

Penerapan Pemrograman Dinamis dalam Menentukan Rute Penjualan Makanan Ringan Dana Usaha di Laboratorium Teknik Institut Teknologi Bandung

Ilham Firdausi Putra / 13516140
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13516140@std.stei.itb.ac.id

Abstrak—Penjualan makanan ringan dalam rangka mencari dana sudah tidak asing bagi mahasiswa. Kegiatan yang biasa disebut *danus*, singkatan dari dana usaha, tersebut merupakan suatu alternatif mencari uang untuk memenuhi kebutuhan organisasi, unit, himpunan, maupun terkadang diri sendiri. Menyempatkan diri melakukan *danus* ditengah-tengah sibuknya perkuliahan bukanlah hal yang mudah. Oleh karena itu, dibutuhkan suatu cara untuk menentukan rute penjualan yang terbesar untungnya dan terpendek rutenya. Makalah ini membahas strategi penentuan rute paling optimal dengan pendekatan pemrograman dinamis dalam model laboratorium teknik di Institut Teknologi Bandung.

Kata Kunci— Pemrograman Dinamis, Dana Usaha, Untung Maksimal, Rute Terpendek.

I. PENDAHULUAN

Dewasa ini, mahasiswa sudah tidak asing dengan istilah berhimpun. Dengan berhimpun mahasiswa dapat saling bekerja sama dan berdiskusi dalam satu tema. Pada jenjang universitas terdapat banyak pilihan berhimpun. Mulai dari himpunan jurusan, unit kegiatan mahasiswa, hingga organisasi di luar kampus. Dalam himpunan-himpunan tersebut acap kali muncul program kerja-program kerja ambisius yang memerlukan tidak sedikit dana. Untuk menutupi keadaan serba kekurangan khas mahasiswa, berbagai alternatif pengumpulan dana pun dijalani.

Bentuk dari pengumpulan dana tersebut biasa disebut *danus*. *Danus* yang merupakan singkatan dari dana usaha memiliki beragam bentuk. Mulai dari penjualan *merchandise*, penyediaan jasa, pencarian *sponsor*, hingga menjual makanan ringan. Semua dana yang terkumpul kemudian disalurkan ke himpunan atau organisasi masing-masing untuk melancarkan program kerja.

Salah satu bentuk *danus* yang paling populer adalah penjualan makanan ringan. Makanan ringan yang biasanya disediakan bervariasi mulai dari goreng-gorengan, donat, kue, *brownies*, hingga *sushi*. Hal tersebut didesain sedemikian rupa untuk memenuhi kebutuhan kudapan mahasiswa sebelum, sesudah, atau disela-sela pergantian kelas perkuliahan.

Kotak yang berisi makanan ringan tersebut biasanya dibawa oleh mahasiswa-mahasiswa sepanjang perkuliahannya. Di tengah kesibukannya menuntut ilmu, mereka menyempatkan diri memenuhi kewajibannya sebagai anggota sebuah himpunan atau organisasi. Ketika terdapat kesempatan, dengan cepat mereka mengiklankan dagangannya dengan harapan dagangannya laku dan mereka dapat lanjut fokus belajar.



Gambar 1. Penampakan sebuah kotak *danus* makanan ringan.

Sumber: <http://akhsadew.blogspot.co.id/2016/08/pasar-persaingan-danusan.html>.

Adanya masalah membagi waktu antara berjualan *danus* makanan ringan dan belajar inilah yang mendorong diperlukannya sebuah rute penjualan *danus* yang optimal. Rute penjualan *danus* yang optimal adalah yang terbesar untungnya dan terpendek rutenya. Dengan adanya rute yang optimal tersebut, mahasiswa dapat dengan cepat menyelesaikan kewajiban himpunan atau organisasinya dan kembali menuntut ilmu tanpa harus khawatir akan tidak tercapainya objektif penjualan hari itu.

Makalah ini akan mengaplikasikan pemrograman dinamis (*dynamic programming*) yang merupakan salah satu topik mata kuliah “Strategi Algoritma” untuk menyelesaikan permasalahan memilih rute terbesar untungnya dan terpendek rutenya dalam

menjual danus makanan ringan. Permasalahan ini dapat dimodelkan menjadi sebuah graf 2 dimensi yang merepresentasikan laboratorium teknik di mana penulis sehari-hari menimba ilmu.

Laboratorium teknik tersebut memiliki beberapa properti seperti adanya tangga di sebelah kiri dan kanannya di mana pengunjung dapat naik ke lantai di atasnya. Pada setiap lantai terdapat kelas-kelas di mana mahasiswa melakukan proses pembelajaran. Kelas-kelas ini akan menjadi objektif kunjungan para mahasiswa dan akan disimbolkan sebagai simpul. Pada tiap simpul tersebut terdapat sebuah nilai beratnya adalah banyaknya pembeli potensial dikelas tersebut. Sisi yang menghubungkan antar kelas pada lantai yang sama dan antar tangga memiliki nilai 1. Memaksimalkan pembeli potensial yang dikunjungi dan meminimalkan rute pengunjungan merupakan tujuan akhir dari program yang penulis ciptakan.



Gambar 2. Penampakan laboratorium teknik 5 di Institut Teknologi Bandung.

Sumber:

<https://alifanuraniputri.files.wordpress.com/2013/05/fti-007.jpg>.

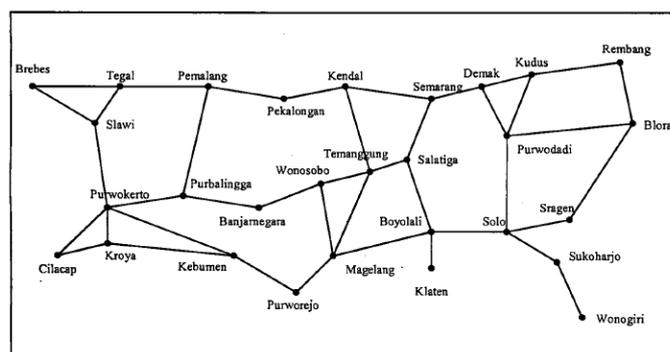
II. DASAR TEORI

A. Graf

Berdasarkan buku Matematika Diskrit edisi 3 – Rinaldi Munir[1]:

“Graf G didefinisikan sebagai pasangan himpunan (V,E) , ditulis dengan notasi $G=(V,E)$ yang dalam hal ini V adalah himpunan tidak-kosong dari simpul-simpul (*vertices* atau *node*) dan E adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul.”

Dalam praktiknya, graf banyak digunakan untuk merepresentasikan objek-objek yang dapat dinyatakan dengan simpul-simpul dan sisi-sisi. Salah satu contohnya adalah representasi jalan raya di provinsi Jawa Tengah ini.



Gambar 3. Representasi jalan raya di provinsi Jawa Tengah dalam bentuk graf tidak berarah.

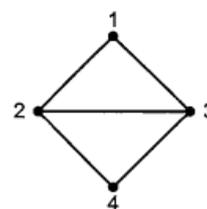
Sumber: Buku Matematika Diskrit Edisi 3-Rinaldi Munir

A. 1. Jenis-Jenis Graf

Berdasarkan ada tidaknya sisi ganda atau kalang graf dikelompokkan menjadi[1]:

1. Graf Sederhana

Graf sederhana adalah graf yang tidak mengandung sisi ganda atau kalang. Di bawah terdapat gambar contoh graf sederhana yang dapat merepresentasikan sebuah jaringan komputer dengan setiap simpul menyatakan komputer dan sisi menyatakan saluran telepon untuk berkomunikasi.

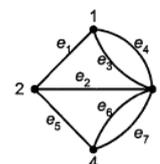


Gambar 4. Graf sederhana.

Sumber: Buku Matematika Diskrit Edisi 3-Rinaldi Munir

2. Graf Tak Sederhana

Graf tak sederhana adalah graf yang mengandung sisi ganda atau kalang. Di bawah terdapat gambar contoh graf tak sederhana yang dapat merepresentasikan sebuah jaringan telekomunikasi, di mana sisi ganda pada graf tersebut menandakan bahwa komunikasi dapat dilakukan pada jalur alternatif jika saluran pertama sedang sangat padat.



Gambar 5. Graf tak sederhana.

Sumber: Buku Matematika Diskrit Edisi 3-Rinaldi Munir

Berdasarkan ada tidaknya orientasi arah pada sisi graf dikelompokkan menjadi[1]:

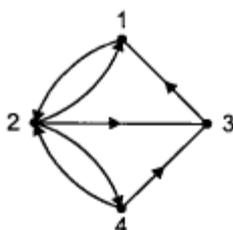
1. Graf Tak Berarah

Graf tak berarah adalah graf yang sisinya tidak memiliki orientasi arah. Gambar 4 di atas adalah salah

satu contoh graf sederhana tak berarah. Pada jaringan telepon, sisi pada graf berarah bisa diartikan saluran telepon tersebut hanya beroperasi pada arah yang ditentukan.

2. Graf Berarah

Graf berarah adalah graf yang sisinya memiliki orientasi arah. Di bawah terdapat contoh graf berarah yang dapat merepresentasikan sebuah saluran telepon. Arah pada sisi suatu graf dapat diartikan sebagai arah beroperasinya telepon tersebut. Graf berarah juga sering kali digunakan dalam merepresentasikan jalan. Arah pada graf yang merepresentasikan jalan dapat diartikan sebagai orientasi lalu lintas pada jalan tersebut. Dalam pembahasan makalah kali ini, penulis akan menggunakan graf jenis ini dalam merepresentasikan jalan.



Gambar 6. Graf berarah.

Sumber: Buku Matematika Diskrit Edisi 3-Rinaldi Munir

A. 2. Terminologi Graf

Graf memiliki beberapa terminologi (istilah) dasar. Beberapa istilah dasar yang berkaitan dengan graf adalah[1]:

1. Bertetangga

Dalam graf bertetangga didefinisikan sebagai dua buah simpul pada graf tak berarah G yang mana keduanya terhubung langsung dengan sebuah sisi. Hal ini dilambangkan dengan (u, v) yang berarti simpul u bertetangga dengan simpul v .

2. Bersisian

Dalam graf bersisian didefinisikan sebagai sembarang sisi $e = (u, v)$, menandakan e bersisian dengan simpul u dan simpul v .

3. Simpul Terpencil

Dalam graf simpul terpencil didefinisikan sebagai simpul yang tidak mempunyai sisi yang bersisian dengannya atau simpul yang tidak satupun bertetangga dengan simpul-simpul lainnya.

4. Graf Kosong

Dalam graf graf kosong didefinisikan sebagai graf yang himpunan sisinya merupakan himpunan kosong.

5. Derajat

Dalam graf derajat suatu simpul didefinisikan sebagai jumlah sisi yang bersisian dengan simpul tersebut.

Pada graf berarah, derajat simpul v dinyatakan dengan $d_{in}(v)$ dan $d_{out}(v)$.

$d_{in}(v)$ = derajat masuk = jumlah busur yang masuk ke simpul v .

$d_{out}(v)$ = derajat keluar = jumlah busur yang keluar dari simpul v .

Dengan total keseluruhan $d(v) = d_{in}(v) + d_{out}(v)$

6. Lintasan

Dalam graf lintasan didefinisikan sebagai barisan selang-seling antara simpul-simpul dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n, v_n$.

7. Siklus atau Sirkuit

Dalam graf siklus atau sirkuit didefinisikan sebagai lintasan yang berawal dan berakhir pada simpul yang sama.

8. Terhubung

Dalam graf sebuah graf disebut terhubung jika untuk setiap pasang simpul u dan v di dalam himpunan simpul-simpul V terdapat lintasan dari simpul u ke v . Jika tidak, maka graf tersebut disebut graf tak terhubung.

9. Upagraf

Dalam graf upagraf didefinisikan sebagai himpunan bagian dari himpunan simpul-simpul dan sisi-sisi sebuah graf.

10. Upagraf Merentang

Dalam graf upagraf merentang didefinisikan sebagai upagraf yang mengandung semua simpul-simpul dari graf aslinya.

11. Cut-Set

Dalam graf *cut-set* adalah himpunan sisi yang bila dibuang dari graf utamanya menyebabkan graf tersebut tidak terhubung.

12. Graf Berbobot

Graf berbobot adalah graf yang sisinya diberi sebuah harga. Dalam makalah kali ini, penulis akan menggunakan graf berbobot untuk merepresentasikan jarak.

B. Masalah Rute Optimal

Permasalahan rute optimal adalah permasalahan mencari rute terpendek berdasarkan batasan yang diberikan. Dalam makalah kali ini, batasan yang diberikan adalah untung terbanyak. Oleh karena itu misal E adalah sebuah sisi dan R adalah sebuah nilai, diberikan sebuah fungsi nilai $f: E \rightarrow R$, dan sebuah graf tidak berarah G , rute terpendek dari v ke v' adalah rute $P = (v_1, v_2, \dots, v_n)$ (di mana $v_1 = v$ dan $v_n = v'$) yang semua kemungkinan n -nya memaksimalkan jumlah dari :

$$\sum_{i=1}^{n-1} f(e_{i,i+1})$$

Dalam kasus ini, nilai setiap sisi memiliki nilai 1. Maka kasus mencari rute terpendeknya ekuivalen dengan mencari banyaknya simpul terdikit yang dikunjungi. Dari seluruh hasil yang ada, dibatasi hasilnya hanya rute yang memiliki untung maksimum. Untung terbanyak didapatkan dengan mengunjungi pengunjung potensial sebanyak mungkin, maksimum sebanyak barang yang akan dijual.

C. Pemrograman Dinamis

Diambil dari R. Munir, Program Dinamis (Dynamic Programming). 2018. [Online]:

Pemrograman Dinamis (Dynamic Programming) adalah salah satu metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan tahapan (stage) sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan. Istilah pemrograman dinamis pertama kali diperkenalkan oleh seorang professor dari Universitas Princeton yang juga bekerja di RAND corporation, bernama Richard Bellman, pada era tahun 1950-an. Istilah ini muncul karena perhitungan solusi dilakukan dengan menggunakan tabel-tabel. Pemrograman dinamis umumnya digunakan untuk menyelesaikan persoalan optimasi.

Pemecahan masalah dengan menggunakan metode pemrograman dinamis memiliki kemiripan dengan metode greedy yang juga membentuk solusi secara bertahap. Perbedaan kedua metode ini terletak pada rangkaian keputusan yang dipertimbangkan dalam penyelesaian masalah. Pada metode greedy, hanya satu rangkaian keputusan saja yang dipertimbangkan, yaitu rangkaian keputusan yang terdiri dari keputusan-keputusan yang memberikan hasil paling baik untuk setiap tahapnya (optimal lokal), yang harapannya dapat mengarah ke optimal global. Pada metode greedy keputusan yang dipilih pada tahap tertentu tidak mempertimbangkan rangkaian keputusan untuk tahapan berikutnya maupun rangkaian keputusan yang telah dipilih pada tahap sebelumnya. Salah satu contoh permasalahan yang tidak dapat diselesaikan dengan metode greedy (tidak selalu benar) adalah Integer (1/0) Knapsack Problem.

Berbeda dengan metode greedy, metode pemrograman dinamis mempertimbangkan lebih dari satu rangkaian keputusan. Rangkaian-rangkaian keputusan tersebut dibuat dengan menggunakan prinsip optimalitas yang menyatakan bahwa jika solusi total optimal, maka bagian solusi sampai tahap ke- k juga optimal. Prinsip optimalitas berarti bahwa jika kita bekerja dari tahap k ke tahap $k + 1$, kita dapat menggunakan hasil optimal dari tahap k tanpa harus kembali ke tahap awal. Berdasarkan prinsip optimalitas ini dapat dirumuskan bahwa ongkos pada tahap $k + 1$ adalah (ongkos yang dihasilkan pada tahap k) + (ongkos dari tahap k ke tahap $k + 1$). Pemilihan keputusan pada suatu tahap pada metode pemrograman dinamis dilakukan dengan menggunakan persyaratan optimasi dan kendala untuk membatasi pilihan yang harus dipertimbangkan.

Pemrograman dinamis dapat diterapkan pada persoalan yang memiliki karakteristik sebagai berikut [2].

1. Persoalan dapat dibagi menjadi beberapa tahap (stage), yang pada setiap tahap hanya diambil satu keputusan.
2. Masing-masing tahap terdiri dari sejumlah status (state) yang berhubungan dengan tahap tersebut. Status merupakan kemungkinan masukan yang ada pada tahap tersebut.
3. Hasil dari keputusan yang diambil pada setiap tahap ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya.
4. Ongkos (cost) pada suatu tahap meningkat secara teratur (steadily) dengan bertambahnya jumlah tahapan.
5. Ongkos pada suatu tahap bergantung pada ongkos tahap-tahap yang sudah berjalan dan ongkos pada tahap tersebut.
6. Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan yang dilakukan pada tahap sebelumnya.
7. Adanya hubungan rekursif yang mengidentifikasi keputusan terbaik untuk setiap status pada tahap k memberikan keputusan terbaik untuk setiap status pada tahap $k + 1$.
8. Prinsip optimalitas berlaku pada persoalan tersebut.

Terdapat dua pendekatan yang dapat digunakan untuk menyelesaikan suatu persoalan dengan pemrograman dinamis yaitu sebagai berikut [2].

1. Pemrograman dinamis maju (forward atau updown). Pemrograman dinamis bergerak mulai dari tahap 1, lalu ke tahap 2, 3, dan seterusnya sampai tahap ke- n . Rangkaian keputusannya adalah x_1, x_2, \dots, x_n .
2. Pemrograman dinamis mundur (backward atau bottom-up). Pemrograman dinamis bergerak mulai tahap n , lalu mundur ke tahap $n - 1, n - 2$, dan seterusnya sampai tahap ke 1. Rangkaian keputusannya adalah x_n, x_{n-1}, \dots, x_1 .

Kedua pendekatan ini ekuivalen dan sama-sama menghasilkan solusi yang optimum. Pada umumnya untuk mengembangkan algoritma pemrograman dinamis terdapat empat langkah yaitu sebagai berikut [2].

1. Karakteristikan struktur solusi optimum.
2. Definisikan secara rekursif nilai solusi optimal.
3. Hitung nilai solusi optimal secara maju atau mundur.
4. Konstruksi solusi optimal.

Pada implementasinya, pemrograman dinamis dapat diimplementasikan secara rekursif maupun secara iteratif dengan memanfaatkan larik (array) multi dimensi (tergantung banyaknya parameter yang merepresentasikan status untuk suatu tahap) untuk melakukan memoisasi. Implementasi secara iteratif akan membangun solusi dari bawah ke atas (bottom-up) dan menelusuri semua status yang mungkin untuk setiap tahap, sedangkan implementasi secara rekursi akan membangun solusi dari atas ke bawah (top-down) membentuk sebuah pohon dan hanya akan mengunjungi status yang dibutuhkan pada setiap tahapnya untuk mencapai solusi optimal (tidak menelusuri

semua status yang mungkin untuk setiap tahapnya).

III. IDENTIFIKASI MASALAH DAN IMPLEMENTASI

A. Identifikasi

Permasalahan ini dapat digambarkan dengan mencari rute terpendek dari simpul awal ke simpul akhir sembari mengumpulkan nilai-nilai di tiap simpul yang di kunjungi. Untuk permasalahan ini kita definisikan model laboratorium teknik direpresentasikan dengan sebuah matriks 4×6 . Titik awal kita definisikan berada di paling bawah, baris 0 kolom 0. Titik akhir yang berupa lantai di mana kelas sang mahasiswa berada dan ingin dicapai berada di lantai teratas.

Di tiap *cell* matriks terdapat nilai-nilai yang merepresentasikan modelnya. Sebuah cell dengan nilai "-1" berada di sebelah kiri dan kanan matriks, hal ini merepresentasikan tangga yang ada di laboratorium teknik Institut Teknologi Bandung. Sebuah cell dengan nilai ≥ 0 merepresentasikan banyaknya pembeli potensial pada kelas tersebut.

Oleh karena itu, sebuah laboratorium teknik di Institut Teknologi Bandung dapat dimodelkan sebagai:

-1	0	0	0	0	-1
-1	4	5	0	1	-1
-1	0	0	6	4	-1
-1	1	1	0	1	-1

Dengan titik awal (0,0) berada di kiri bawah berwarna merah dan tujuan akhir adalah lantai paling atas berwarna biru.

Pemrograman dinamis dipilih sebagai strategi penyelesaian masalah ini dikarenakan properti-propertinya yang sesuai. Strategi bruteforce tidak akan mangkus dalam menyelesaikan masalah ini dikarenakan banyaknya upa-masalah yang diulang pencariannya. Sedangkan strategi greedy tidak akan mangkus dalam menyelesaikan masalah ini dikarenakan kecenderungannya untuk jatuh ke maksimum lokal. Strategi pemrograman dinamislah yang memastikan didapatkannya jawaban yang mangkus dan sangkil.

B. Penjelasan Kode

Implementasi solusi dimulai dengan deklarasi:

1. Variable untuk menyimpan banyak lantai, kelas perlantai, makanan ringan yang akan dijual.
2. Matriks (Banyak Lantai) \times (Banyak Kelas + 2) sebagai representasi dari model laboratorium teknik.
3. Vector sebuah Pair <Integer dan Vector sebuah Integer> bernama DP. Integer pada Pair berisi banyaknya langkah yang sudah dilakukan dan Vector sebuah Integer di dalam pair berisi rute dari langkah tersebut. Vector DP ini memiliki 3 dimensi. Dimensi pertama merepresentasikan lantai. Dimensi kedua merepresentasikan datangnya dari tangga kiri atau kanan. Dimensi ketiga merepresentasikan banyaknya makanan ringannya yang tersisa.

4. Arrah sebuah Vector sebuah Pair<Integer & Integer> bernama pembeliPotensial. Vector ini digunakan untuk mengingat letak dari pembeli potensial pada suatu lantai dan kelasnya.

Langkah-langkah penyelesaian berupa:

1. Masukan berupa banyaknya lantai, kelas perlantai, makanan ringan yang akan dijual akan diterima.
2. Kemudian sebuah integer sebanyak (banyak lantai X (kelas perlantai + 2)) akan diterima dan dimasukkan ke dalam matriks.
3. Selanjutnya, pembeliPotensial diisi berdasarkan matriks.
4. Selanjutnya, sebuah fungsi rekursif pencari solusi dari titik (0,0) dengan total makanan yang akan dijual dipanggil. Fungsi ini akan melakukan:
 - a. Pengecekan apakah sudah masuk kasus basis, jika sudah sampai lantai paling atas dan semua makanan ringan sudah terjual maka kembalikan hasilnya dan rutenya. Jika makanan belum terjual habis, maka keluarkan sebuah nilai tak hingga (atau sebuah nilai atas yang didefinisikan).
 - b. Jika bukan kasus basis, lakukan pengunjungan ke kombinasi kunjungan yang mungkin (diambil dari pembeliPotensial) baik dari tangga kiri maupun tangga kanan dengan cara memanggil fungsi penyelesaian secara rekursif dengan parameter yang sudah disesuaikan.
 - c. Hasil dari tiap kunjungan dibandingkan nilai langkahnya yang terkecil.
 - d. Hasil yang terkecil dijadikan nilai pada matriks 3 dimensi DP[lantai saat itu][posisi tangga][banyak makanan tersisa]
 - e. Kembalikan nilai DP[lantai saat itu][posisi tangga][banyak makanan tersisa]
5. Hasil dari pemanggilan fungsi rekursif pencari solusi tersebut lalu cetak.

C. Eksekusi

Masukkan ke dalam program **ditebalkan**.

Masukkan:

Banyak lantai: 4
Kelas perlantai: 4
Banyak makanan: 9
-1 0 0 0 0 -1
-1 4 5 0 1 -1
-1 0 0 6 4 -1
-1 1 1 0 1 -1

Keluaran:

Banyak langkah untuk memperoleh hasil maksimum: 7
Langkah:
(0,0)->(0,1)->(0,2)->(1,2)->(2,2)->(1,2)->(0,2)->(0,3)

D. Analisis

1. Hasil keluaran merupakan rute paling mangkus dan sangkil dengan langkah terkecil dan seluruh danus makanan ringan terbeli.
2. Jika strategi yang digunakan adalah *Greedy*, program akan terjebak dengan memilih simpul (1,0) yang nampaknya lebih menguntungkan terlebih dahulu.
3. Jika menggunakan strategi *Bruteforce*, akan banyak permasalahan yang dikalkulasi ulang mengakibatkan tidak mangkusnya program.
4. Kekurangan dari program ini adalah diperlukannya pendataan mengenai hasil penjualan hari-hari sebelumnya agar didapatkan senarai pembeli potensial tiap kelas.

IV. KESIMPULAN

Penggunaan pemrograman dinamis dalam menyelesaikan masalah rute penjualan danus makanan ringan menghasilkan keluaran yang mangkus dan sangkil. Hanya saja, diperlukan data pembeli potensial tiap kelas terlebih dahulu. Dengan koordinasi antar panitia program kerja sebuah himpunan atau organisasi untuk mendata penjualan tiap harinya agar mendapatkan senarai pembeli potensial, pendekatan ini dapat digunakan untuk memaksimalkan pendapatan dana usaha sembari meminimalkan energi dan waktu yang habis dalam penjualan.

V. UCAPAN TERIMA KASIH

Pertama penulis ingin mengucapkan puji syukur kepada Tuhan Yang Maha Esa karena dengan rahmat dan karunai-Nya penulis dapat menyelesaikan makalah dengan judul “Penerapan Pemrograman Dinamis dalam Menentukan Rute Penjualan Makanan Ringan Dana Usaha di Laboratorium Teknik Institut Teknologi Bandung” ini dengan baik. Penulis juga berterima kasih kepada para dosen pengajar mata kuliah IF2211 Strategi Algoritma, Dr. Nur Ulfa Maulidevi, S.T., M. Sc., Dr. Ir. Rinaldi Munir, M.T., dan Masayu Leylia Khodra, S.T., M.T., atas bimbingan mereka selama ini dalam mengajar dan memberikan ilmu sehingga penulis mampu membuat makalah ini. Penulis juga berterima kasih kepada rekan-rekan yang telah memberikan semangat dan dorongan kepada penulis.

REFERENSI

- [1] Rinaldi Munir, *Diktat Kuliah IF2120: Matematika Diskrit*. Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2006.
- [2] R. Munir, Program Dinamis (Dynamic Programming). 2018. [Online] Tersedia dalam [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Program-Dinamis-\(2018\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Program-Dinamis-(2018).pdf). [diakses 14 Mei 2018 pukul 09.56 WIB]

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, 14 Mei 2018



Ilham Firdausi Putra / 13516140