Optimasi Skor Dengan Program Dinamis Pada Permasalahan Soal Tes Berbobot

(Variasi Integer Knapsack Problem)

Hengky Surya Angkasa - 13518048
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail : 13518048@std.stei.itb.ac.id

***Abstrak*—Program Dinamis adalah suatu metode yang digunakan untuk optimasi, antara minimasi atau maksimasi dalam sebuah persoalan. Persoalan dibagi menjadi tahap-tahap dan dicari solusi optimum per tahap. Pendekatan Program Dinamis dapat dilakukan dengan Program Dinamis Maju dan Program Dinamis Mundur. Salah satu persoalan yang dapat diselesaikan oleh Program Dinamis adalah *Integer Knapsack Problem*. *Integer Knapsack Problem* merupakan persoalan dimana ada beberapa objek yang memiliki profit dan bobot untuk dimasukkan ke dalam *Knapsack* yang memiliki kapasistas tertentu. Objektifnya adalah optimasi profit dengan konstrain kapasitas. Salah satu variasi *Integer Knapsack Problem* yang berkaitan dengan kehidupan sehari-hari adalah persoalan atau permasalahan soal tes berbobot yang memiliki skor tertentu tiap soal dan dibatasi oleh waktu untuk menyelesaikan suatu soal.**

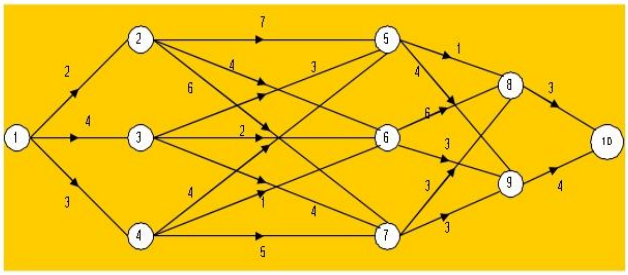
Keywords—Program; Dinamis; Optimasi; Tahap; Bobot; Solusi; Knapsack; Profit; Soal; Tes

I. PENDAHULUAN

Salah satu algoritma yang dipelajari dalam mata kuliah Strategi Algoritma merupakan Program Dinamis atau *Dynamic Programming*. Program dalam hal ini bukanlah berkaitan dengan pemograman. Dinamis dalam hal ini berkaitan dengan pencarian solusi dilakukan dengan perhitungan dengan menggunakan tabel yang dapat berkembang. Program Dinamis sendiri pertama dikembangkan oleh Richard E. Bellman pada tahun 1957. Program Dinamis merupakan suatu algoritma yang menarik karena solusi dipecah menjadi beberapa tahap sehingga ketika melakukan penggabungan solusi akan mendapat hasil yang maksimum.

Dibandingkan dengan algoritma lain, seperti algoritma *Greedy*, Program Dinamis dapat menghasilkan lebih dari satu rangkaian keputusan dan algoritma *Greedy* hanya menghasilkan satu rangkaian keputusan.

Program Dinamis dapat menyelesaikan berbagai persoalan yang dapat dibagi menjadi tahap-tahap. Persoalan yang dapat diselesaikan seperti mencari lintasan terpendek dari suatu simpul ke simpul lain. Adapun dapat menyelesaikan persoalan *Integer Knapsack Problem*. Untuk persoalan lebih kompleks, Program Dinamis juga dapat menyelesaikan persoalan *Travelling Salesman Problem (TSP)* dan *Capital Budgeting*.



Gambar 1.1 Persoalan Lintasan Terpendek
Sumber:http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Program-Dinamis-(2018).pdf

Integer Knapsack Problem merupakan suatu persoalan dimana terdapat berbagai objek dengan bobot dan profit yang dapat dimasukkan ke *knapsack* yang memiliki maksimal bobot yang dapat ditampung. Akibatnya, kita perlu maksimasi profit dengan syarat objek-objek terpilih tidak melebihi kapasitas *knapsack*. *Integer Knapsack Problem* dapat diselesaikan dengan berbagai algoritma yang telah dipelajari dalam mata kuliah Strategi Algoritma. Salah satunya menggunakan Program Dinamis. Dengan Program Dinamis, dapat memasukkan objek-objek dengan profit maksimum.

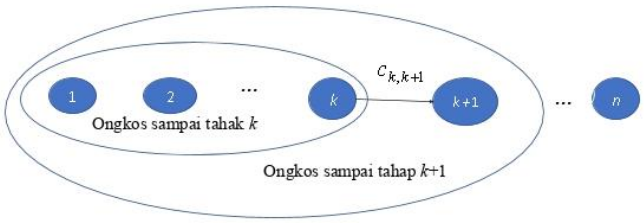
Salah satu variasi pada kehidupan sehari-hari *Integer Knapsack Problem* adalah persoalan soal tes yang berbobot atau memiliki skor tertentu dan waktu penyelesaian tertentu. Persoalan ini mengharapkan profit atau skor yang optimum. Skor yang optimum berguna dalam suatu penerimaan atau seleksi, jadi persoalan ini cukup penting.

II. LANDASAN TEORI

A. Program Dinamis

Program dinamis merupakan suatu metode pemecahan masalah dimana masalah dipecah menjadi beberapa tahap sehingga setiap tahap terdapat solusi. Solusi masalah merupakan suatu rangkaian keputusan yang masing-masing saling berkaitan. Akibatnya, terdapat lebih dari satu rangkaian keputusan untuk dipertimbangkan.

Prinsip Optimalitas adalah jika solusi total optimal, maka bagian solusi tahap ke-k juga optimal [1]. Lebih lengkapnya, jika berpindah dari suatu tahap k ke tahap k+1, maka hasil optimal dari tahap k dapat digunakan tanpa harus kembali ke tahap awal. Ongkos pada tahap k+1 adalah ongkos yang didapat dari tahap k ditambah dengan ongkos dari tahap k ke tahap k+1. Rangkaian keputusan dibuat menggunakan Prinsip Optimalitas. Hal tersebut agar rangkaian keputusan yang didapat sudah optimal.



Gambar 1.2 Visualisasi Ongkos pada Program Dinamis. Sumber:http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Program-Dinamis-(2018).pdf

Terdapat tujuh karakteristik persoalan Program Dinamis[1].

1. Pertama, persoalan dapat dibagi menjadi beberapa tahap dan pada tahap tersebut diambil satu keputusan.
2. Kedua, pada masing-masing tahap terdapat status-status yang berkorelasi dengan tahap tersebut. Umumnya, status merupakan kemungkinan masukan pada suatu tahap.
3. Ketiga, hasil keputusan yang diambil pada setiap tahap ditransformasi dari status bersangkutan ke status berikutnya pada tahap berikutnya.
4. Keempat, ongkos pada suatu tahap meningkat secara teratur sejalan dnegan bertambahnya tahapan.
5. Kelima, ongkos pada suatu tahap bergantung pada ongkos tahap-tahap yang sudah berjalan dan ongkos pada tahap tersebut.
6. Keenam, terdapat hubungan rekursif dalam identifikasi keputusan terbaik untuk tiap status pada tahap k sehingga memberikan keputusan terbaik untuk tiap status pada tahap k+1.
7. Ketujuh, Prinsip Optimalitas berlaku pada persoalan tersebut.

Pada Program Dinamis terdapat dua pendekatan. Pertama, program dinamis maju. Program dinamis maju merupakan pendekatan dimana proses penyelesaian persoalan dimulai dari tahap 1, 2, ..., n-1, n. Kedua, program dinamis mundur merupakan pendekatan dimana proses penyelesaian persoalan dimulai dari tahap n, n-1, ..., 2, 1. Misalkan X1, X2, ..., Xn merupakan variabel keputusan yang ditentukan pada tahap 1, 2, ..., n. Pada pendekatan program dinamis maju, rangkaian keputusan adalah X1, X2, ..., Xn. Sedangkan, pada pendekatan program dinamis mundur, rangkaian peubah keputusan adalah Xn, Xn-1, ..., 1.

Secara umum, berikut langkah-langkah pengembangan algoritma program dinamis untuk menyelesaikan suatu persoalan[1].

1. Pertama, karakterisasi struktur solusi optimal seperti tahap, variabel keputusan, status.
2. Kedua, mendefinisikan secara rekursif nilai solusi optimal, dimana terdapat hubungan nilai optimal suatu tahap dengan tahap sebelumnya.
3. Ketiga, menghitung nilai solusi optimal secara maju atau mundur.
4. Keempat, rekonstruksi solusi optimal, dilakukan secara mundur.

B. Fungsi rekursif

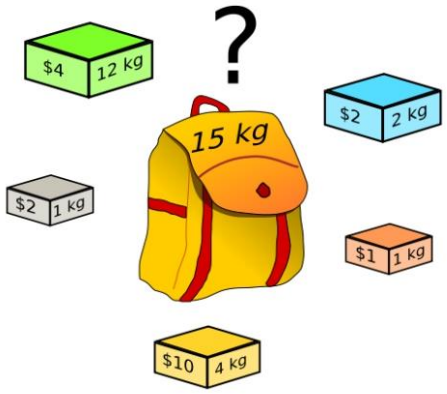
Fungsi rekursif merupakan fungsi yang memanggil dirinya sendiri, baik secara langsung maupun lewat fungsi lain[2]. Proses pemanggilan tersebut disebut rekursi. Didalam fungsi rekursif harus terdapat basis dan rekurensnya. Basis suatu fungsi rekursif boleh lebih dari 1. Berikut contoh fungsi rekursif F untuk menghitung bilangan fibonacci.

F(n) = 0, n = 0 ; F(n) = 1, n = 1 (BASIS)

F(n) = F(n-1) + F(n-2), n > 1 (REKURENS)

C. Integer Knapsack Problem

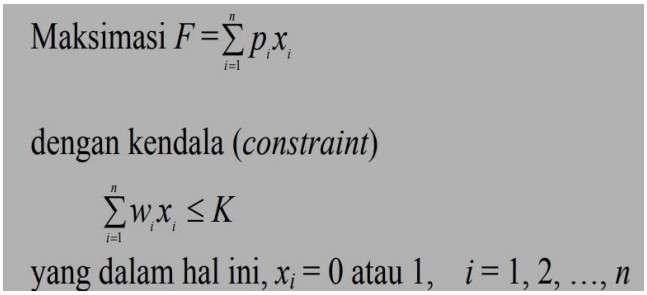
Integer Knapsack Problem merupakan suatu persoalan dimana terdapat n buah objek dan sebuah *knapsack*. Tiap objek memiliki atribut bobot (*w*) dan atribut profit (*p*). *Knapsack* memiliki atribut kapasitas bobot maksimum (*K*). Akibatnya, tidak semua objek dapat muat dalam *knapsack*. Hanya objek-objek yang memiliki profit maksimal dan total bobot kurang-dari-sama-dengan kapasitas *knapsack* yang dimasukkan.



Gambar 1.3 Ilustrasi *Integer Knapsack Problem*. Sumber: http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Brute-Force-(2016).pdf

Pendekatan umum, terdapat himpunan bagian dari semua objek yang muat dalam *knapsack*. Dari hal tersebut, dapat dicari himpunan yang memiliki profit terbesar.

Solusi persoalan dinyatakan sebagai himpunan (bisa juga vektor) X = {X1, X2, ..., Xn} dimana Xi = 1 jika objek ke-i dipilih dan Xi = 0 jika objek ke-i tidak dipilih.



Gambar 1.3 Formulasi matematis pada persoalan *Integer Knapsack Problem*. Sumber: http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Brute-Force-(2016).pdf

III. PERSOALAN

A. *Inisialisasi Persoalan*

Dalam suatu tes, terdapat lima soal yang dapat dikerjakan oleh peserta. Setiap soal tidak wajib dijawab. Peserta terpilih merupakan peserta (dapat lebih dari satu) yang mendapat skor tertinggi. Peserta hanya memiliki waktu 3 jam.

Dari persoalan diketahui bahwa "kapasitas *knapsack"* atau batas waktu tes adalah 3 jam (K = 3). Adapun "bobot dan profit objek" atau batas waktu suatu soal dan skor suatu soal dinyatakan dalam tabel berikut.

Soal-i	Batas Waktu (jam)	Skor
1	1	25
2	2	40
3	1	30
4	2	35
5	1	20

Tabel 1.1 Soal beserta batas waktu dan skor.

B. *Analisis Persoalan dengan Program Dinamis*

Pada persoalan diatas, Tahap(k) adalah proses mengerjakan suatu soal dalam tes. Terdapat lima tahap pada persoalan tersebut. Status(y) adalah sisa waktu tes setelah mengerjakan suatu soal pada tahap sebelumnya.

Untuk perhitungan profit (skor), dilakukan perbandingan skor yang didapat pada tahap ke k (p_k) ditambah $f_{k-1}(y-w_k)$ dengan skor pada tahap k-1 atau $f_{k-1}(y)$. Jika $p_k + f_{k-1}(y-w_k)$ lebih kecil dari $f_{k-1}(y)$, maka soal ke k tidak perlu dikerjakan. Jika $p_k + f_{k-1}(y-w_k)$ lebih besar dari $f_{k-1}(y)$, maka soal ke k perlu dikerjakan.

Fungsi rekursif untuk perhitungan skor,

$f_0(y) = 0, y = 0, 1, 2, 3$

$f_k(y) = -\infty, y < 0$

$f_k(y) = \max\{ f_{k-1}(y) , p_k + f_{k-1}(y-w_k) \}, k = 1, 2, 3, 4, 5$

Keterangan:

$f_k(y)$ adalah skor optimum pada tahap k untuk sisa waktu sebesar y

$f_k(y) = -\infty$ adalah skor untuk y negatif

$f_0(y) = 0$ adalah skor apabila tidak mengerjakan semua soal

Solusi akan dinyatakan dalam vektor $X = (X_1,X_2,X_3,X_4,X_5)$ dimana X_i elemen dari {0,1}, dimana 0 untuk soal tidak dipilih untuk dikerjakan dan 1 untuk soal yang harus dikerjakan. Dimana, i adalah soal ke-i.

IV. HASIL DAN PEMBAHASAN

1. Menggunakan pendekatan Program Dinamis Maju (Pengecekan dari soal pertama)

Tahap 1

Untuk soal nomor 1, memiliki batas waktu 1 jam (w) dalam pengerjaan dan skor (p) sebesar 25

$f_1(y) = \max\{ f_0(y) , p_1 + f_0(y-w_1) \}$
$f_1(y) = \max\{ f_0(y) , 25 + f_0(y-1) \}$

y			Solusi Optimum	
	$f_0(y)$	$25 + f_0(y-1)$	$f_1(y)$	$(X_1^*,X_2^*,X_3^*,X_4^*,X_5^*)$
0	0	$-\infty$	0	(0,0,0,0,0)
1	0	25	25	(1,0,0,0,0)
2	0	25	25	(1,0,0,0,0)
3	0	25	25	(1,0,0,0,0)

Tabel 1.2 Hasil Tahap 1

Tahap 2

Untuk soal nomor 2, memiliki batas waktu 2 jam (w) dalam pengerjaan dan mendapat skor 40 (p)

$f_2(y) = \max\{ f_1(y) , p_2 + f_1(y-w_2) \}$

$f_2(y) = \max\{ f_1(y) , 40 + f_1(y-2) \}$

y			Solusi Optimum	
	$f_1(y)$	$40 + f_1(y-2)$	$f_2(y)$	$(X_1^*,X_2^*,X_3^*,X_4^*,X_5^*)$
0	0	-∞	0	(0,0,0,0,0)
1	25	-∞	25	(1,0,0,0,0)
2	25	40	40	(0,1,0,0,0)
3	25	65	65	(1,1,0,0,0)

Tabel 1.3 Hasil Tahap 2

Tahap 3

Untuk soal nomor 3, memiliki batas waktu pengerjaan 1 jam (w) dan mendapatkan skor 30 (p)

$f_3(y) = \max\{ f_2(y) , p_3 + f_2(y-w_3) \}$
$f_3(y) = \max\{ f_2(y) , 30 + f_2(y-1) \}$

y			Solusi Optimum	
	$f_2(y)$	$30 + f_2(y-1)$	$f_3(y)$	$(X_1^*,X_2^*,X_3^*,X_4^*,X_5^*)$
0	0	-∞	0	(0,0,0,0,0)
1	25	30	30	(0,0,1,0,0)
2	40	55	55	(1,0,1,0,0)
3	65	70	70	(0,1,1,0,0)

Tabel 1.4 Hasil Tahap 3

Tahap 4

Untuk soal nomor 4, memiliki batas waktu pengerjaan 2 jam (w) dan mendapatkan skor 35 (p)

$f_4(y) = \max\{ f_3(y) , p_4 + f_3(y-w_4) \}$
$f_4(y) = \max\{ f_3(y) , 35 + f_3(y-2) \}$

y			Solusi Optimum	
	$f_3(y)$	$35 + f_3(y-2)$	$f_4(y)$	$(X_1^*,X_2^*,X_3^*,X_4^*,X_5^*)$
0	0	-∞	0	(0,0,0,0,0)
1	30	-∞	30	(0,0,1,0,0)
2	55	35	55	(1,0,1,0,0)
3	70	65	70	(0,1,1,0,0)

Tabel 1.5 Hasil Tahap 4

Tahap 5

Untuk soal nomor 5, memiliki batas waktu pengerjaan 1 jam (p) dan mendapat skor 20 (p)

$f_5(y) = \max\{ f_4(y) , p_5 + f_4(y-w_5) \}$
$f_5(y) = \max\{ f_4(y) , 20 + f_4(y-1) \}$

y			Solusi Optimum	
	$f_4(y)$	$20 + f_4(y-1)$	$f_5(y)$	$(X_1^*,X_2^*,X_3^*,X_4^*,X_5^*)$
0	0	-∞	0	(0,0,0,0,0)
1	30	20	30	(0,0,1,0,0)
2	55	50	55	(1,0,1,0,0)
3	70	75	75	(1,0,1,0,1)

Tabel 1.6 Hasil Tahap 5

2. Menggunakan pendekatan Program Dinamis Mundur (Pengecekan dari soal terakhir)

Tahap 1

Untuk soal nomor 5, memiliki batas waktu pengerjaan 1 jam (w) dan mendapatkan skor 35 (p)

$f_1(y) = \max\{ f_0(y) , p_1 + f_0(y-w_1) \}$

$f_1(y) = \max\{ f_0(y) , 20 + f_0(y-1) \}$

y			Solusi Optimum	
	$f_0(y)$	$20 + f_0(y-1)$	$f_1(y)$	$(X_1^*,X_2^*,X_3^*,X_4^*,X_5^*)$
0	0	-∞	0	(0,0,0,0,0)
1	0	20	20	(0,0,0,0,1)
2	0	20	20	(0,0,0,0,1)
3	0	20	20	(0,0,0,0,1)

Tabel 1.7 Hasil Tahap 1

Tahap 2

Untuk soal nomor 4, memiliki batas waktu pengerjaan 2 jam (w) dan mendapatkan skor 35 (p)

$f_2(y) = \max\{ f_1(y) , p_2 + f_1(y-w_2) \}$
$f_2(y) = \max\{ f_1(y) , 35 + f_1(y-2) \}$

y			Solusi Optimum	
	$f_1(y)$	$35 + f_1(y-2)$	$f_2(y)$	$(X_1^*,X_2^*,X_3^*,X_4^*,X_5^*)$
0	0	-∞	0	(0,0,0,0,0)
1	20	-∞	20	(0,0,0,0,1)
2	20	35	35	(0,0,0,1,0)
3	20	55	55	(0,0,0,1,1)

Tabel 1.8 Hasil Tahap 2

Tahap 3

Untuk soal nomor 3, memiliki batas waktu pengerjaan 1 jam (w) dan mendapatkan skor 30 (p)

$f_3(y) = \max\{ f_2(y) , p_3 + f_2(y-w_3) \}$
$f_3(y) = \max\{ f_2(y) , 30 + f_2(y-1) \}$

y		Solusi Optimum		
	$f_2(y)$	$30 + f_2(y-1)$	$f_3(y)$	$(X_1^*,X_2^*,X_3^*,X_4^*,X_5^*)$
0	0	-∞	0	(0,0,0,0,0)
1	20	30	30	(0,0,1,0,0)
2	35	50	50	(0,0,1,0,1)
3	55	65	65	(0,0,1,1,0)

Tabel 1.9 Hasil Tahap 3

Tahap 4

Untuk soal nomor 2, memiliki batas waktu pengerjaan 2 jam (w) dan mendapatkan skor 40 (p)

$f_4(y) = \max\{ f_3(y) , p_4 + f_3(y-w_4) \}$
$f_4(y) = \max\{ f_3(y) , 40 + f_3(y-2) \}$

y			Solusi Optimum	
	$f_3(y)$	$40 + f_3(y-2)$	$f_4(y)$	$(X_1^*,X_2^*,X_3^*,X_4^*,X_5^*)$
0	0	-∞	0	(0,0,0,0,0)
1	30	-∞	30	(0,0,1,0,0)
2	50	40	50	(0,0,1,0,1)
3	65	70	70	(0,1,1,0,0)

Tabel 1.10 Hasil Tahap 4

Tahap 5

Untuk soal nomor 1, memiliki batas waktu pengerjaan 1 jam (w) dan mendapatkan skor 25 (p)

$f_5(y) = \max\{ f_4(y) , p_5 + f_4(y-w_5) \}$
$f_5(y) = \max\{ f_4(y) , 25 + f_4(y-1) \}$

y		Solusi Optimum

	$f_4(y)$	$25 + f_4(y-1)$	$f_5(y)$	$(X_1^*,X_2^*,X_3^*,X_4^*,X_5^*)$
0	0	-∞	0	(0,0,0,0,0)
1	30	25	30	(0,0,1,0,0)
2	50	55	55	(1,0,1,0,0)
3	70	75	75	(1,0,1,0,1)

Tabel 1.11 Hasil Tahap 5

Pada persoalan ini, solusi persoalan terdapat pada tahap terakhir pada baris terakhir di kolom solusi optimum. Dari hasil diatas maka didapat vektor solusi optimum X = (1,0,1,0,1).

Dengan pendekatan program dinamis maju dan mundur, didapatkan bahwa agar peserta mendapat skor maksimum dalam waktu 3 jam adalah dengan mengerjakan soal 1 dengan batas waktu pengerjaan 1 jam dan mendapatkan skor 25, soal 3 dengan batas waktu pengerjaan 1 jam dan mendapatkan skor 30, dan soal 5 dengan batas waktu pengerjaan 1 jam dan mendapatkan skor 20. Peserta akan mendapatkan skor 75. Jika dibandingkan dengan algoritma *Greedy (*diluar topik pembahasan, didapatkan skor 70) tentu Program Dinamis lebih menghasilkan hasil yang optimum.

V. KESIMPULAN

Integer Knapsack Problem memiliki beberapa variasi yang berkaitan dengan kehidupan sehari-hari seperti dalam pengerjaan soal tes. Adapun, Program Dinamis dapat menghasilkan skor (profit) yang maksimal dengan konstrain waktu (bobot). Dari hasil percobaan, didapatkan juga pendekatan Program Dinamis Maju dan Program Dinamis Mundur menghasilkan keputusan yang sama.

VIDEO LINK YOUTUBE

https://youtu.be/baf7c1aSpr0

UCAPAN TERIMA KASIH

Pertama-tama penulis mengucapkan puji dan syukur ke hadirat Tuhan Yang Maha Esa karena rahmat-Nya dan pertolongan-Nya penulis dapat menyelesaikan tugas makalah IF2211 Strategi Algoritma ini, dan juga tugas-tugas lain pada mata kuliah ini.

Penulis juga mengucapkan terima kasih kepada dosen-dosen IF2211 Strategi Algoritma yaitu Bu Nur Ula Maulidevi, Bu Masayu Leylia Khodra dan Pak Rinaldi Munir (sebagai dosen K3). Terima kasih atas ilmu yang diajar dikelas selama ini, dan pertengahan akhir semester yang diharuskan kita kuliah dari rumah atas anjuran Pemerintah untuk dirumah saja #dirumahaja.

Penulis juga mengucapkan terima kasih kepada keluarga, terutama orangtua yang telah memberikan dukungan ekonomi dan semangat dalam kuliah selama ini.

Penulis juga mengucapkan terima kasih kepada teman-teman kelas K3 yang telah menemani selama satu semester ini dan mengerjakan Tugas Besar bersama-sama.

REFERENSI

[1] Munir, Rinaldi. *Diktat Kuliah IF2211 Strategi Algortima*. Program Studi Teknik Informatika ITB. 2020

[2] http://aren.cs.ui.ac.id/sda/resources/sda2010/05_rekursif.pdf (diakses pada 30 April 2020 pukul 16.00)

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.

Jakarta, 30 April 2020

Hengky Surya Angkasa