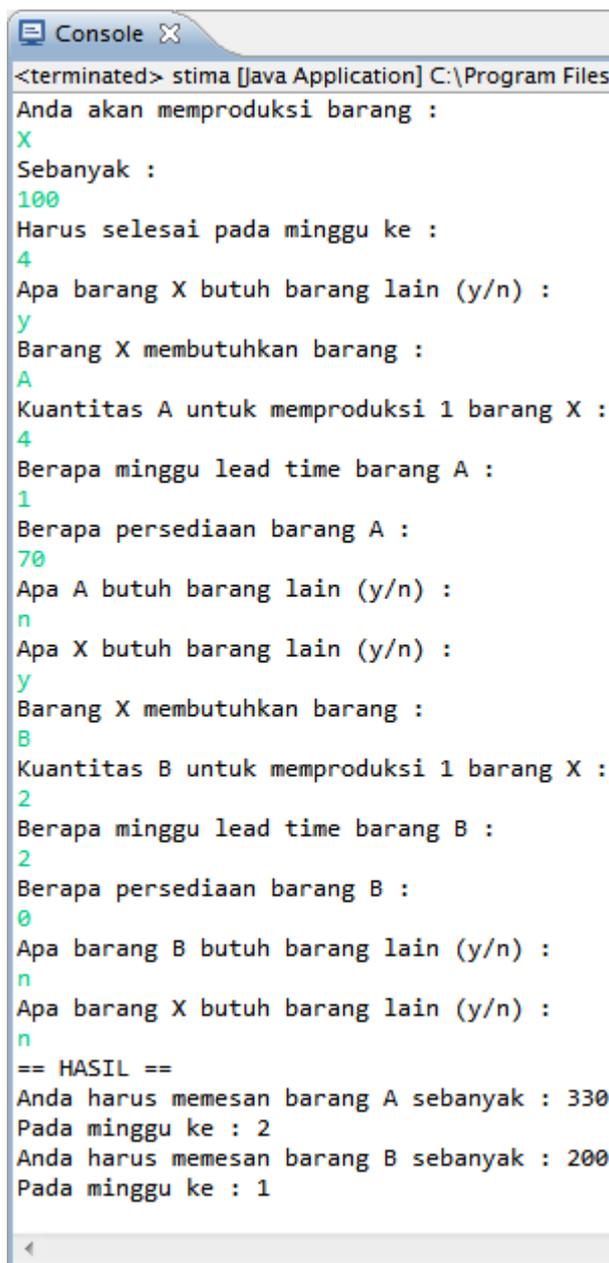
Misalkan ada suatu kasus sebagai berikut:

- Sebuah perusahaan kayu membuat produk X, menerima pesanan sebesar 100 unit.
- Pesanan harus selesai pada permulaan minggu ke-4.
- Produk X terdiri dari 4 unit bagian A dan 2 unit

bagian B.

- Bagian A dibuat sendiri oleh perusahaan yang membutuhkan waktu selama 1 minggu, sedangkan bagian B dipesan dari luar dengan lead time 2 minggu.
- Untuk merakit produk X memerlukan waktu 1 minggu.
- Persediaan bagian A ada 70 unit.
- Tentukan jumlah dan waktu kapan suatu *planned order releases* diperlukan agar pesanan bisa datang sesuai dengan yang dibutuhkan produksi.

Solusi dari contoh keluaran program dalam bahasa *Java* dan dijalankan pada IDE *Eclipse* ada pada *Gambar 2*.



```
<terminated> stima [Java Application] C:\Program Files
Anda akan memproduksi barang :
X
Sebanyak :
100
Harus selesai pada minggu ke :
4
Apa barang X butuh barang lain (y/n) :
Y
Barang X membutuhkan barang :
A
Kuantitas A untuk memproduksi 1 barang X :
4
Berapa minggu lead time barang A :
1
Berapa persediaan barang A :
70
Apa A butuh barang lain (y/n) :
n
Apa X butuh barang lain (y/n) :
y
Barang X membutuhkan barang :
B
Kuantitas B untuk memproduksi 1 barang X :
2
Berapa minggu lead time barang B :
2
Berapa persediaan barang B :
0
Apa barang B butuh barang lain (y/n) :
n
Apa barang X butuh barang lain (y/n) :
n
== HASIL ==
Anda harus memesan barang A sebanyak : 330
Pada minggu ke : 2
Anda harus memesan barang B sebanyak : 200
Pada minggu ke : 1
```

Gambar 2

IV. ANALISIS

Algoritma *divide and conquer* sangat membantu dalam penghitungan jumlah dan waktu barang yang harus dipesan. Dengan penerapan algoritma ini waktu untuk pemrosesan akan jauh lebih cepat.

Bayangkan jika secara manual kita harus membuat tabel dari setiap barang yang terlibat. Semua perhitungan akan memakan waktu yang cukup lama dan resiko kesalahannya besar.

V. KESIMPULAN

Algoritma *divide and conquer* memiliki kegunaan yang sangat banyak. Tidak hanya di bidang komputerisasi tapi juga bisa untuk bidang yang lain. Dalam makalah ini dibahas penerapan algoritma *divide and conquer* di bidang manajemen inventarisasi yaitu *material requirements planning* (MRP).

MRP sangat membutuhkan ketelitian dan sulit diselesaikan jika hanya dengan cara manual. Selain itu MRP akan membutuhkan waktu yang lama jika diselesaikan hanya dengan cara manual dan dapat mengakibatkan kesalahan perhitungan. Kesalahan menghitung satu barang akan mengakibatkan kesalahan keseluruhan perhitungan. Jika terdapat kesalahan pada perencanaan maka keberlangsungan proses bisnis pun dapat terancam.

Algoritma *divide and conquer* cocok digunakan untuk MRP karena algoritma ini merupakan paradigma desain algoritma rekursif yang menyelesaikan persoalan besar dengan cara membagi masalah tersebut menjadi masalah kecil lalu menggabungkan solusi masalah-masalah kecil tersebut menjadi solusi dari permasalahan awal. Sesuai dengan sifat MRP yaitu membagi kebutuhan suatu barang sampai barang-barang yang tidak membutuhkan barang lainnya lagi.

Dengan algoritma *divide and conquer* maka perencanaan pemesanan barang akan lebih mudah dilakukan dan kesalahan akan semakin kecil.

REFERENSI

- [1] Munir, Rinaldi. 2007. *Diktat kuliah IF2251 Strategi Algoritmik*. Bandung:Teknik Informatika ITB.
- [2] Heizer J., Render B. 2005. *Operations Management, 7th ed.* New Jersey: Pearson Education-Prentice Hall.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Desember 2011



Dimas Gilang Saputra
13509038