

Penentuan Rotasi Tanaman yang Paling Menguntungkan dengan *Branch and bound*

Jonathan Richard Sugandhi / 13519128
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519128@std.stei.itb.ac.id

Abstrak—Dalam pertanian maupun perkebunan, terkadang kita akan kebingungan akan tanaman atau pola tanam yang akan kita tanam. Terdapat beberapa hal yang harus diperhatikan dalam pola tanam ini, seperti unsur hara, penyakit dan hama, atau bahkan keuntungan. Dengan memperhatikan semua hal ini, akan digunakan algoritma *branch and bound* untuk menentukan pola tanam yang paling menguntungkan secara finansial.

Keywords—Pertanian, Rotasi Tanaman, Dynamic Programming,

I. PENDAHULUAN

Ilmuwan sepakat bahwa agrikultur pertama kali dilakukan oleh manusia pada zaman Mesopotamia jauh sebelum masehi. Agrikultur adalah sebuah terobosan penting dalam sejarah manusia karena memperkenalkan cara terbaik dalam memenuhi kebutuhan pangan secara konstan mengingat manusia zaman praaksara hanya mengenal perburuan pada masa itu. Agrikultur secara singkat adalah sebuah upaya pemanfaatan bumi untuk menghasilkan pangan. Kegiatan agrikultur dapat berupa pertanian, perkebunan, ataupun peternakan.

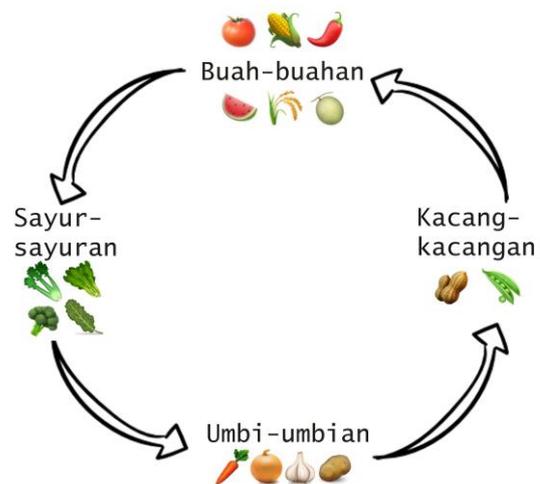


Gambar 1. Pertanian Kentang

(sumber: <https://paktanidigital.com/artikel/5-langkah-sukses-budidaya-kentang/#.YJoZz4fitPY>)

Dalam pertanian dan perkebunan, ada sangat banyak hal yang perlu diperhatikan. Mulai dari pemilihan jenis tanaman

yang ingin ditanam, lalu pemilihan pupuk, sistem pengairan, pengendalian hama, jenis tanah dan iklim, dan masih banyak lainnya. Untuk mengatasi beberapa masalah ini, dalam dunia pertanian dan perkebunan dikenal sebuah istilah rotasi tanaman.



Skema rotasi tanaman | 8villages

Gambar 2. Rotasi Tanaman

(sumber:

<https://8villages.com/full/petani/artikel/id/59db479a536469d27e7b881b>)

[3] Rotasi tanaman adalah salah satu teknik budidaya tanaman dimana pada sebuah lahan yang sama, ditanami lebih dari 1 jenis tanaman dalam waktu yang berbeda. Rotasi tanaman yang baik akan dapat memberikan banyak keuntungan. Yang pertama adalah dapat mengurangi intensitas serangan hama atau penyakit karena saat dilakukan rotasi tanaman dengan famili yang berbeda, siklus hama dan penyakit yang menyerang tanaman sebelumnya akan terputus. Keuntungan lainnya ialah dapat meningkatkan kesuburan tanaman. Setiap tanaman memiliki karakteristik yang unik dalam memengaruhi tingkat kesuburan tanaman. Terdapat beberapa jenis tanaman yang bersifat rakus terhadap unsur hara seperti umbi-umbian dan juga terdapat tanaman yang dapat menyediakan unsur hara yaitu bangsa leguminosae.

Lalu apakah dapat dibuat sebuah sistem rotasi tanaman yang paling menguntungkan? Salah satu cara menemukan rotasi tanaman terbaik ini adalah dengan menggunakan algoritma *branch and bound*.

[1] *Branch and bound* adalah sebuah algoritma yang melibatkan pembangkitan pohon ruang status yang biasa digunakan untuk persoalan optimasi. Karena pencarian sebuah rotasi yang paling menguntungkan dapat dikatakan sebuah persoalan optimasi, maka *branch and bound* cocok digunakan dalam persoalan ini.

II. DASAR TEORI

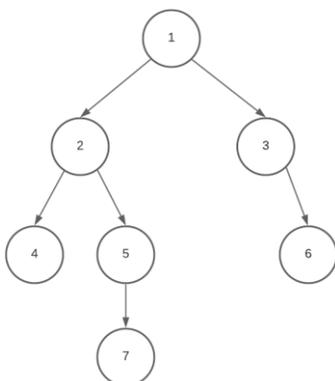
A. Algoritma *Breadth first search*

[4] Algoritma *Breadth First Search* atau BFS merupakan sebuah algoritma traversal graf dimana penelusuran terjadi secara melebar. Algoritma ini termasuk dalam algoritma *blind search* yang dalam penerapannya membutuhkan sebuah struktur data *queue*. *Queue* secara singkat adalah sebuah struktur data berjenis *first in first out* (FIFO) yang sifatnya sangat mirip dengan antrian pada dunia nyata.

Secara umum, algoritma *breadth first search* yang berawal dari simpul *v* ialah:

1. Masuk ke simpul *v*
2. Bangkitkan semua simpul yang bertetangga dengan simpul *v*. Tandai simpul *v*
3. Telusuri simpul yang belum dibangkitkan/ditandai dan bertetangga dengan simpul yang sudah dibangkitkan tadi. Tandai setiap simpul yang sudah ditelusuri
4. Algoritma berlanjut sampai semua simpul yang dapat ditelusuri sudah habis

Berikut adalah contoh penerapan algoritma *breadth first search* pada sebuah graf sederhana, dengan nomor pada simpul adalah urutan simpul yang ditelusuri:



Gambar 3. *Breadth first search*

(sumber: Dibuat sendiri dengan menggunakan lucidchart)

B. Algoritma *Branch and bound*

Algoritma *branch and bound* adalah sebuah algoritma yang biasa digunakan untuk persoalan optimasi. [1] *Branch and bound* melibatkan pembangkitan pohon ruang status dengan bantuan sebuah fungsi pembatas (*constraints*) untuk memangkas simpul. *Branch and bound* dikatakan sebagai kombinasi dari *breadth first search* dengan *least cost search* sehingga pada algoritma *branch and bound*, pembangkitan simpul tidak hanya sekedar FIFO.

Berikut adalah algoritma *branch and bound* dengan simpul awal *v* secara umum:

1. Definisikan fungsi pembatas dan nilai taksiran untuk simpul
2. Masuk ke simpul *v* dan *expand* simpul *v*
3. Cari simpul dengan nilai taksiran paling minimal/maksimal (bergantung pada tiap kasus). *Expand* simpul tersebut.
4. Jika ditemukan solusi, matikan semua simpul yang nilai taksirannya lebih buruk daripada solusi.
5. Lanjutkan hingga tidak ada simpul hidup yang tersisa.

C. Rotasi Tanaman

Dalam pertanian dan perkebunan monokultur, petani seringkali mengalami masalah penyakit/hama yang berkelanjutan, hasil tanaman yang kurang baik karena tanah yang hilang kesuburan, dan terkadang membuat gizi petani kurang memadai. Rotasi tanaman adalah sebuah teknik dalam pertanian dan perkebunan untuk mengatasi masalah-masalah tersebut.

Rotasi tanaman rupanya sudah ditemukan sejak lama secara tidak sengaja dimana petani menanam sereal dengan legumin secara bergantian. Rotasi tanaman menyelesaikan sebagian besar masalah terutama dalam hal pemutusan penyakit tanaman dan kesuburan tanah. Berikut adalah manfaat dan penjelasan singkat dari teknik rotasi tanaman:

1. Memutus penyakit tanaman
Hama dan penyakit pada tiap tanaman yang berbeda terutama secara famili berbeda-beda. Karena sifat ini, maka rotasi tanaman dengan melibatkan tanaman yang berbeda famili akan menyebabkan rantai penyakit terputus sehingga penanaman kembali nantinya akan bebas dari penyakit.
2. Menjaga kesuburan tanah
Beberapa tanaman bersifat mengambil kesuburan tanah, sehingga terkadang dibutuhkan penggiliran penanaman untuk menjaga kesuburan tanah. Penggiliran biasanya melibatkan legum karena legum dapat memfiksasi nitrogen bebas di udara sehingga legum dapat menambahkan unsur hara di tanah.

3. Menstabilkan kebutuhan pasar
 Karena penanaman tanaman melibatkan banyak jenis tanaman, jelas pasar akan lebih tercukupkan kebutuhannya.

[2] Salah satu aturan yang umum dalam rotasi tanaman ialah aturan Kacang – Daun – Buah – Umbi. Berikut adalah penjelasan akan urutan tersebut:

1. Kacang
 Kacang adalah salah satu anggota leguminosae yang memiliki sifat mengembalikan kesuburan tanah. Penanaman kacang diawal didasari untuk mengembalikan kesuburan tanah agar siap ditanami tanaman yang membutuhkan unsur hara. Jenis tanaman yang merupakan kacang adalah kacang tanah.
2. Daun
 Yang dimaksud daun adalah tanaman yang dipanen daunnya. Tanaman ini membutuhkan kondisi tanah yang sudah kaya dalam unsur hara agar dapat tumbuh subur sehingga penanaman kacang sebelum daun sangatlah baik untuk dilakukan. Contoh dari tanaman yang dipanen daunnya adalah bayam dan kangkung.
3. Buah
 Buah seperti namanya adalah tanaman yang dipanen buahnya. Tanaman ini memang membutuhkan kondisi tanah yang kaya akan unsur hara, namun saat memasuki fase generative/fase berbuah tanaman ini lebih membutuhkan fosfat sehingga nitrogen menjadi tidak terlalu mempengaruhi hasil. Adapun tanaman buah ini adalah melon dan mentimun.
4. Umbi
 Umbi ialah tanaman yang dipanen umbinya. Umbi adalah sebuah organ khusus dari tanaman untuk menyimpan cadangan makanan. Umbi ini dapat berupa akar, batang, maupun modifikasinya. Tanaman ber-umbi memiliki sifat yang cukup unik. Tanaman ber-umbi bersifat rakus akan unsur hara sehingga benar-benar mengurangi tingkat kesuburan tanah. Karena sifat inilah tanaman ber-umbi sebaiknya ditanam di paling akhir. Adapun tanaman ber-umbi ialah kentang dan ubi.

III. IMPLEMENTASI

Untuk mencari pola tanaman yang paling menguntungkan, pertama-tama kita akan definisikan fungsi pembatas berdasarkan asumsi bahwa pola tanaman ini dilakukan untuk setengah tahun saja. Karena itu fungsi pembatas dari algoritma adalah jumlah hari tidak melebihi 182 hari. Untuk menggunakan algoritma, harus didefinisikan fungsi untuk mencari nilai taksiran *cost* untuk tiap simpulnya.

Didefinisikan *cost* sebagai berikut:

$$\hat{c}(i) = (182 - \Sigma L) * X + CP$$

Dengan:

- $\hat{c}(i)$ adalah *cost* dari simpul i
- L adalah lama waktu tanam dari tanaman yang pernah diambil pada simpul sebelumnya.
- X adalah keuntungan terbesar yang mungkin dalam 1 hari
- CP adalah *current profit* atau keuntungan yang sudah didapat pada simpul ini dihitung dari harga jual semua tanaman yang terambil.

Catatan:

- Karena terdapat pengecualian dimana umbi bersifat rakus unsur hara, maka setelah penanaman umbi harus diikuti dengan penanaman kacang-kacangan.
- Untuk mencegah terjadinya penerusan hama, maka tanaman yang sama tidak boleh ditanam berturut-turut. Sebagai contoh kacang tanah tidak boleh ditanam apabila tanaman sebelumnya yang ditanam adalah kacang tanah juga.
- Dalam algoritma akan diambil *cost* tertinggi yang menyatakan keuntungan, jadi X harus nilai terbesar.
- Karena dalam algoritma akan dicari nilai yang maksimum, maka simpul yang di-*expand* adalah simpul yang memiliki *cost* tertinggi.
- Nilai X akan menjadi 0 jika simpul tidak dapat di-*expand*
- ΣL adalah 0 pada simpul akar

Berikut adalah algoritma yang akan diterapkan:

1. Pada simpul akar, *expand* simpul-simpul yang mungkin.
2. Dengan memperhatikan batasan yang diungkapkan dalam catatan, *expand* simpul yang memiliki *cost* tertinggi.
3. Jika ditemukan simpul yang tidak lagi dapat di-*expand*, maka ditemukan simpul solusi sehingga matikan semua simpul yang *cost*-nya lebih kecil daripada simpul solusi.
4. Lanjutkan pencarian hingga tidak lagi ada simpul yang dapat di-*expand*.

IV. PENGUJIAN

Akan dilakukan sebuah pengujian algoritma dengan menggunakan data yang telah disederhanakan sebagai berikut:

Tabel 1. Estimasi data tanaman (data bersifat *dummy*)

Tanaman	Waktu tanam	Hasil panen/ha	Harga/kg
Kacang Tanah	100 hari	4.5 ton	25.000
Kentang	80 hari	60 ton	17.000
Kangkung Cabut	30 hari	15 ton	22.500
Melon	90 hari	30 ton	25.000

Terlebih dahulu akan dicari seluruh data yang dibutuhkan untuk melaksanakan algoritma dengan beberapa asumsi untuk mempermudah perhitungan yaitu sebagai berikut:

- Lahan memiliki luas 1 hektar
- Lahan hanya akan ditanami 4 tanaman yang didefinisikan sebelumnya
- Lahan dapat ditanami tanaman apapun pada awal
- Jumlah benih tidak terbatas
- Fungsi pembatas adalah ΣL tidak melebihi setengah tahun yang dianggap sebagai 182 hari
- Jika terdapat nilai $\Sigma L = 182 - \min(\text{Waktu tanam}) = 152$ maka dapat dipastikan simpul tak dapat di-expand. Jika demikian maka X otomatis bernilai 0.
- Harga yang didapat sudah harga bersih dan dijamin terjual
- 3 angka '0' dibelakang tidak ditulis agar hasil terlihat lebih rapi.

Maka dapat disimpulkan:

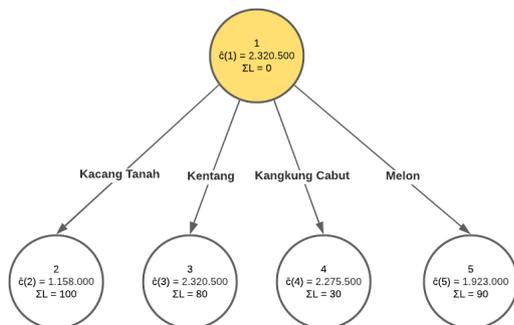
1. Profit = Hasil Panen/ha * 1000 * Harga/Kg
2. $X = \max(\text{Profit/waktu tanam})$

Berikut adalah tabel data yang telah diolah:

Tanaman	Waktu tanam	Profit (000)
Kacang Tanah	100 hari	112.500
Kentang	80 hari	1.020.000
Kangkung Cabut	30 hari	337.500
Melon	90 hari	750.000

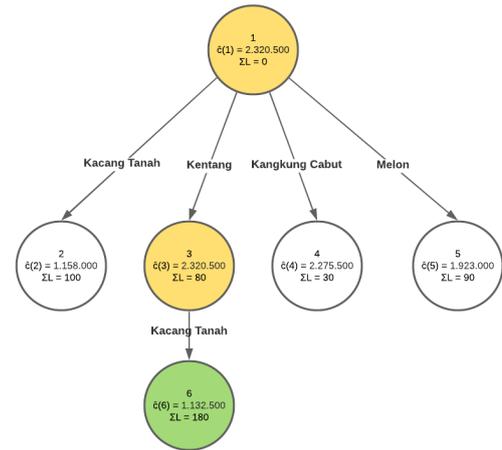
Jadi didapat juga $X = 12.750$

Karena seluruh data sudah didapatkan, algoritma dapat dimulai. Akan dibuat tahapan algoritma dalam bentuk pohon ruang status dimana warna putih berarti simpul hidup, hijau berarti simpul solusi, kuning berarti simpul sudah di-expand, merah berarti simpul mati. Berikut adalah tahap pertama dalam algoritma:



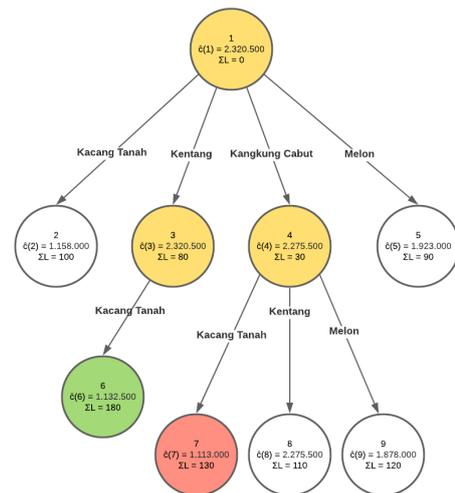
Gambar 4. Penerapan Algoritma Tahap 1
(sumber: Dibuat sendiri dengan menggunakan lucidchart)

Karena sudah didefinisikan dalam asumsi bahwa tanah dapat ditanami oleh semua tanaman, maka dari simpul akar dapat dibuat 4 simpul berikutnya. Karena simpul 3 memiliki taksiran *cost* yang tertinggi, maka simpul 3 akan di-expand.



Gambar 5. Penerapan Algoritma Tahap 2
(sumber: Dibuat sendiri dengan menggunakan lucidchart)

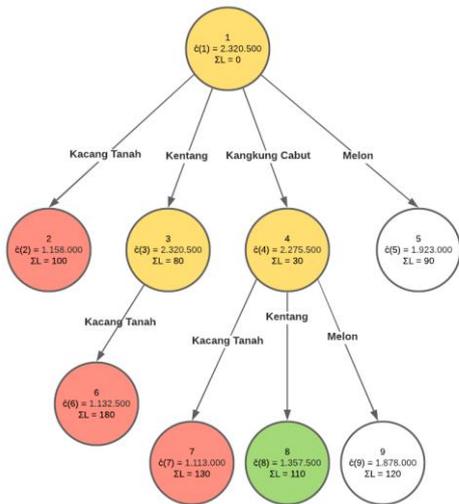
Dari salah satu batasan, setelah penanaman umbi (kentang) haruslah dilakukan penanaman legum (kacang tanah) karena hanya legum yang dapat hidup pada tanah kurang subur dan juga untuk mengembalikan kesuburan tanah. Dilihat dari *cost* nya, maka akan di-expand simpul 4.



Gambar 6. Penerapan Algoritma Tahap 3
(sumber: Dibuat sendiri dengan menggunakan lucidchart)

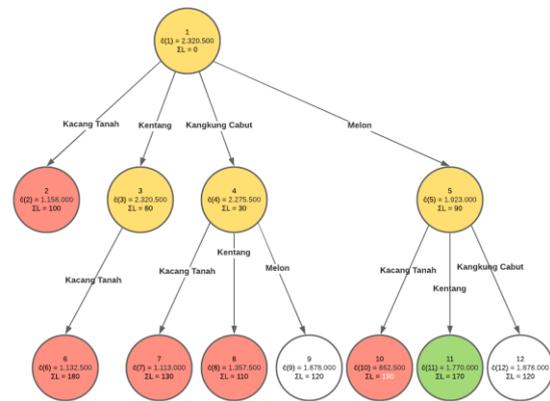
Setelah ekspansi simpul 4, perhatikan bahwa simpul 7 dapat dimatikan karena memiliki *cost* yang kurang daripada simpul solusi. Perhatikan juga simpul untuk penanaman kangkung kembali tidak ada karena dalam batasan sudah didefinisikan bahwa tidak boleh terdapat tanaman yang sama ditanam secara berurutan. Dapat dilihat pada pohon sekarang

bahwa *cost* tertinggi ada pada simpul 8, sehingga simpul 8 akan di-*expand*.



Gambar 7. Penerapan Algoritma Tahap 4
(sumber: Dibuat sendiri dengan menggunakan lucidchart)

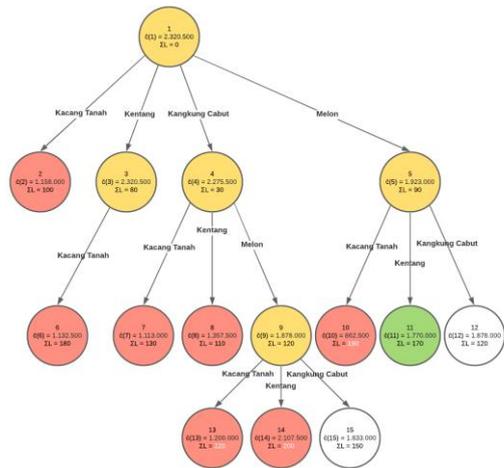
Pada ekspansi simpul 8, terdapat batasan bahwa setelah kentang (umbi) ditanam hanya kacang tanah (legum) yang dapat ditanam, namun jika kacang tanah diambil maka simpul akan langsung mati karena melanggar fungsi pembatas. Oleh karena itu simpul 8 dapat dikatakan sebagai simpul solusi. Nilai X segera dibuat 0. Karena *cost* simpul 8 lebih besar dari *cost* simpul solusi sebelumnya maka simpul 8 adalah simpul solusi yang lebih baik. Matikan semua simpul yang memiliki *cost* lebih kecil dari simpul solusi. Sekarang lakukan ekspansi pada simpul hidup dengan *cost* terbesar yaitu simpul 5.



Gambar 8. Penerapan Algoritma Tahap 5
(sumber: Dibuat sendiri dengan menggunakan lucidchart)

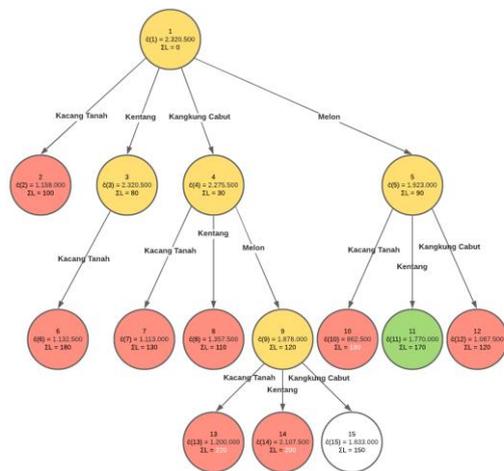
Pada ekspansi simpul 5 dapat dilihat simpul 10 langsung mati karena menyalahi fungsi pembatas. Lalu simpul 11 menjadi simpul solusi karena $\Sigma L > 152$. Pada simpul 11 nilai X menjadi 0 dan dibandingkan dengan simpul solusi sebelumnya. Karena *cost* simpul 11 lebih tinggi maka simpul 11 menjadi simpul solusi dan semua simpul yang memiliki

cost lebih kecil langsung dimatikan. Sekarang lakukan ekspansi pada simpul 9.



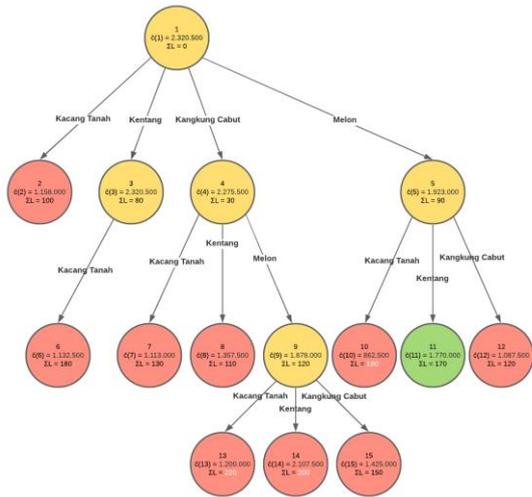
Gambar 9. Penerapan Algoritma Tahap 6
(sumber: Dibuat sendiri dengan menggunakan lucidchart)

Dapat dilihat bahwa simpul 13 dan 14 langsung dimatikan karena tidak memenuhi fungsi pembatas. Sekarang akan dilakukan ekspansi pada simpul dengan *cost* terbesar selanjutnya yaitu simpul 12.



Gambar 10. Penerapan Algoritma Tahap 7
(sumber: Dibuat sendiri dengan menggunakan lucidchart)

Perhatikan bahwa simpul 12 sudah memiliki $\Sigma L = 120$ sehingga hanya penanaman kangkung saja lah satu-satunya cara untuk meneruskan simpul, namun karena sebelumnya kangkung sudah ditanam maka simpul 12 sudah tidak dapat di-*expand* sehingga menjadi simpul solusi. Simpul 12 lalu dibandingkan dengan simpul solusi sebelumnya dan didapatkan bahwa simpul solusi sebelumnya memiliki *cost* yang lebih tinggi daripada simpul 12. Maka simpul 12 dimatikan. Sekarang simpul hidup terakhir yaitu simpul 15 akan di-*expand*.



Gambar 11. Penerapan Algoritma Tahap 8
(sumber: Dibuat sendiri dengan menggunakan lucidchart)

Pada ekspansi simpul 15, karena $\Sigma L = 150$ maka satu-satunya ekspansi yang mungkin adalah penanaman kangkung. Namun perhatikan karena kangkung sudah ditanam sebelumnya, maka tidak ada lagi anak simpul yang mungkin sehingga simpul 15 menjadi simpul solusi, nilai X menjadi 0, dan dibandingkan dengan nilai pada simpul solusi sebelumnya. Karena $cost$ simpul 15 lebih kecil dengan $cost$ simpul solusi sebelumnya maka simpul 15 dimatikan.

Karena tidak ada lagi simpul hidup, maka dapat dinyatakan ditemukan solusi berupa penanaman Melon \rightarrow Kentang sebagai solusi terbaik untuk pola tanam selama setengah tahun.

V. KESIMPULAN

Dengan menggunakan algoritma *branch and bound* didapatkan pola tanam yang paling menguntungkan dalam waktu setengah tahun atau 182 hari yaitu penanaman melon \rightarrow kentang. Perhatikan bahwa hasil ini masih jauh dari sempurna karena dalam perhitungan digunakan cukup banyak asumsi. Hasil ini juga masih harus disempurnakan agar dapat digunakan dalam kehidupan nyata, karena masih mengabaikan unsur penting seperti musim, geografis, pilihan tanah, dan lainnya.

UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas rahmatNya sehingga penulis dapat menyelesaikan makalah dengan baik dan tepat waktu. Ucapan terima kasih

juga penulis sampaikan kepada Bapak Prof. Dwi Hendratmo W., Ph.D. selaku dosen pengajar kelas 3 mata kuliah Strategi Algoritma yang sudah membimbing penulis hingga memahami materi perkuliahan khususnya pada algoritma *branch and bound*. Terimakasih juga penulis sampaikan kepada Bapak Dr. Ir. Rinaldi Munir, MT. selaku pemilik *website* yang membantu penulis memahami materi dan membantu pengerjaan tugas mata kuliah strategi algoritma. Tak lupa terimakasih juga penulis sampaikan kepada orangtua dan keluarga yang sudah menyediakan berbagai macam kebutuhan sehingga tugas makalah ini dapat terselesaikan dengan baik.

REFERENSI

- [1] Rinaldi Munir, "Algoritma Branch & Bound Bagian 1" [Online] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branch-and-Bound-2021-Bagian1.pdf> (Diakses 10 Mei 2021)
- [2] Ike Wahyuni, Sp., "Pola Rotasi Tanam Yang Benar Dan Tips Agar Panen Berhasil" [Online] cybex.pertanian.go.id/artikel/94522/pola-rotasi-tanam-yang-benar-dan-tips-agar-panen-berhasil/ (Diakses 10 Mei 2021)
- [3] Administrator Alumni, "Beragam Manfaat Sistem Rotasi Tanaman" [Online] <https://haipb.ipb.ac.id/opini/beragam-manfaat-sistem-rotasi-tanaman> (Diakses 10 Mei 2021)
- [4] Rinaldi Munir, "Breadth/Depth First Search (BFS/DFS) Bagian 1" [Online] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/BFS-DFS-2021-Bag1.pdf> (Diakses 11 Mei 2021)

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.

Cirebon, 11 Mei 2021

Jonathan Richard Sugandhi / 13519128