

Penerapan Algoritma Program Dinamis pada Penjadwalan Aktivitas

Krishna Aurelio Noviandri - 13516108

Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13516108@std.stei.itb.ac.id

Abstract—Setiap hari, seseorang pasti memiliki aktivitas yang perlu atau ingin dikerjakan. Akan tetapi, tidak semua aktivitas bisa dikerjakan karena setiap aktivitas butuh waktu dan waktu setiap orang sangatlah terbatas. Oleh karena itu, melakukan penjadwalan terhadap aktivitas yang akan dikerjakan adalah hal yang perlu dilakukan. Supaya mendapatkan jadwal aktivitas yang memanfaatkan waktu dengan efisien atau optimum, dapat diterapkan algoritma program dinamis karena penjadwalan aktivitas adalah kasus yang membutuhkan solusi paling optimum.

Keywords—Aktivitas, Pemilihan, Penjadwalan, Efisien, Optimum, Program dinamis.

I. INTRODUCTION

Melakukan penjadwalan bukanlah hal yang asing untuk dilakukan pada dunia modern ini. Banyak sekali hal-hal yang membutuhkan penjadwalan seperti aktivitas sehari-hari, jadwal produksi barang, maintenance mesin, scheduling pada sistem operasi, pekerjaan sampingan dan sebagainya. Dengan menerapkan penjadwalan, waktu yang digunakan untuk melakukan aktivitas-aktivitas tersebut akan lebih efisien atau optimum.

Pada dasarnya penjadwalan yaitu memilih aktivitas mana saja yang akan dilakukan dalam kurun waktu tertentu. Tentu saja tidak mungkin ada dua aktivitas yang dapat dilakukan dalam waktu yang bersamaan. Oleh karena itu, mungkin saja aktivitas-aktivitas yang berada pada kurun waktu yang bersamaan hanya dilakukan sebagian saja, sisanya tidak dilakukan atau ditunda. Oleh karena itu, diperlukan algoritma yang dapat mengoptimalkan penjadwalan. Salah satunya yaitu algoritma program dinamis. Berikut contoh penjadwalan:

Mulai	Selesai	Kegiatan
07:00	07:30	Olahraga
07:30	07:45	Mandi
07:45	08:00	Makan
08:00dst.

Table 1: Contoh jadwal kegiatan sehari-hari

II. LANDASAN TEORI

A. Sejarah Awal Program Dinamis

Istilah program dinamis, pertama kali diperkenalkan pada era 1950-an oleh Richard Bellman seorang professor di Universitas Princeton dan juga bekerja pada RAND corporation, perlu diketahui bahwa RAND corporation pada era itu merupakan suatu perusahaan yang dibentuk untuk menawarkan analisis dan riset untuk angkatan bersenjata Amerika Serikat. Saat itu, Bellman bekerja di bidang pengambilan keputusan multi tahap (*multistage desicion process*) dan mengerjakan beberapa metode matematis, beberapa tahun kemudian setelah Bellman berada di RAND, lahirah istilah pemrograman dinamis. Istilah ini tidak secara langsung berhubungan dengan pemrograman, melainkan digunakan sebagai judul *project* yang kemudian yang diusulkan RAND corporation pada Angkatan Udara Amerika Serikat.

B. Pengertian Program Dinamis Menurut para ahli

Menurut Dimayati (1992), program dinamis adalah suatu teknik matematis yang biasanya digunakan untuk membuat suatu keputusan dari serangkaian keputusan yang saling berkaitan. Tujuan utama model ini adalah untuk mempermudah penyelesaian persoalan optimasi yang mempunyai karakteristik tertentu.

Sedangkan menurut Siagian (2003), program dinamis adalah satu teknik matematis yang digunakan untuk mengoptimalkan proses pengambilan keputusan secara bertahap-ganda. Dalam teknik ini, keputusan yang menyangkut suatu persoalan dioptimalkan secara bertahap dan bukan secara sekaligus. Inti teknik ini adalah membagi satu persoalan atas beberapa bagian persoalan yang dalam program dinamis disebut tahap, kemudian memecahkan tiap tahap dengan mengoptimalkan keputusan atas tiap tahap sampai seluruh persoalan telah terpecahkan.

C. Program Dinamis

Program dinamis adalah metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan tahapan atau *stage*. Sehingga solusi dari persoalan dapat dipandang sebagai serangkaian keputusan yang saling berkaitan.

Karakteristik Penyelesaian persoalan dengan Program Dinamis ada 3:

1. Jumlah pilihan yang mungkin diambil berhingga
2. solusi pada setiap tahap dibangun berdasarkan solusi dari tahap sebelumnya
3. menggunakan prinsip optimasi dan kendala untuk membatasi jumlah pilihan yang harus dipertimbangkan pada suatu tahap.

Pada dasarnya, semua persoalan algoritma program dinamis diselesaikan dengan cara yang mirip dengan algoritma greedy. Kedua algoritma ini sama-sama mengambil keputusan pada setiap tahapan mencari solusi. Akan tetapi, pada algoritma greedy solusinya tidak selalu optimum karena hanya satu rangkaian keputusan saja yang diambil. Sedangkan pada algoritma program dinamis lebih dari satu rangkaian keputusan yang dipertimbangkan dan solusi yang dihasilkan akan selalu optimum. Berikut adalah contoh gambar graph yang menunjukkan perbedaan hasil dari algoritma greedy dan algoritma pemograman dinamis:

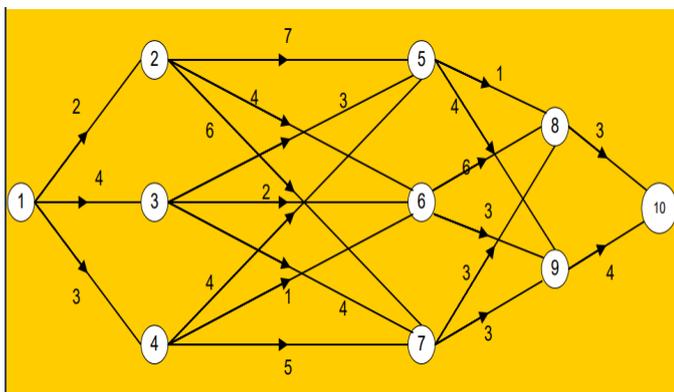


Illustration 1: Contoh Graph

Dari graph tersebut, jalur terpendek yang dihasilkan jika menggunakan algoritma greedy adalah 1-2-6-9-10 dengan total bobot 13. sedangkan dengan menggunakan algoritma program dinamis didapat tiga solusi yaitu 1-3-5-8-10, 1-4-5-8-10, dan 1-4-6-9-10 dengan total bobot 11.

D. Prinsip Optimalitas

Pada pemograman dinamis, rangkaian keputusan akan dibentuk dengan menggunakan prinsip optimalitas. Prinsip optimalitas yaitu prinsip dimana jika solusi total optimal, maka solusi sebelum-sebelumnya atau solusi sampai tahap ke-k juga harus optimal. Jadi, untuk mencari suatu solusi yang pada tahap k+1 maka dengan mengetahui solusi pada tahap ke k saja sudah cukup, tidak perlu lagi mencari dari tahap awal. Bisa dituliskan dengan rumus cost pada tahap k = (cost yang dihasilkan sampai tahap k) + (cost yang dibutuhkan dari tahap k ke tahap k+1). Berikut ilustrasi dari prinsip optimalitas:

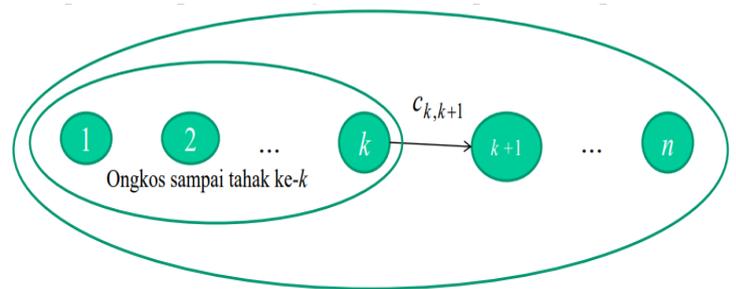


Illustration 2: Gambaran prinsip optimalitas

E. Karakteristik Persoalan Program Dinamis

Karakteristik Persoalan-Persoalan yang dapat diselesaikan dengan menggunakan program dinamis:

1. Persoalan dapat dibagi menjadi beberapa tahap atau *stage*, yang pada setiap saat hanya diambil satu keputusan. Tahapan untuk setiap persoalan tidak selalu menggambarkan hal yang sama.
2. Pada masing-masing tahap terdiri dari sejumlah status atau *state* yang berhubungan dengan tahap tersebut. Status secara umum adalah kemungkinan-kemungkinan masukkan yang ada pada tahap tersebut.
3. Hasil dari suatu keputusan pada setiap tahapan diubah dari status tahapan tersebut ke status pada tahapan berikutnya.
4. Cost pada suatu tahap meningkat secara teratur dengan bertambahnya jumlah tahapan. Tidak mungkin pada tahap berikutnya cost lebih sedikit.
5. Cost pada suatu tahap ditentukan oleh cost pada tahapan sebelumnya dan cost pada tahap tersebut.
6. Keputusan terbaik pada suatu tahap tidak bergantung atau independen terhadap keputusan yang dilakukan pada tahap sebelumnya. Keputusan terbaik pada tahap k+1 independen terhadap keputusan pada tahap k.
7. Terdapat hubungan rekursif yang mengidentifikasi keputusan terbaik untuk setiap status pada tahap k memberikan keputusan terbaik untuk setiap status pada k+1.
8. Prinsip optimalitas berlaku pada persoalan tersebut

F. Dua Jenis Pendekatan Program Dinamis

Ada dua jenis pendekatan yang digunakan pada program dinamis:

1. Program Dinamis Maju (*forward* atau *up-down*)

Bergerak mulai dari tahap 1 terus maju sampai tahap ke 2, 3 dan seterusnya sampai tahap ke n. dengan variabel keputusannya adalah X_1, X_2, \dots, X_n . Prinsip yang digunakan yaitu cost pada tahap ke k+1 = (ongkos yang dihasilkan pada tahap ke k) + (cost dari tahap k ke tahap k+1), dengan $k = 1, 2, 3, \dots, n-1$.

2. Program Dinamis Mundur (*backward* atau *botto-up*)

Bergerak mulai dari tahap n terus mundur ke tahap n-1, n-2 dan seterusnya sampai tahap ke 1. dengan variabel keputusannya adalah $X_n, X_{n-1}, X_{n-2}, \dots, X_1$. Prinsip yang digunakan yaitu cost pada tahap k = (ongkos yang dihasilkan pada tahap k + 1) + (ongkos dari tahap k+1 ke tahap k). dengan k = n, n-1, ..., 1.

III. PEMBAHASAN

Pada pembahasan ini, akan dijelaskan beberapa kasus persoalan penjadwalan aktivitas menggunakan algoritma program dinamis. Untuk menyelesaikan persoalan penjadwalan aktivitas terdapat beberapa teknik program dinamis yang dapat digunakan. Penyelesaian persoalan pada pembahasan ini akan menggunakan perogram dinamis dengan pendekatan maju.

A. Penyelesaian dengan graf multistage

Mulai	Akhir	Aktivitas
01:00	02:00	Aktivitas 1
02:00	03:00	Aktivitas 2
02:00	04:00	Aktivitas 3
03:00	06:00	Aktivitas 4
03:00	05:00	Aktivitas 5
04:00	05:00	Aktivitas 6
04:00	06:00	Aktivitas 7
05:00	08:00	Aktivitas 8
05:00	07:00	Aktivitas 9
06:00	08:00	Aktivitas 10

Table 2: Contoh Kasus Penjadwalan Aktivitas

Dari contoh kasus diatas, dapat dibuat graf multistage dimana setiap simpul pada graf tersebut menyatakan status sedangkan setiap V_1, V_2, V_3, \dots menyatakan tahap. Berikut

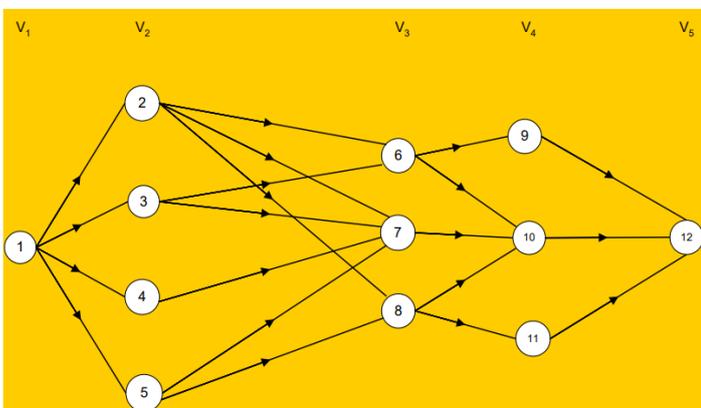


Illustration 3: Contoh graf multistage contohnya :

Dari kasus di tabel 2, dapat dibentuk graf multistage dimana simpul adalah status menyatakan waktu pada saat ini,

setiap aktivitas menjadi sebuah sisi berarah yang menyambungkan simpul status awal aktivitas dengan simpul status akhir aktivitas dilihat dari waktu awal dan akhir suatu aktivitas. Cost untuk setiap sisi atau aktivitas ditentukan dengan waktu akhir dikurangi dengan waktu awal. Setiap pergantian tahapan selalu diambil satu aktivitas sampai tidak ada lagi aktivitas yang dapat diambil. Berikut adalah graf multistage yang dihasilkan:

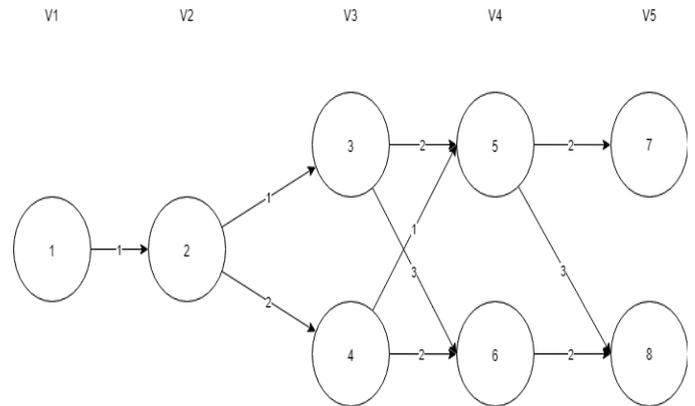


Illustration 4: Graf Multistage untuk persoalan 1

Dari Ilustrasi 4, dapat dilihat bahwa persoalan pada tabel 2 terdiri dari 8 buah status yang berarti terdapat 8 buah keadaan jam yang mungkin, terdapat 10 sisi yang menyatakan 10 aktivitas yang mungkin diambil setiap sisi. Setiap sisi boleh diberikan identitas untuk mempermudah mengidentifikasi sisi tersebut menyatakan aktivitas yang mana, dan terdiri dari 5 tahapan yang menyatakan jumlah aktivitas yang diambil pada setiap tahapannya.

Karena solusi optimum yang dicari adalah untuk memanfaatkan waktu dengan sebaik mungkin, maka hasil yang dicari adalah hasil yang memiliki cost yang terbesar. Dari hal tersebut kita bisa dapat fungsi rekurens program dinamis sebagai berikut:

Basis : $f_1(s) = c_{x_1, s}$

Rekurens : $f_k(s) = \max_{x_k} \{c_{x_k, s} + f_{k-1}(x_k)\}$

Keterangan:

- X_k : Peubah keputusan pada tahap k
- $c_{x_k, s}$: Cost atau bobot sisi dari x_k ke s
- $f_k(s)$: nilai maksimum pada state tersebut

Dengan rumus rekurens dan graf multistage, persoalan pada tabel 2 dapat dicari solusinya. Berikut tabel-tabel solusi untuk setiap tahapan:

Tahap 1:

s	Solusi Optimum	
	f ₁ (s)	Aktivitas
2	1	1

Tahap 2:

s\X ₂	C _{X₂} + f ₁ (X ₂)		Solusi optimum	
	2	3	f ₂ (s)	Aktivitas
3	2	3	2	(1,2)
4	3	4	3	(1,3)

Tahap 3:

s\X ₃	C _{X₃} + f ₂ (X ₃)			Solusi optimum	
	3	4	5	f ₃ (s)	Aktivitas
5	4	5	6	4	(1,2,5), (1,3,6)
6	5	6	7	5	(1,2,4), (1,3,7)

Tahap 4:

s\X ₄	C _{X₄} + f ₃ (X ₄)				Solusi optimum	
	5	6	7	8	f ₃ (s)	Aktivitas
7	6	-	8	9	6	(1,2,5,9), (1,3,6,9)
8	7	7	8	9	7	(1,2,5,8), (1,3,6,8), (1,2,4,10), (1,3,7,10)

Dari perhitungan tabel tersebut, kita bisa mendapatkan 4 kemungkinan solusi yang optimum yaitu 1-2-5-8, 1-3-6-8, 1-2-4-10, dan 1-3-7-10 dengan total pemakaian waktu sebesar 7 jam. berikut adalah jadwal yang dapat dibentuk dari solusi tersebut:

Solusi 1:

Mulai	Akhir	Aktivitas
01:00	02:00	Aktivitas 1
02:00	03:00	Aktivitas 2
03:00	05:00	Aktivitas 5
05:00	08:00	Aktivitas 8

Solusi 2:

Mulai	Akhir	Aktivitas
01:00	02:00	Aktivitas 1

02:00	04:00	Aktivitas 3
04:00	05:00	Aktivitas 6
05:00	08:00	Aktivitas 8

Solusi 3:

Mulai	Akhir	Aktivitas
01:00	02:00	Aktivitas 1
02:00	03:00	Aktivitas 2
03:00	06:00	Aktivitas 4
06:00	08:00	Aktivitas 10

Solusi 4:

Mulai	Akhir	Aktivitas
01:00	02:00	Aktivitas 1
02:00	04:00	Aktivitas 3
04:00	06:00	Aktivitas 7
06:00	08:00	Aktivitas 10

Akan tetapi, cara tersebut tidak bisa digunakan untuk semua persoalan penjadwalan karena ada persoalan penjadwalan yang tidak bisa dibuat graf multistage.

B. Penyelesaian dengan graf berarah

Mulai	Akhir	Aktivitas
01:00	02:00	Aktivitas 1
01:00	03:00	Aktivitas 2
02:00	03:00	Aktivitas 3
02:00	04:00	Aktivitas 4
02:00	05:00	Aktivitas 5
03:00	06:00	Aktivitas 6
03:00	05:00	Aktivitas 7
04:00	05:00	Aktivitas 8
04:00	06:00	Aktivitas 9
04:00	08:00	Aktivitas 10
05:00	06:00	Aktivitas 11
05:00	08:00	Aktivitas 12
05:00	07:00	Aktivitas 13
06:00	08:00	Aktivitas 14
07:00	08:00	Aktivitas 15

Table 3: Contoh Kasus Penjadwalan Aktivitas 2

Dari persoalan diatas, graf multistage tidak dapat dibentuk, sehingga yang terbentuk hanyalah graf berarah. berikut adalah graf berarah yang dibentuk:

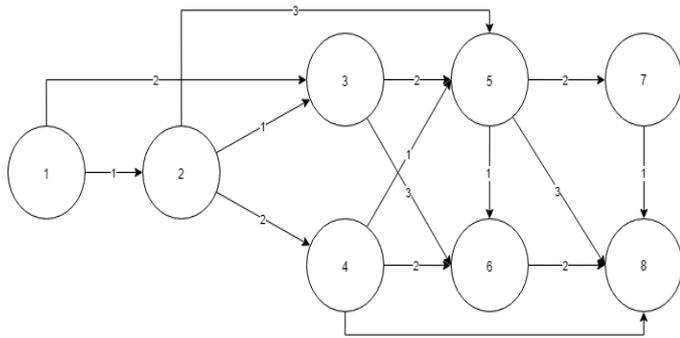


Illustration 5: Graf Berarah persoalan 2

Dari Illustrasi 5, dapat dilihat bahwa persoalan 2 tersebut terdiri dari 8 buah status yang berarti terdapat 8 buah keadaan jam yang mungkin, terdapat 15 sisi yang menyatakan 15 aktivitas yang mungkin diambil setiap sisi. Setiap sisi boleh diberikan identitas untuk mempermudah mengidentifikasi sisi tersebut menyatakan aktivitas yang mana. Karena graf tersebut bukanlah graf multi tahap, maka setiap tahapan yang dibentuk menandakan jumlah sisi atau banyaknya aktivitas yang telah diambil sampai ke tahap tersebut.

Karena solusi optimum yang dicari adalah untuk memanfaatkan waktu dengan sebaik mungkin, maka hasil yang dicari adalah hasil yang memiliki cost yang terbesar. Dari hal tersebut kita bisa dapat fungsi rekurens program dinamis sebagai berikut:

Basis : $f_1(s) = c_{x_1s}$

Rekurens : $f_k(s) = \max_{x_k} \{c_{x_k s} + f_{k-1}(x_k)\}$

Keterangan:

- x_k : Peubah keputusan pada tahap k
- $c_{x_k s}$: Cost atau bobot sisi dari x_k ke s
- $f_k(s)$: nilai maksimum pada state tersebut

Dengan rumus rekurens dan graf berarah, persoalan pada tabel 2 dapat dicari solusinya. Berikut tabel-tabel solusi untuk setiap tahapan:

Tahap 1:

s	Solusi Optimum	
	$f_1(s)$	x_1^*
2	1	1
3	2	1

Tahap 2:

$s \setminus x_2$	$c_{x_2 s} + f_1(x_2)$		Solusi optimum	
	2	3	$f_2(s)$	x_2^*
3	2	2	2	2 atau 3
4	3	-	3	2
5	4	4	4	2 atau 3
6	-	5	5	3

Tahap 3:

$s \setminus x_3$	$c_{x_3 s} + f_2(x_3)$				Solusi optimum	
	3	4	5	6	$f_3(s)$	x_3^*
5	4	4	-	-	4	3,4,atau 5
6	5	5	5	-	5	3,4,5
7	-	-	6	-	6	5
8	-	7	7	7	7	4,5,atau 6

Tahap 4:

$s \setminus x_4$	$c_{x_4 s} + f_3(x_4)$				Solusi optimum	
	5	6	7	8	$f_4(s)$	x_4^*
6	5	-	-	-	5	5
7	6	-	-	-	6	5
8	7	7	7	-	7	5,6,atau 7

Tahap 5:

$s \setminus x_5$	$c_{x_5 s} + f_4(x_5)$			Solusi optimum	
	6	7	8	$f_5(s)$	x_5^*
8	7	7	-	7	6 atau 7

Dari perhitungan tabel tersebut, terdapat banyak solusi yang mungkin berdasarkan jumlah tahapannya. Solusi dengan jumlah aktivitas paling sedikit atau tiga dapat diambil dari tahap 3 menggunakan x_3^* untuk mentrace ulang jalur yang diambil. Solusi yang didapat dari tahap 3 yaitu 1-4-10, 2-7-12, dan 2-6-14 dengan pemakaian waktu sebesar 7 jam. Berikut jadwal yang dapat dibentuk dari solusi tsb:

Solusi 1:

Mulai	Akhir	Aktivitas
01:00	02:00	Aktivitas 1
02:00	04:00	Aktivitas 4

04:00	08:00	Aktivitas 10
-------	-------	--------------

Solusi 2:

Mulai	Akhir	Aktivitas
01:00	03:00	Aktivitas 2
03:00	05:00	Aktivitas 7
05:00	08:00	Aktivitas 12

Solusi 3:

Mulai	Akhir	Aktivitas
01:00	02:00	Aktivitas 2
02:00	04:00	Aktivitas 6
04:00	08:00	Aktivitas 14

IV. KESIMPULAN

Dari Pembahasan diatas, dapat disimpulkan bahwa dengan menggunakan algoritma program dinamis dapat digunakan untuk mendapatkan solusi optimum dalam pemilihan aktivitas yang akan dijadwalkan. Penyelesaian menggunakan graf multistage lebih mudah diimplementasikan dan lebih cepat dibandingkan Penyelesaian menggunakan graf berarah. Tetapi, menggunakan graf berarah bisa menyelesaikan lebih banyak persoalan dibandingkan dengan menggunakan graf multistage. Sehingga pemilihan cara penjadwalan aktivitas menggunakan algoritma program dinamis harus melihat kondisi dari persoalan dan hasil yang ingin didapat.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada bapak Rinaldi Munir, ibu Masayu, dan ibu Ulfa selaku dosen mata kuliah IF 2211 Strategi Algoritma yang telah mengajarkan pembelajaran mata kuliah ini selama satu semester. Selain itu, penulis juga mengucapkan terima kasih kepada keluarga dan teman-teman yang telah membantu dan sama-sama berjuang untuk menyelesaikan makalah ini.

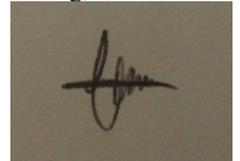
REFERENCES

- [1] [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Program-Dinamis-\(2018\)f.pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Program-Dinamis-(2018)f.pdf) diakses pada tanggal 13 Mei 2018
- [2] <https://sutrinoadityo.wordpress.com/2013/10/12/program-dinamis/> diakses pada tanggal 12 Mei 2018

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, 13 Mei 2018



Krishna Aurelio Nobiandri
13516108