

Memaksimalkan Keuntungan *Shuttle Service* Melalui Optimalisasi Penjadwalan dengan Program Dinamis dan Binary Search

Makalah IF2211 Strategi Algoritma

Muhammad Raihan Asyraf Desanto (13517027)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung Jl. Ganesha 10 Bandung 40132, Indonesia

13517027@std.itb.ac.id

Abstrak—*Shuttle service* merupakan jasa pelayanan transportasi dari tempat ke tempat (*point to point*) yang hanya memberangkatkan dan menurunkan penumpang di tempat tertentu. Hampir 24 jam jasa *shuttle service* beroperasi, Setiap selang beberapa menit jasa *shuttle service* menerima kedatangan dan keberangkatan beberapa bus yang ada. Dengan adanya keterbatasan waktu dan tempat, perlu adanya penjadwalan keberangkatan dan kedatangan yang terstruktur dan optimal agar menghasilkan keuntungan bagi jasa *shuttle service*. Makalah ini akan membahas penggunaan program dinamis sebagai strategi dalam optimalisasi penjadwalan *shuttle service* agar mendapatkan keuntungan yang maksimal.

Kata Kunci—*Shuttle service*; Program dinamis; Optimalisasi; Penjadwalan; Maksimal; Keuntungan

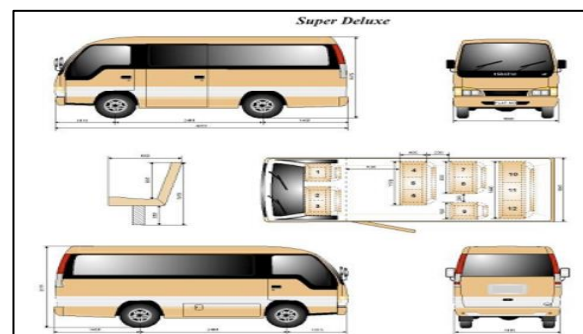
I. PENDAHULUAN

Sebuah perusahaan *shuttle service* merupakan sebuah bisnis pelayanan jasa yang sangat menjanjikan karena bisnis ini melayani jasa transportasi yang hanya memberangkatkan dan menurunkan penumpang pada tempat yang sudah ditentukan. Dengan adanya penjadwalan yang lebih frekuentif dan lebih bervariasi daripada menggunakan jasa pelayanan kereta api akibat adanya jumlah armada bus atau mobil travel yang disediakan oleh perusahaan *shuttle service* lebih banyak daripada kereta yang disediakan jasa pelayanan kereta api. Sehingga banyak sekali penumpang, bahkan perantau yang menggunakan jasa *shuttle service* ini. Biasanya perusahaan *shuttle service* ini menyediakan ruang tunggu bagi para penumpang untuk menunggu kedatangan bus atau mobil travel dari tempat yang lain. Setelah itu akan diumumkan bahwa bus atau mobil travel tersebut siap untuk diberangkatkan. Maka para penumpang yang telah menunggu di ruang tunggu bergegas masuk ke dalam bus atau mobil travel. Akhirnya ruang tunggu tersebut akan diisi oleh penumpang dengan jadwal keberangkatan yang selanjutnya. Begitulah rangkaian aktivitas dari jasa *shuttle service*. Siklus ini terjadi secara berulang-ulang setiap harinya.

Kendala yang muncul kemudian adalah keterbatasan waktu yang diperlukan untuk tiap keberangkatan dan kedatangan, serta jumlah maksimum dari bus atau mobil travel yang dapat

ditampung oleh perusahaan *shuttle service* yang bersangkutan. Jika jumlah kendaraan yang dapat di tampung oleh perusahaan *shuttle service* terlalu sedikit, maka akan sedikit pula jadwal keberangkatan dan kedatangan yang disediakan oleh perusahaan *shuttle service* tersebut, sehingga akan dihasilkan keuntungan yang sedikit. Di sisi lain penambahan lahan baru untuk tempat penampungan kendaraan juga membutuhkan biaya yang tidak sedikit sehingga memerlukan pertimbangan yang matang. Permasalahan ini akan menjadi semakin kompleks. Untuk itu, diperlukan sebuah strategi yang optimal agar mendapatkan keuntungan yang maksimal dengan mempertimbangkan paket pelayanan *shuttle service* yang lebih menguntungkan dengan cara melakukan penjadwalan keberangkatan dan kedatangan yang efektif dan efisien. Dalam permasalahan optimalisasi ini, program dinamis tepat untuk diimplementasikan.

Optimalisasi penentuan jadwal dalam penjadwalan pada proses pelayanan *shuttle service* dapat dilakukan menggunakan program dinamis. Program dinamis akan mengatur penjadwalan proses kedatangan serta keberangkatan pada sebuah perusahaan *shuttle service* agar lebih efektif dan efisien. Dengan adanya optimalisasi penjadwalan ini, diharapkan perusahaan *shuttle service* ini akan mendapatkan keuntungan yang maksimal. Strategi optimalisasi ini juga dapat dikembangkan untuk merumuskan kebijakan pembangunan lahan baru apabila jumlah pelayanan keberangkatan dan kedatangan yang harus dilayani semakin meningkat dengan sehingga menghasilkan keuntungan yang maksimal. Selain itu, juga untuk meningkatkan kinerja program dalam melakukan penjadwalan, dalam penerapan program dinamis tersebut perlu ditambahkan *binary search*.



Gambar 1. Ilustrasi dimensi dan denah tempat duduk dari mobil travel.
 Sumber: <http://www.donymagazine.com/category/bus/>

II. DASAR TEORI

A. Program Dinamis

Program dinamis (*dynamic programming*) merupakan salah satu metode pemecahan masalah dengan cara menguraikan masalah yang rumit menjadi beberapa upa masalah yang lebih sederhana. Dengan kata lain, permasalahan tersebut dibagi menjadi sekumpulan tahapan (*stage*) sedemikian sehingga solusi dari persoalan tersebut dapat dipandang dari serangkaian keputusan yang saling berkaitan. Akibatnya, apabila terdapat beberapa upa masalah yang memiliki kemiripan, maka tidak perlu melakukan komputasi ulang untuk mencari solusinya. Seorang hanya perlu mencari solusi yang sebelumnya pernah dipecahkan. Teknik menyimpan solusi dari upa masalah disebut *memorization* [1].

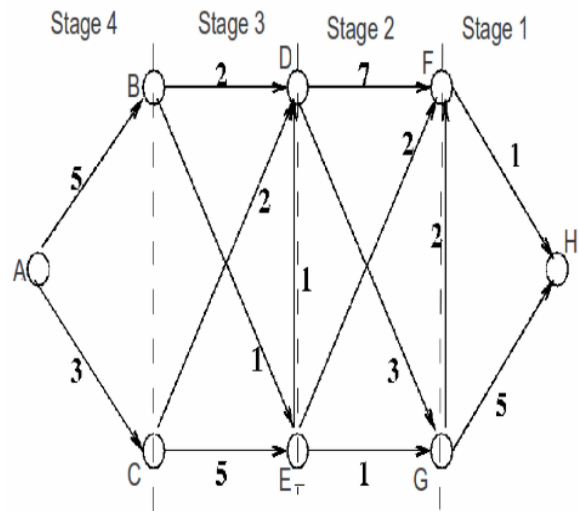
Program dinamis umumnya digunakan untuk menyelesaikan persoalan optimasi. Program dinamis memiliki kemiripan dengan metode *greedy* yang juga membentuk rangkaian solusi secara bertahap. Perbedaan kedua metode ini terletak pada rangkaian keputusan yang diambil pada tiap tahapannya. Pada metode *greedy*, hanya satu rangkaian keputusan saja yang dipertimbangkan, yaitu rangkaian keputusan yang terdiri dari keputusan yang hanya menghasilkan optimal lokal pada tiap tahapannya, sehingga diharapkan dapat mengarah ke solusi optimal global. Pada metode *greedy* keputusan yang dipilih tidak mempertimbangkan rangkaian keputusan untuk tahapan berikutnya maupun rangkaian keputusan sebelumnya. Sehingga, terkadang penyelesaian dengan metode *greedy* tidak selalu optimum.

Berbeda dengan metode *greedy*, metode pemrograman dinamis mempertimbangkan lebih dari satu rangkaian keputusan. Beberapa rangkaian keputusan tersebut dibuat dengan menggunakan prinsip optimalitas yang berbunyi bahwa solusi optimal dari sebuah persoalan dapat dihasilkan dari gabungan solusi optimal tiap upa masalah atau dengan kata lain, 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 $k + 1$) [2]. Pemilihan keputusan yang diambil pada tiap tahapan metode pemrograman dinamis dilakukan dengan menggunakan persyaratan optimasi dan kendala untuk membatasi pilihan yang harus dipertimbangkan. Untuk menentukan solusi optimal sebuah persoalan yang kompleks dengan metode program dinamis, perlu diturunkan solusinya secara rekursif dari upa masalah yang lebih kecil.

Program dinamis diterapkan pada persoalan yang memiliki karakteristik sebagai berikut [2]:

1. Persoalan dapat dibagi menjadi beberapa tahapan (*stage*), yang pada setiap tahapannya hanya diambil satu keputusan.

2. Setiap tahap terdiri dari sejumlah status (*state*) yang berhubungan dengan tahapan tersebut. Pada umumnya, status merupakan berbagai jenis kemungkinan masukan yang ada pada tahap tersebut. Jumlah status bisa berhingga (*finite*) atau tidak berhingga (*infinite*).



Gambar 2. Ilustrasi status dan tahapan pada program dinamis.
 Sumber: <http://www.donymagazine.com/category/bus/>

3. Hasil dari keputusan yang diambil pada setiap tahapan ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya.
4. Biaya (*cost*) pada suatu tahap meningkat secara teratur dengan bertambahnya jumlah tahapan yang telah dilalui.
5. Biaya pada suatu tahapan bergantung pada biaya pada tahap-tahap yang sudah berjalan sebelumnya dan biaya pada tahap tersebut.
6. Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan yang dilakukan pada tahap sebelumnya.
7. Terdapat 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 dalam menyelesaikan permasalahan dengan menggunakan algoritma program dinamis, yaitu [2].

1. Program dinamis maju (*forward* atau *up-down*) merupakan pendekatan dengan memproses tiap tahapan mulai dari tahap yang pertama hingga tahapan terakhir ($ke-n$). Runtutan peubah keputusan adalah x_1, x_2, \dots, x_n .
2. Program dinamis mundur (*backward* atau *bottom-up*) merupakan pendekatan dengan memproses tiap

tahapan mulai dari tahap yang ke- n hingga tahapan proses pertama (ke-1). Runtutan peubah keputusan adalah x_n, x_{n-1}, \dots, x_1 .

Secara umum, terdapat empat langkah yang perlu dilakukan dalam mengembangkan algoritma program dinamis [1]:

1. Karakteristikkan struktur dari solusi optimal.
2. Mendefinisikan nilai solusi optimal secara rekursif.
3. Hitung nilai solusi optimal, biasanya dengan pendekatan program dinamis maju.
4. Membangun solusi optimal dari informasi yang dihitung.

B. Binary Search

Binary search adalah salah satu algoritma dasar dalam ilmu komputer. Binary search digunakan dalam mencari sebuah nilai pada array yang terurut nilainya. Binary search bekerja dengan menggunakan subsequence kontigu yang mana pasti terdapat nilai yang kita cari yang kita sebut ruang pencarian. Awalnya, ruang pencarian binary search adalah seluruh array. Pada tiap tahap, binary search membandingkan median dari subsequence terhadap nilai yang kita cari. Berdasarkan hasil perbandingan, kita dapat memangkas ruang pencarian sampai setengahnya. Sebagai contoh bila kita mencari nilai 55 dari

0, 5, 13, 19, 22, 41, 55, 68, 72, 81, 98

Pertama kita tetapkan ruang pencarian dari indeks ke-0 sampai indeks ke-10. Lalu kita bandingkan nilai median (41) dengan nilai yang kita cari (55). Karena nilai median < nilai yang kita cari, pastilah nilai yang kita cari berada di sebelah kanan median. Ruang pencarian kita dapat dipangkas mencari

55, 68, 72, 81, 98

Kita lakukan kembali hal yang sama dan kita dapatkan

55, 68

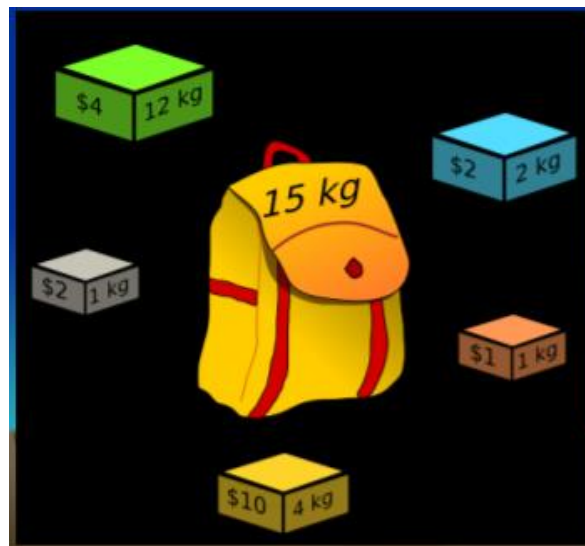
Bergantung cara kita menentukan median, kita akan mendapat indeks nilai yang kita cari pada tahap ini atau tahap Makalah IF2211 Strategi Algoritma, Semester II Tahun 2015/2016 selanjutnya. Dan kita dapati bahwa nilai yang kita cari berada pada indeks ke-6.

C. Integer(1/0) Knapsack Problem

Integer (1/0) Knapsack Problem adalah permasalahan memilih n buah objek untuk dimasukkan ke dalam sebuah knapsack (karung, tas, buntelan, dsb) yang memiliki kapasitas bobot K sedemikian rupa sehingga objek-objek yang dipilih tidak melebihi kapasitas knapsack namun keuntungan yang didapatkan maksimal. Setiap objek yang akan dipilih memiliki properti bobot (weight) w_i dan keuntungan atau nilai (value) v_i

. Pada Integer (1/0) Knapsack Problem, setiap objek hanya dapat dipilih sekali. Secara matematis, Integer (1/0) Knapsack Problem dapat dituliskan sebagai berikut.

Maksimasi $\sum_{i=1}^n v_i x_i$ dengan batasan (constraint) $\sum_{i=1}^n w_i x_i \leq K$ n $i=1$ dengan $x_i \in \{0, 1\}$ Solusi: $\{x_1, x_2, \dots, x_n\}$



Gambar 3. Ilustrasi integer knapsack problem [4]

Integer (1/0) Knapsack Problem dapat digunakan untuk memodelkan berbagai persoalan optimasi yang solusinya berupa subset yang memiliki batasan (constraint) tertentu. Contoh permasalahan yang dapat dimodelkan sebagai Integer (1/0) Knapsack Problem adalah penyimpanan barang di gudang dan pengaturan pengiriman barang seefisien mungkin untuk mendapatkan keuntungan terbesar. Dalam teori komputasi, Integer (1/0) Knapsack Problem adalah satu persoalan yang termasuk ke dalam persoalan NP-Complete. Ini berarti belum ditemukan sebuah algoritma yang tepat dan mangkus (dapat menyelesaikan persoalan dalam waktu polinomial) untuk menyelesaikan Integer (1/0) Knapsack Problem. Walaupun belum ditemukan algoritma yang mangkus untuk menyelesaikan persoalan ini, terdapat algoritma yang berjalan dalam waktu pseudo-polynomial yang dapat menyelesaikan permasalahan ini, yaitu dengan menggunakan pemrograman dinamis (dynamic programming).

Selain dengan menggunakan pemrograman dinamis (dynamic programming), persoalan ini juga dapat diselesaikan dengan metodemetode lain, yaitu exhaustive search (mengenumerasi semua kemungkinan solusi), branch and bound, dan greedy (pendekatan dengan greedy tidak selalu menghasilkan solusi yang optimal, tetapi dapat digunakan mengaproksimasi bobot (cost) yang dapat digunakan untuk pendekatan branch and bound).

Terdapat beberapa variasi dari Integer (1/0) Knapsack Problem yang muncul dari berbagai aplikasi dari Integer (1/0) Knapsack Problem. Variasi-variasi ini muncul dengan mengganti parameter-parameter dari permasalahan semula,

misalnya jumlah tujuan, jumlah knapsack yang digunakan, atau jumlah batasan (constraint). Variasi-variasi dari Integer (1/0) Knapsack Problem diantaranya adalah Bounded Knapsack Problem (setiap objek dapat dipilih sampai maksimal c kali), Unbounded Knapsack Problem (tidak ada batas pemilihan untuk setiap objek), Multi-objective Knapsack Problem (tujuan / hal yang ingin dioptimasi lebih dari 1), Multi-dimensional Knapsack Problem (batasan / constraint lebih dari 1), dan Multiple Knapsack Problem (jumlah knapsack lebih dari 1).

D. Shuttle Service

Shuttle service merupakan sebuah perusahaan yang melayani jasa pelayanan angkutan dengan cara mengumpulkan para penumpang pada suatu titik atau tempat tertentu. Untuk diberangkatkan dan diantar hingga ke tempat tertentu yang sudah ditetapkan. Hal ini berbeda dengan travel, dimana layanan angkutan yang diberikan dengan cara menjemput penumpang/klien dari rumahnya dan kemudian mengantarnya ke tempat alamat yang dituju.

Adapun penjadwalan yang dilakukan oleh perusahaan *shuttle service* adalah dengan cara memberangkatkan bus atau mobil travel dari tempat itu dan menerima kedatangan bus atau mobil travel dari tempat lainnya. Dengan penjadwalan yang sudah dirancang sedemikian rupa hingga menjadi jadwal yang terbaik.

SCHEDULES			
Dipati Ukur > Soekarno-Hatta		Soekarno-Hatta > Dipati Ukur	
01:15	10:15	07:15	16:15
02:15	11:15	08:15	17:15
03:15	12:15	09:15	18:15
04:15	13:15	10:15	19:15
05:15	14:15	11:15	20:15
06:15	15:15	12:15	21:15
07:15	16:15	13:15	22:15
08:15	17:15	14:15	23:15
09:15		15:15	

Gambar 4. Ilustrasi *integer knapsack problem*
 Sumber: <https://www.cititrans.co.id/soekarnohatta-airport>

III. STUDI KASUS DAN ANALISIS

A. Studi Kasus Permasalahan penjadwalan pada *shuttle service*

Tentunya dalam setiap kali keberangkatan dan kedatangan sebuah bus atau mobil travel membutuhkan jeda waktu antara proses keberangkatan dan kedatangan sehingga dibutuhkan pengelolaan jadwal se-efektif dan se-efisien mungkin guna menghasilkan keuntungan yang sebesar-besarnya tanpa terjadi konflik antar jadwal kedatangan dan keberangkatan. Selain itu, faktor lain yang mempengaruhi jumlah keuntungan yang didapatkan oleh perusahaan *shuttle service* adalah banyaknya armada serta kapasitas penampungan armada dari perusahaan *shuttle service* tersebut. Karena dengan banyaknya kapasitas penampungan armada, semakin banyak pula armada bus atau mobil travel yang dapat ditampung. Sehingga, dengan banyaknya armada bus dan mobil travel dapat meminimumkan terjadinya konflik antar jadwal keberangkatan dan kedatangan.

Permasalahan yang didapatkan adalah biaya yang dibutuhkan untuk menambah armada dan menambahkan kapasitas penampungan armada pada perusahaan *shuttle service* tidaklah sedikit. Untuk itu pada masalah kali ini, penulis hanya membatasi hanya untuk memaksimalkan keuntungan dengan pemilihan penjadwalan tanpa dipengaruhi oleh faktor banyaknya armada bus atau mobil travel tersebut.

Agar mendapatkan penjadwalan keberangkatan dan kedatangan dari bus atau mobil travel yang memaksimalkan keuntungan dalam penerapannya perlu kita ketahui, beberapa syarat untuk memenuhi hasil keuntungan maksimal yang diharapkan, yaitu:

1. Penjadwalan haruslah teratur, artinya tidak ada jadwal kedatangan ataupun keberangkatan yang saling bentrok dalam satu waktu yang sama.
2. Perlu disimpan status pada tiap tahapannya (*stage*) yang berisi pilihan solusi oleh jadwal yang tidak bentrok dengan jadwal sebelumnya.
3. Pada tiap tahapan (*stage*) perlu dipilih sebuah solusi yang paling optimal. Sehingga membentuk hasil yang optimal secara keseluruhan.

Untuk studi kasus, penulis akan memodelkan optimalisasi penjadwalan pada *shuttle service* dalam memaksimalkan keuntungan dengan sebuah perusahaan *shuttle service* fiktif yang bernama SI-X. Perusahaan *shuttle service* ini memiliki data berupa opsi jadwal travel dengan keuntungan setiap satu kali keberangkatan dan kedatangan sebuah bus atau mobil travel. Asumsikan keuntungan sudah termasuk keuntungan bersih setiap satu kali kedatangan dan keberangkatan.

TABEL I. Data Opsi Penjadwalan pada *shuttle service* SI-X dalam sehari

Keuntungan Bersih	Waktu Keberangkatan	Waktu Kedatangan
Rp. 1.200.000	5:00	7:00
Rp. 1.100.000	5:30	10:00
Rp. 1.300.000	6:00	10:00
Rp. 1.000.000	10:00	12:00
Rp. 1.500.000	9:00	11:30
Rp. 2.000.000	11:00	15:00
Rp. 1.300.000	12:00	14:30
Rp. 1.100.000	14:30	16:30
Rp. 1.700.000	16:00	18:30
Rp. 1.500.000	17:00	19:00
Rp. 2.100.000	18:00	21:00

```

JADWAL TERPILIH
Rp 1.200.000 | 5:00 | 7:00
Rp 1.500.000 | 9:00 | 11:30
Rp 1.300.000 | 12:00 | 14:30
Rp 1.700.000 | 16:00 | 18:30
Rp 2.100.000 | 18:00 | 21:00
Total keuntungan maksimum: Rp 7.800.000

```

Dengan pemodelan tersebut langkah – langkah untuk penyelesaian persoalan tersebut adalah:

1. Pertama-tama, perlu dibuat list berisi jadwal keberangkatan dan kedatangan beserta keuntungannya untuk memudahkan pemrosesan secara rekursif.
2. Sebelum melakukan proses perlu di urutkan menaik terlebih dahulu berdasarkan waktu kedatangan. Agar memudahkan implementasi algoritma *binary search*.
3. Untuk tiap tahapan perlu disyaratkan pemilihan tiap kandidat yang memenuhi syarat dengan cara membandingkan apakah akan terjadi konflik jika memilih jadwal travel tersebut dengan menggunakan algoritma *binary search*.
4. Kemudian untuk, tiap tahapan perlu disimpan status, pada tahapan (*stage*). Untuk nantinya dipilih solusi yang menghasilkan keuntungan maksimum tiap tahapannya.
5. Setelah itu, solusi akan direkonstruksi berdasarkan hasil solusi optimum dari tiap tahapan yang terpilih.

B. Hasil dan Analisis

Dari hasil eksekusi program dengan memasukan data opsi jadwal keberangkatan dan kedatangan pada Tabel I, dimana terdapat sepuluh pilihan opsi jadwal. Diperoleh *output* sebagai berikut.

Dari hasil tersebut didapatkan bahwa dari proses tersebut telah didapatkan jadwal kedatangan dan keberangkatan dari sebuah perusahaan *shuttle service* fiktif yang bernama SI-X. dengan melakukan program dinamis solusi sudah pasti terjamin selalu optimal jika dibandingkan dengan algoritma *greedy*. Pada program ini, penulis melakukan pendekatan persoalan *Integer (1/0) Knapsack Problem* untuk optimalisasi dengan batas pada tiap tahapannya (*stage*) yang merupakan. Pengecekan antara waktu kedatangan akhir dari jadwal sebelumnya dengan waktu keberangkatan dari jadwal saat ini.

Dengan adanya optimalisasi tersebut, penjadwalan menjadi lebih efisien dan efektif karena, persoalan tersebut merupakan persoalan yang kompleks dan dapat diselesaikan dengan cara manual tetapi akan memakan waktu yang sangat lama. Oleh karena itu, diharapkan dengan optimalisasi menggunakan program dinamis dan *binary search*, dapat mengotomasi pembuatan jadwal serta, menghemat waktu dalam pembuatan jadwal kedatangan dan keberangkatan *shuttle service*.

IV. KESIMPULAN

Berdasarkan hasil uji coba implementasi algoritma program dinamis dengan penggabungan algoritma pencarian *binary search*. Terdapat beberapa hal yang dapat disimpulkan:

1. Permasalahan Optimasi pada penjadwalan keberangkatan dan kedatangan dari perusahaan *shuttle service* guna memaksimalkan keuntungan. Dapat diselesaikan dengan metode program dinamis.
2. Dalam optimasi kinerja program, dapat ditingkatkan melalui *binary search* karena pencarian untuk setiap Batasan antara kedatangan dan keberangkatan program tidak perlu di cek semuanya.
3. Penyelesaian menggunakan program dinamis dapat menentukan tingkat efektivitas dan efisiensi suatu perusahaan *shuttle service* guna peningkatan pelayanan bagi pengguna jasa. Hasil penyelesaian juga bermanfaat bagi perusahaan *shuttle service* untuk merumuskan kebijakan jadwal keberangkatan dan kedatangan dari travel tersebut.

UCAPAN TERIMA KASIH

Pertama-tama penulis mengucapkan syukur kepada Allah SWT karena berkat rahmat dan pertolongan-Nya penulis dapat menyelesaikan perkuliahan Strategi Algoritma beserta tugas-tugasnya dengan lancar pada semester ini. Kemudian penulis juga berterima kasih kepada dosen-dosen pengajar mata kuliah IF2211 Strategi Algoritma, Drs. Nur Ulfa Maulidevi, Dr. Ir. Rinaldi Munir, M.T., dan Dr. Masayu Leylia Khodra ST., MT. atas bimbingan dan dukungan yang telah diberikan kepada

penulis beserta ilmu-ilmu bermanfaat yang telah diajarkan. Penulis juga berterima kasih kepada keluarga dan teman-teman yang telah memberikan dukungan, motivasi, dan bantuan kepada penulis selama pengerjaan makalah ini.

REFRENSI

- [1] Levitin, Anany. 2011. Introduction to The Design and Analysis of Algorithms. London: Pearson.
- [2] Munir, Rinaldi. Diklat Kuliah IF2211 Strategi Algoritma. Program Studi Teknik Informatika ITB. 2018.
- [3] Steven and Felix Halim. 2010. Competitive Programming 1st Edition. Singapore: NUS.
- [4][http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/Algoritma-Greedy-\(2019\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/Algoritma-Greedy-(2019).pdf) (diakses pada 25/04/2019)

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, 29 April 2012



M. Raihan Asyraf D. 13517027