Penerapan Algoritma Branch and Bound Dalam Pembagian Tugas Berkelompok

Michael Pascalis Simanjuntak - 120140137
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Sumatera, Jalan Terusan Ryacudu
E-mail (gmail): michael.120140137@student.itera.ac.id

***Abstract*—Pembagaian tugas dalam kelompok sering menjadi permasalahan yang terjadi dikarenakan pembagian tugas yang tidak sesuai dengan kemampuan dan kemauan individu yang menyebabkan pengerjaan tugas tidak berjalan lancar. Diluar faktor tersebut, terdapat juga faktor lain, seperti ketersediaan waktu masing-masing anggota yang dapat menghambat pengerjaan tugas. Maka, penulis mengimplementasikan algoritma branch and bound untuk optimisasi pembagian tugas kelompok**

Keywords—tugas kelompok, optimisasi, branch and bound

I. PENDAHULUAN

Dalam pembelajaran di sekolah, siswa maupun mahasiswa tidak dapat terlepas dari tugas. Tugas dapat berupa tugas individu atau tugas kelompok. Tugas individu merupakan tugas yang diberikan dan menjadi tanggung jawab tiap individu, sedangkan tugas kelompok merupakan tugas yang menjadi tanggung jawab sekelompok orang.

Dikarenakan tugas kelompok adalah tugas yang menjadi tanggung jawab sekelompok orang, maka dilakukan pembagian tugas kepada tiap anggota kelompok agar pengerjaan tugas menjadi lebih cepat dan efisien. Akan tetapi, seperti yang sering terjadi, pembagian tugas kelompok dapat juga menjadi hambatan jika pembagian tugas tidak dilakukan dengan sama rata, tidak sesuai minat dan kurangnya rasa tanggung jawab terhadap tugas yang dimiliki.

Permasalahan pertama yang dapat terjadi dalam pembagian tugas kelompok adalah mendapat tugas yang lebih berat/banyak dibanding anggota lainnya. Pembagian tugas yang sama rata, penting untuk menjaga keberhasilan pengerjaan tugas dalam kelompok. Ketika salah satu anggota mendapati tugas yang lebih dari anggota lain, maka pengerjaan tugas dapat berakibat dengan pengerjaan asal-asalan, dan kurangnya rasa tanggung jawab.

Permasalahan berikutnya seperti mendapat tugas yang tidak sesuai dengan kemampuan atau minat. Dengan mendapat tugas yang sesuai dengan minat dan kemampuan masing-masing anggota, penting dalam meningkatkan efisiensi pengerjaan tugas. Ketika tugas sesuai minat maka akan terhindar dari rasa malas mengerjakan tugas, dan ketika sesuai dengan kemampuan, maka pengerjaan akan membutuhkan waktu yang lebih cepat dibanding dengan mendapatkan tugas yang tidak sesuai kemampuan.

Permasalah lain seperti adanya ketersediaan waktu oleh masing-masing anggota kelompok dalam mengerjakan tugas. Seperti yang kita ketahui, banyak siswa maupun mahasiswa yang mempunyai aktivitas di luar kegiatan akademik. Adanya kegiatan ini cukup menyita waktu dari siswa maupun mahasiswa. Sehingga ketersediaan waktu yang tidak optimal dapat menjadi salah satu hambatan dalam menyelesaikan tugas dengan optimal.

Untuk menyelesaikan permasalahan-permasalahan yang terdapat, maka penulis membuat matriks nilai berdasarkan kemampuan, kemauan dan ketersediaan waktu oleh masing-masing anggota terhadap tugas yang diberikan berdasarkan jumlah anggota dengan menggunakan algoritma branch and bound untuk optimasi pembagian tugas tersebut.

II. LANDASAN TEORI

A. Algoritma Branch and Bound

Algoritma Branch and Bound adalah algoritma yang digunakan untuk penyelesaian masalah optimasi kombinatorial, yaitu meminimalkan atau memaksimalkan suatu fungsi objektif dan tidak melanggar batasan yang diberikan oleh persoalan.

Algoritma ini dapat dianggap sebagai kombinasi dari Breadth-first Search (BFS) dan algoritma Least Cost Search. Perbedaan antara BFS murni dan Branch and Bound adalah dalam ekspansi simpul, dalam BFS yang diperluas mengikuti aturan FIFO (First In First) sedangkan dalam Branch and Bound setiap simpul diberi nilai bobot (biaya) pertama, simpul yang diperluas merupakan simpul dengan bobot paling optimal.

Dibandingkan dengan algoritma backtracking, ada persamaan dan perbedaan antara algoritma Branch and Bound. Persamaan yang ada antara lain mencari solusi dengan membentuk state space tree dan dilakukan untuk mencegah perluasan simpul mencapai solusi yang diinginkan. Sedangkan perbedaan antara kedua algoritma terletak pada bentuk

masalah yang diselesaikan oleh dan pembangkitan simpul. Algoritma backtracking biasa digunakan untuk masalah non optimasi sedangkan algoritma branch and bound biasa digunakan untuk masalah optimasi. Pembangkitan simpul pada algoritma backtracking biasanya dilakukan dengan menggunakan metode DFS (depth-first search), sedangkan algoritma branch and bound biasanya menggunakan pendekatan pertama yang terbaik.

B. Fungsi Pembatas

Seperti namanya, algoritma branch and bound melakukan penghentian pada simpul yang melintasi batas, yaitu simpul yang tidak lagi mengarah ke solusi yang diharapkan atau tidak mungkin menghasilkan solusi terbaik. Kriteria terminasi yang diterapkan dalam algoritma ini biasanya:

- Nilai bobot simpul tidak lebih baik dari nilai bobot terbaik sejauh ini
- Simpul tidak menggambarkan solusi yang feasible karena ada batasan yang dilanggar
- Solusi yang feasible pada simpul tersebut hanya terdiri atas satu titik atau tidak ada pilihan lainnya. Solusi terbaik diambil ddengan membandingkan nilai fungsi objektif dengan solusi terbaik saat ini lalu mengambil nilai yang terbaik.

C. Assignment Problem

Assignment problem merupakan permasalahan pada memberikan n butir pekerjaan (assignment) pada n orang. Setiap orang akan diberikan sebuah pekerjaan lalu masih ada ongkos (cost) buat menugaskan setiap orang menggunakan sebuah pekerjaan yg digambarkan pada sebuah matriks nilai. Tujuan berdasarkan assignment persoalan ini merupakan menugaskan n orang menggunakan n butir pekerjaan tadi sebagai akibatnya total ongkos yg wajib dikeluarkan seminimal mungkin.

Penyelesaian berdasarkan permasalahan ini memanfaatkan prosedur pemecahan branch and bound. Nilai batasan (bound) berdasarkan persoalan ini didapat dari menjumlahkan nilai minimum berdasarkan setiap baris matriks dikarenakan tidak mungkin terdapat solusi yg total ongkosnya lebih kecil berdasarkan penjumlahan nilai minimum tadi. Kemudian buat bobot atau ongkos setiap simpul didapat menggunakan menjumlahkan nilai ongkos menggunakan status terkini & nilai minimum dalam setiap baris yg belum ditugaskan dan memperhatikan bahwa tidak terdapat pekerjaan yg ditugaskan ke dua orang atau lebih.

Langkah-langkah untuk menyelesaikan masalah ini adalah sebagai berikut. Misalkan ada 4 pekerjaan yang perlu ditugaskan ke 4 orang dan memiliki matriks biaya seperti di bawah ini.

$$C = \begin{bmatrix} Job\,1 & Job\,2 & Job\,3 & Job\,4 & \\ 9 & 2 & 7 & 8 & \text{Orang } a \\ 6 & 4 & 3 & 7 & \text{Orang } b \\ 5 & 8 & 1 & 8 & \text{Orang } c \\ 7 & 6 & 9 & 4 & \text{Orang } d \end{bmatrix}$$

Gambar 1. Matriks
Sumber:
PowerPoint Presentation (itb.ac.id) diakses pada 29 Maret 2022

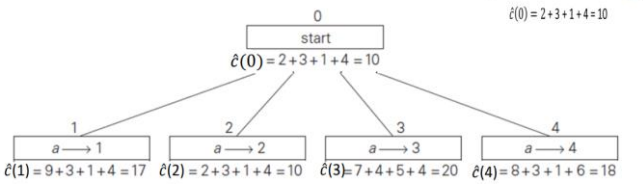
Ongkos untuk simpul akar merupakan penjumlahan dari setiap baris matriks, c(0) = 2+3+1+4 = 10

$$C = \begin{bmatrix} Job\,1 & Job\,2 & Job\,3 & Job\,4 & \\ 9 & \textcircled{2} & 7 & 8 & \text{Orang } a \\ 6 & 4 & \textcircled{3} & 7 & \text{Orang } b \\ 5 & 8 & \textcircled{1} & 8 & \text{Orang } c \\ 7 & 6 & 9 & \textcircled{4} & \text{Orang } d \end{bmatrix}$$

$$\hat{c}(0) = 2+3+1+4 = 10$$

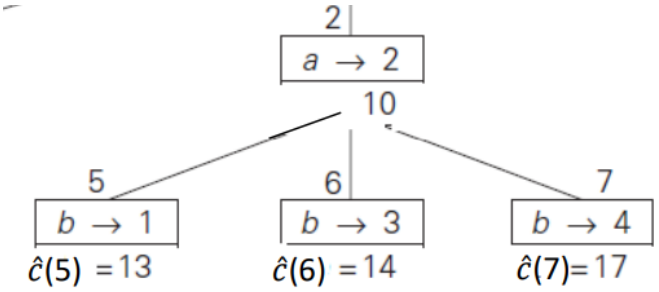
Gambar 2. Matriks
Sumber:
PowerPoint Presentation (itb.ac.id) diakses pada 29 Maret 2022

Kemudian dari simpul akar akan menghasilkan simpul anak, yang memberikan pekerjaan kepada orang a dan kemudian menghitung biaya setiap simpul anak.



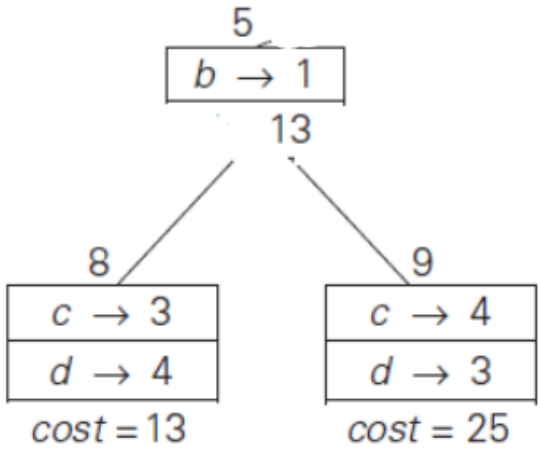
Gambar 3 Pembangkitan Anak dari Simpul Akar
Sumber:
PowerPoint Presentation (itb.ac.id) diakses pada 29 Maret 2022

Ongkos yang menjadi nilai minimum adalah simpul 2 sehingga simpul 2 menjadi silmpul yang diekspansi berikutnya.



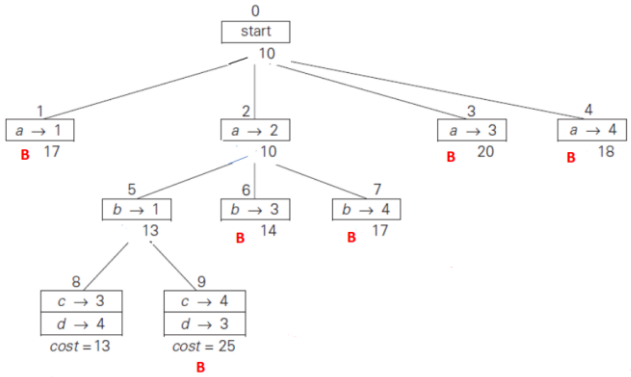
Gambar 4. Pembangkitan Anak dari Simpul 2
Sumber:
PowerPoint Presentation (itb.ac.id) diakses pada 29 Maret 2022

Simpul hidup saat ini adalah 1, 3, 4, 5, 6, dan 7. Simpul dengan ongkos paling kecil adalah simpul 5 sehingga simpul 5 menjadi simpul yang diekspansi berikutnya.



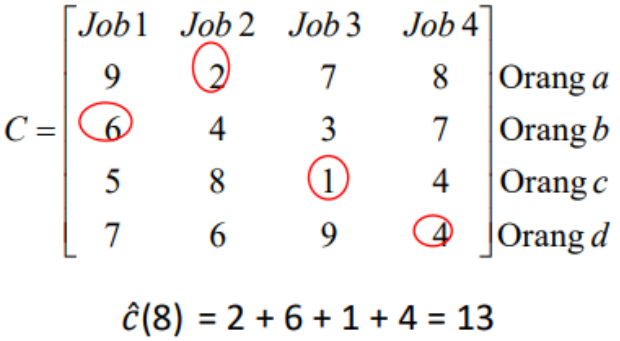
Gambar 5. Pembangkitan Anak dari Simpul 5
Sumber:
PowerPoint Presentation (itb.ac.id) diakses pada 29 Maret 2022

Simpul 8 merupakan simpul solusi dan merupakan solusi optimal. Semua simpul hidup yang memiliki ongkos lebih besar dari 13 dimatikan karena tidak mungkin menghasilkan solusi yang lebih optimal dari 13.



Gambar 6. Pohon Ruang Status Keseluruhan
Sumber:
PowerPoint Presentation (itb.ac.id) diakses pada 29 Maret 2022

Solusi optimal yang diperoleh : (a -> 2, b -> 1, c -> 3, d -> 4) dengan total ongkos 13.



$$\hat{c}(8) = 2 + 6 + 1 + 4 = 13$$

Gambar 7. Solusi Akhir dari Contoh Assignment Problem
Sumber:
PowerPoint Presentation (itb.ac.id) diakses pada 29 Maret 2022

III. Optimasi Pembagian Tugas Kelompok

A. *Ongkos Pembagian Tugas Kelompok*

Biaya yang digunakan untuk mengoptimalkan pembagian kerja untuk kelompok ini adalah nilai yang diperoleh dari kuesioner atau survei singkat yang melibatkan tiga komponen utama dalam pembagian kerja, yaitu kemampuan, kemauan, dan ketersediaan waktu. Oleh karena itu, nilai matriks biaya akan digunakan dengan menggunakan rumus :

Ongkos = (Nilai kemampuan + Nilai kemauan + Nilai ketersediaan waktu)/3

B. *Langkah Kerja Optimisasi Pembagian Tugas Kelompok*

- Membagi tugas kelompok menjadi sub-tugas sebanyak jumlah anggota kelompok
- Membuat kuesioner singkat mengenai kemampuan, kemauan, dan ketersediaan waktu anggota dalam mengerjakan sub-tugas dengan indikator skala 1-N, dengan N adalah jumlah anggota kelompok. Nilai yang lebih kecil menunjukkan anggota lebih mahir atau lebih mau atau lebih memiliki waktu dalam mengerjakan sub-tugas
- Mengubah nilai-nilai yang diperoleh dari kuesioner menjadi nila bobot menggunakan rumus
- Membuat matriks nilai berdasarkan nilai bobot yang dihasilkan
- Menerapkan algoritma branch and bound untuk menemukan pembagian tugas yang optimal menggunakan matriks nilai yang sudah dibuat

C. *Notasi Algoritmit Optimisasi Pembagian Tugas Kelompok*

a) Notasi algoritmik mengubah nilai kuesioner menjadi nilai bobot

```
function QuestMatrixToCostMatrix (input :
```

```
questMatrix[N][3N], output :
costMatrix[N][N]) {N :
jumlah anggota kelompok}
 {inisialisasi costMatrix}
 for(i←0; i<N; i++)
 for(j←0; j<N; j++)
 costMatrix[i][j] ← 0
 endfor
 endfor
 {pembentukan costMatrix}
 for(i←0 ; i<N; i++)
 for(j←0; j<N; j++)
 k ← j*3
 costMatrix[i][j] ← questMatrix[i][k] +
questMatrix[i][k+1] + questMatrix[i][k+2]
 endfor
 endfor
 return costMatrix
```

b) Notasi algoritmik optimisasi pembagian tugas

Notasi algoritmik algoritma optimisasi pembagian tugas menggunakan algoritma branch and bound dengan referensi dari https://www.geeksforgeeks.org/job-assignment-problem-using-branch-and-bound/

```
procedure OptimizePembagianTugas (input:
questMatrix[N][3N]) {Prosedur mencetak
pembagian tugas yang optimal menggunakan
branch and bound}
 KAMUS
 min : Node
 pq : Priority Queue of Node
 child : Node
 type Node : <parent : pointer to Node,
 pathCost : integer,
 cost : integer,
 studentID : integer,
 jobID : integer,
 assigned : array[1..N] of boolean>
 FUNGSI/PROSEDUR
 function initPrioQueue(output : Priority
Queue of
Node pq) {inisialisasi priority queue bertipe
Node
dengan top queue merupakan Node dengan cost
terkecil}
 function QuestMatrixToCostMatrix (input :
questMatrix[N][3N], output :
costMatrix[N][N])
{Mengubah matriks nilai kuesioner menjadi
matriks
nilai bobot}
 function calculateCost(costMatrix :
array[1..N] of
array[1..N] of integer; sID, jID : integer;
assigned :
array[1..N] of boolean) → integer {menghitung
dan
memperbaharui bobot simpul}
 function printAssignments(min : Node)
{mencetak
pembagian tugas melalui optimisasi}
 function newNode(i,j : integer ; assigned :
array[1..N] of boolean, parent : Node) → Node
{membuat Node baru ketika pekerjaan j
ditugaskan
kepada siswa i}
 ALGORITMA
 costMatrix ←
QuestMatrixToCostMatrix(questMatrix)
 pq ← initPrioQueue()
 while (pq not empty) do
 min ← Top(pq)
 Pop(pq)
 i ← min.studentID + 1
 if (i = N) then
 printAssignments(min)
 return min.cost
 for (j←0; j<N; j++)
 if(not min.assigned[j]) then
 child ← newNode(i, j, min.assigned, min)
 child.pathCost ← min.pathCost +
costMatrix[i][j]
 child.cost ← child.pathCost +
calculateCost(costMatrix, i, j,
child.assigned)
 Push(pq, child)
 REALISASI FUNGSI/PROSEDUR
 function newNode(x,y : integer ; assigned :

array[1..N] of boolean, parent : Node) → Node
 for (int j = 0; j < N; j++)
 assign[j] ← assigned[j];
 assign[y] = true;
 node ← <parent, 0, 0, x, y, assign>

 return node;
```

```
 function calculateCost(costMatrix :
array[1..N] of
array[1..N] of integer; sID, jID : integer;
assigned :
array[1..N] of boolean) → integer
 cost ← 0;

 available ← {true}

 for (int i ← x + 1; i < N; i++)
 min ← 9999, minIndex ← -1
 for (int j ← 0; j < N; j++)
 if (not assigned[j] and costMatrix[i][j] <
min) then
 {
 minIndex ← j;
 min ← costMatrix[i][j]
 }
 }
 cost += min;
 available[minIndex] = false
 }
 return cost;
 function printAssignments(min : Node)
 if(min.parent = NULL) then
 return
 printAssignments(min.parent)
 output("Siswa " + char(min.studentID + 'A')
+ "
mendapat tugas " + min.jobID)
```

D. Contoh Pengujian Algoritma

Misalkan terdapat 4 sub-tugas yang akan diberikan kepada 4 orang. Hasil dari kuesioner mengenai kemampuan, kemauan, dan ketersediaan waktu anggota dalam mengerjakan setiap modul tugas dinyatakan dalam tabel berikut.

Table 1 Tabel Matriks Nilai Kuesioner Kasus Uji

	Tgs1			Tgs2			Tgs3			Tgs4		
	Kemahiran	Kemauan	Waktu	Kemahiran	Kemauan	Waktu	Kemahiran	Kemauan	Waktu	Kemahiran	Kemauan	Waktu
A	1	3	4	4	1	4	3	4	4	2	2	4
B	3	1	2	4	3	2	1	4	2	2	2	2
C	2	2	2	3	1	2	4	4	2	1	3	2
D	2	1	3	1	2	3	3	3	3	4	4	3

Sumber : Dokumen Pribadi

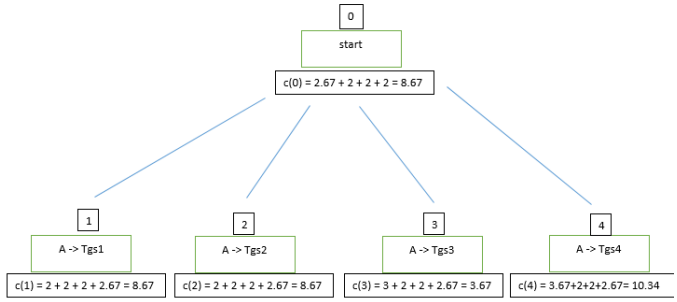
Hasil konversi matriks kuesioner menjadi matriks biaya :

Table 2. Tabel Matriks Biaya Kasus Uji

	Tgs1	Tgs2	Tgs3	Tgs4
A	2.67	3	3.67	2.67
B	2	3	2.33	2
C	2	2	3.33	2
D	2	2	3	3.67

Sumber : Dokumen Pribadi

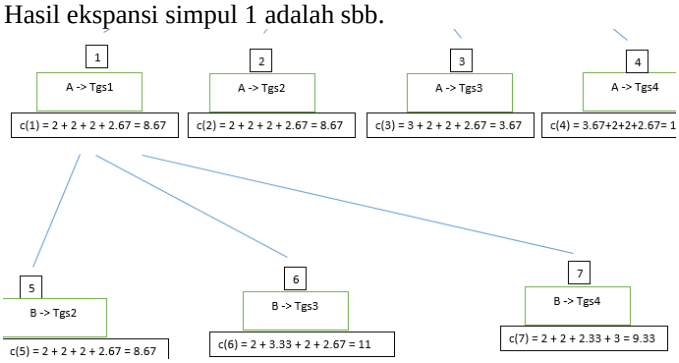
Langkah pertama adalah mengekspansi simpul akar sehingga terbentuk simpul-simpul anak berserta ongkosnya sbb:



Gambar 8. Ekspansi Simpul Akar Kasus Uji

Sumber: Dokumen Pribadi

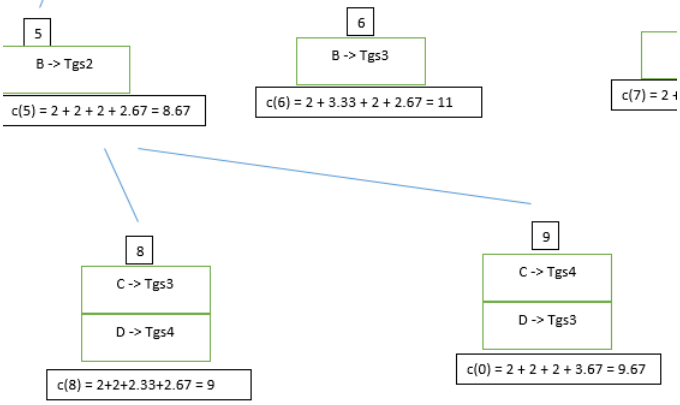
Simpul yang diekspansi berikutnya adalah simpul 1 karena memiliki ongkos yang paling minimun selain simpul akar. Hasil ekspansi simpul 1 adalah sbb.



Gambar 9. Ekspansi Simpul 1Kasus Uji

Sumber: Dokumen Pribadi

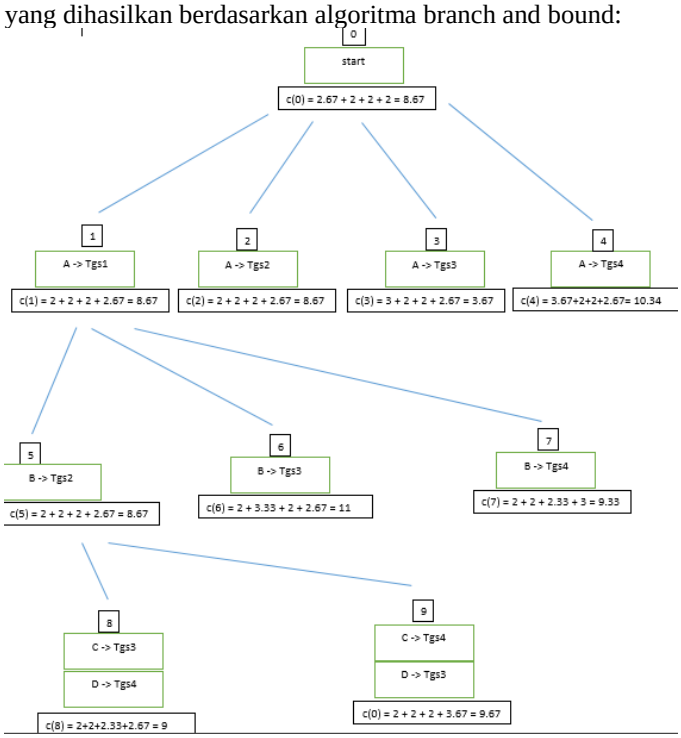
Dapat dilihat simpul 5 memiliki ongkos paling minimum sehingga menghasilkan simpul-simpul anak sbb:



Gambar 10. Ekspansi Simpul 5 Kasus Uji

Sumber : Dokumen Pribadi

Pencarian diberhentikan karena sudah membagikan setiap tugas kepada masing-masing anggota. Pohon ruang status yang dihasilkan berdasarkan algoritma branch and bound:



Gambar 11. Pohon Ruang Status Kasus Uji
Sumber : Dokumen Pribadi

Berdasarkan pohon ruang status yang dihasilkan, didapat solusi optimal : (A -> Tgs1, B -> Tgs2, C -> Tgs3, D -> Tgs4) dengan total ongkos 9

IV. KESIMPULAN

Pembagian tugas dalam berkelompok dapat menjadi permasalahan dalam mengerjakan tugas. Optimasi pembagian tugas dapat dilakukan dengan branch and bound seperti algoritma assignment problem dengan merubah nilai biaya menjadi nilai yang dihasilkan dari tiga komponen yaitu kemampuan, kemauan, dan ketersediaan waktu.

UCAPAN TERIMA KASIH

Rasa syukur saya panjatkan kepada Tuhan YME sehingga penulis dapat menyelesaikan makalah ini dengan baik. Penulis menyadari terdapat kekurangan dan kesalahan kata dalam makalah ini, penulis berharap makalah ini dapat digunakan sebaiknya-baiknya dan dikembangkan sehingga lebih menghasilkan bagi masyarakat luas.

REFERENCES

[1] A. Goel. 2018. "Job Assignment Problem using Branch And Bound". GeeksforGeeks. Diakses pada 9 Mei 2021 dari https://www.geeksforgeeks.org/job-assignment-problem-using-branch and-bound/

[2] Anonim. Diakses pada 8 Mei 2021 dari http://repository.unissula.ac.id/7203/4/BAB%20I_1.pdf

[3] Anonim. Diakses pada 8 Mei 2021 dari http://eprints.ums.ac.id/55978/8/BAB%20I.pdf

[4] R. Munir. 2021. "Algoritma Branch & Bound (Bagian 4)". Sekolah Teknik Elektro dan Informatika ITB. Diakses pada 8 Mei 2021 dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf

[5] R. Munir, N. U. Maulidevi, M. L. Khodra. 2021. "Algoritma Branch & Bound (Bagian 1)". Sekolah Teknik Elektro dan Informatika ITB. Diakses pada 8 Mei 2021 dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian1.pdf

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, 26 April 2021

Michael Pascalis Simanjuntak 120140137