

# Aplikasi *Dynamic Programming* pada Penjadwalan Belajar berdasarkan Prediksi Waktu

Willy Wilsen - 13520160  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
13520160@std.stei.itb.ac.id

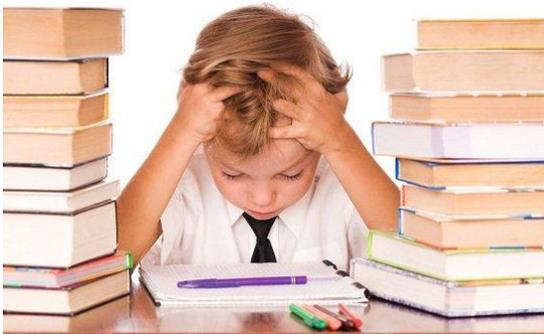
**Abstract**—Pada zaman baik dahulu maupun sekarang ini tentu tidak asing lagi dengan kata belajar. Salah satu hal yang paling memengaruhi proses belajar adalah waktu sehingga hal utama yang diperlukan untuk memperoleh hasil belajar yang optimal adalah dengan menentukan jadwal belajar. Penjadwalan belajar ini dapat diterapkan dengan menggunakan algoritma *Dynamic Programming* yang akan dibahas lebih lanjut dalam makalah ini.

**Keywords**—belajar, waktu, jadwal, *Dynamic Programming*

## I. PENDAHULUAN

Belajar merupakan salah satu proses yang sudah sangat sering dilalui oleh manusia. Belajar dapat dilakukan dengan banyak cara mulai dari membaca, mendengar, dan lain-lain. Seiring bertambah dewasa, manusia semakin ingin mencari tahu dan belajar banyak hal. Salah satu hal yang diinginkan dari belajar itu sendiri adalah hasil yang optimal.

Namun, terkadang hasil yang diinginkan tidak selalu sesuai seperti yang diharapkan. Manusia merupakan makhluk yang terbatas sehingga ada kalanya manusia harus dihadapi oleh kesulitan atau kegagalan dalam proses belajar. Salah satu hal utama yang dapat menyebabkan kesulitan atau kegagalan dalam proses belajar ini adalah keterbatasan waktu yang dimiliki manusia.



Gambar 1.1 Contoh kesulitan belajar

Sumber: [http://3.bp.blogspot.com/-M6vbmssNEnc/VjmN\\_QgSO3I/AAAAAAAAAHmk/ALFB\\_0GXlgU/s1600/cara-menentukan-kesulitan-belajar.jpg](http://3.bp.blogspot.com/-M6vbmssNEnc/VjmN_QgSO3I/AAAAAAAAAHmk/ALFB_0GXlgU/s1600/cara-menentukan-kesulitan-belajar.jpg)

Dengan adanya keterbatasan waktu tersebut, maka diperlukan solusi untuk mengoptimalkan hasil yang diperoleh dalam belajar. Salah satu solusinya adalah melakukan penjadwalan belajar. Oleh karena itu, akan dibahas algoritma dalam menemukan solusi tersebut, yakni *Dynamic Programming* dengan prediksi waktu beserta hasilnya.

## II. TEORI DASAR

### A. Algoritma *Dynamic Programming*

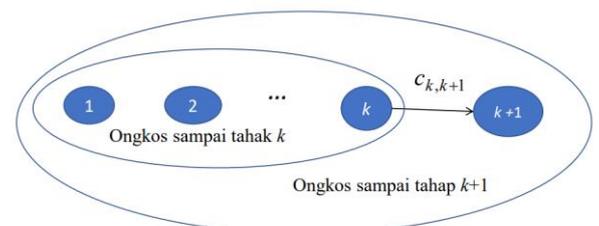
*Dynamic Programming* atau yang dikenal juga dengan program dinamis adalah metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan tahapan yang saling berkaitan. Program dinamis digunakan untuk menyelesaikan persoalan-persoalan optimasi (maksimasi atau minimasi).

Perbedaan algoritma Greedy dan program dinamis adalah pada algoritma Greedy hanya terdapat satu rangkaian keputusan yang dihasilkan sedangkan pada program dinamis terdapat lebih dari satu rangkaian keputusan yang dipertimbangkan.

#### 1. Prinsip optimalitas

Prinsip optimalitas adalah rangkaian keputusan yang optimal pada program dinamis. Prinsip optimalitas berbunyi “Jika solusi total optimal, maka bagian solusi sampai tahap ke- $k$  juga optimal”. Prinsip optimalitas berarti bahwa jika proses yang dilakukan dari tahap  $k$  ke tahap  $k+1$ , maka hasil optimal dari tahap  $k$  dapat digunakan tanpa harus kembali ke tahap awal.

$$\text{Cost}(k+1) = \text{Cost}(k) + \text{CostPindah}(k, k+1)$$

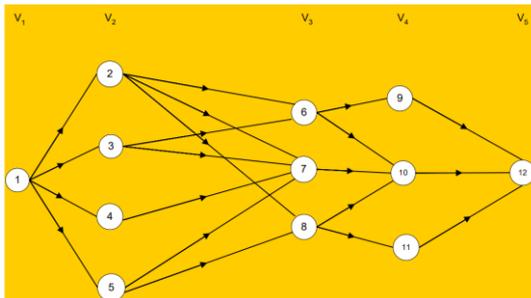


Gambar 2.a.1 Ilustrasi prinsip optimalitas

Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Program-Dinamis-2020-Bagian1.pdf>

2. Karakteristik persoalan dengan program dinamis
  - a. Persoalan dapat dibagi menjadi beberapa tahap yang pada setiap tahap hanya diambil satu keputusan.
  - b. Masing-masing tahap terdiri dari sejumlah status yang berhubungan dengan tahap tersebut. Secara umum, status merupakan bermacam kemungkinan masukan yang ada pada suatu tahap. Tahap ini dapat digambarkan sebagai graf multistage. Tiap simpul dalam graf ini menyatakan status dan  $V_1, V_2$ , hingga  $V_n$  menyatakan tahap.



Gambar 2.b.1 Ilustrasi graf multistage

Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Program-Dinamis-2020-Bagian1.pdf>

- c. Hasil dari keputusan yang diambil pada setiap tahap ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya.
    - d. Ongkos pada suatu tahap meningkat secara teratur dengan bertambahnya jumlah tahapan.
    - e. Ongkos pada suatu tahap bergantung pada ongkos tahap-tahap yang sudah berjalan dan ongkos dari tahap tersebut ke tahap berikutnya.
    - f. Adanya hubungan rekursif yang mengidentifikasi keputusan terbaik untuk setiap status pada tahap  $k$  memberikan keputusan terbaik untuk setiap status pada tahap  $k+1$ .
    - g. Prinsip optimalitas berlaku pada persoalan tersebut.
3. Dua pendekatan program dinamis
  - a. Program dinamis maju  
Program dinamis maju adalah perhitungan yang dilakukan dari tahap  $1, 2, \dots, n-1, n$

- b. Program dinamis mundur

Program dinamis mundur adalah perhitungan yang dilakukan dari tahap  $n, n-1, \dots, 2, 1$

4. Langkah-langkah pengembangan program dinamis
  - a. Karakteristikan struktur optimal berupa tahap, variable keputusan, status, dan sebagainya.
  - b. Definisikan secara rekursif nilai solusi optimal dan hubungkan nilai optimal suatu tahap dengan tahap sebelumnya.
  - c. Hitung nilai solusi optimal secara maju atau mundur menggunakan table.
  - d. Rekonstruksi solusi optimal (optional).

## B. Belajar

Menurut M. Sobry Sutikno, belajar adalah suatu proses yang dilakukan oleh seseorang untuk mendapatkan suatu perubahan yang baru sebagai hasil pengalamannya sendiri dalam interaksi dengan lingkungannya. Tujuan belajar secara umum adalah untuk memperoleh pengetahuan, menanamkan konsep dan keterampilan, dan membentuk sikap. Belajar dapat dibagi menjadi beberapa jenis:

1. Belajar rasional yaitu proses belajar menggunakan kemampuan berpikir sesuai dengan akal sehat (logis dan rasional) untuk memecahkan masalah.
2. Belajar abstrak yaitu proses belajar menggunakan berbagai cara berpikir abstrak untuk memecahkan masalah yang tidak nyata.
3. Belajar keterampilan yaitu proses belajar menggunakan kemampuan gerak motorik dengan otot dan urat syaraf untuk menguasai keterampilan jasmaniah tertentu.
4. Belajar sosial yaitu proses belajar memahami berbagai masalah dan cara penyelesaian masalah tersebut.
5. Belajar kebiasaan yaitu proses pembentukan atau perbaikan kebiasaan ke arah yang lebih baik agar individu memiliki sikap dan kebiasaan yang lebih positif sesuai dengan kebutuhan.
6. Belajar pemecahan masalah yaitu belajar berpikir sistematis, teratur, dan teliti atau menggunakan berbagai metode ilmiah dalam menyelesaikan suatu masalah.
7. Belajar apresiasi yaitu belajar kemampuan dalam mempertimbangkan arti atau nilai suatu objek sehingga individu dapat menghargai berbagai objek tertentu.
8. Belajar pengetahuan yaitu proses belajar berbagai pengetahuan baru secara terencana untuk menguasai materi pelajaran melalui kegiatan eksperimen dan investigasi.

### C. Jadwal

Menurut Glosarium, jadwal adalah pembagian waktu berdasarkan rencana pengaturan urutan kerja. Jadwal juga bisa diartikan sebagai daftar rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci.

### D. Waktu

Menurut KBBI, waktu adalah seluruh rangkaian ketika proses, perbuatan, atau keadaan berlangsung. Waktu juga dapat didefinisikan sebagai interval antara dua buah kejadian atau lama berlangsungnya suatu kejadian.

## III. APLIKASI DYNAMIC PROGRAMMING PADA PENJADWALAN BELAJAR BERDASARKAN PREDIKSI WAKTU

Permasalahan yang akan dibahas adalah penentuan penjadwalan belajar sehingga hasil yang diperoleh maksimal. Kasus yang diambil kali ini adalah pada mata kuliah program studi Teknik Informatika dengan prediksi waktu belajar dan nilai pada mata kuliah tersebut.

### A. Analisis Kasus

#### 1. Inisialisasi

Pada tahap inisialisasi, akan diminta input dari pengguna berupa jumlah mata kuliah yang ada dan batas maksimal prediksi untuk satu mata kuliah. Setelah itu, program akan membuat list matkul yang siap diisi oleh pengguna. Berikut merupakan potongan kodenya.

```
jumlah_matkul = int(input("Masukkan jumlah mata kuliah: "))
max_prediksi = int(input("Masukkan jumlah maksimal prediksi nilai untuk satu mata kuliah: "))
print()
matkul = [[0 for j in range(3)] for i in range(jumlah_matkul)]
for i in range(jumlah_matkul):
    matkul[i][1] = []
    for j in range(max_prediksi + 1):
        matkul[i][1].append((0, 0))
```

Gambar 3.a.1.1 Inisialisasi

#### 2. Input

Pada tahap input, akan diminta input dari pengguna untuk masing-masing nama mata kuliah, prediksi waktu belajar, dan prediksi nilai yang diperoleh. Prediksi dapat dilakukan berkali-kali pada mata kuliah tersebut hingga tidak melebihi batas maksimal prediksi untuk satu mata kuliah. Berikut merupakan potongan kodenya.

```
for i in range(jumlah_matkul):
    print("Masukkan nama mata kuliah ke-" + str(i+1) + end=" ")
    matkul[i][0] = input(": ")
    print()
    count_prediksi = 0
    while (count_prediksi <= max_prediksi):
        print("Masukkan prediksi jam belajar untuk mata kuliah", matkul[i][0], end=" ")
        matkul[i][1][count_prediksi + 1][0] = int(input(": "))
        print("Masukkan prediksi nilai untuk", matkul[i][1][count_prediksi + 1][0], "jam belajar mata kuliah", matkul[i][0], end=" ")
        matkul[i][1][count_prediksi + 1][1] = int(input(": "))
        count_prediksi += 1
    (count_prediksi := max_prediksi)
    input_prediksi = input("ingin memasukkan prediksi lagi? (y/n): ")
    print()
    matkul[i][1][2] = count_prediksi
    for j in range(count_prediksi, max_prediksi):
        matkul[i][1][j] = []
        matkul[i][1][j].append((0, 0))
```

Gambar 3.a.2.1 Input

### 3. Proses Dynamic Programming

Pada proses *Dynamic Programming* untuk kasus ini, tahap(k) adalah tahap mengalokasikan waktu untuk setiap mata kuliah. Status( $x_k$ ) menyatakan jumlah waktu yang dialokasikan pada setiap tahap. Alternatif(p) menyatakan prediksi waktu dan nilai yang diusulkan untuk setiap mata kuliah.

Misalkan,

$R_k(p_k)$  = keuntungan dari alternatif  $p_k$  pada tahap k

$f_k(x_k)$  = keuntungan optimal dari tahap 1, 2, ..., dan k yang diberikan oleh status  $x_k$

Maka,

$$f_1(x_1) = \max\{R_1(p_1)\} \text{ (basis)}$$

$$f_k(x_k) = \max\{R_k(p_k) + f_{k-1}(x_{k-1})\} \text{ (rekurens)}$$

Dengan catatan:

1.  $x_{k-1} = x_k - c_k(p_k)$

$c(p_k)$  adalah waktu untuk alternatif  $p_k$  pada tahap k.

2. Prediksi  $p_k$  dikatakan layak jika waktu  $c(p_k)$  tidak melebihi nilai status  $x_k$  pada tahap k.

Sehingga

$$f_1(x_1) = \max\{R_1(p_1)\} \text{ (basis)}$$

$$f_k(x_k) = \max\{R_k(p_k) + f_{k-1}[x_k - c_k(p_k)]\} \text{ (rekurens)}$$

Berikut merupakan potongan kodenya.

```
MAX_HOUR = int(input("Jam belajar maksimal yang Anda miliki: "))
tahap = [0 for i in range(jumlah_matkul)]
optimal = [0 for i in range(jumlah_matkul)]
for i in range(jumlah_matkul):
    tahap[i] = [0 for j in range(MAX_HOUR + 1)]
    optimal[i] = [0 for j in range(MAX_HOUR + 1)]
    for j in range(MAX_HOUR + 1):
        tahap[i][j] = [0 for k in range(matkul[i][2] + 1)]
        optimal[i][j] = [0 for k in range(2)]
        for k in range(matkul[i][2] + 1):
            if (j >= matkul[i][1][k][0]):
                tahap[i][j][k] = matkul[i][1][k][1] # basis
                if (i != 0):
                    tahap[i][j][k] += optimal[i - 1][j - matkul[i][1][k][0]][0] # rekurens
            else:
                tahap[i][j][k] = '-'
        optimal[i][j][0] = getOptimal(tahap[i][j])[0]
        optimal[i][j][1] = getOptimal(tahap[i][j])[1]
```

Gambar 3.a.3.1 Proses Dynamic Programming

### B. Uji Kasus

Pada tahap ini akan diberikan contoh persoalan dan visualisasi tiap Langkah hingga menuju solusi. Misalkan diberikan persoalan berikut berdasarkan hasil input.

```

Strategi Algoritma
  Prediksi 1:
    Jam belajar: 2
    Nilai: 80
  Prediksi 2:
    Jam belajar: 3
    Nilai: 100
Sistem Operasi
  Prediksi 1:
    Jam belajar: 1
    Nilai: 50
  Prediksi 2:
    Jam belajar: 2
    Nilai: 70
  Prediksi 3:
    Jam belajar: 4
    Nilai: 90
Probabilitas & Statistika
  Prediksi 1:
    Jam belajar: 1
    Nilai: 55
  Prediksi 2:
    Jam belajar: 3
    Nilai: 80

```

Gambar 3.b.1 Persoalan uji kasus

Prediksi	Strategi Algoritma		Sistem Operasi		Probabilitas & Statistika	
	Waktu	Nilai	Waktu	Nilai	Waktu	Nilai
1	0	0	0	0	0	0
2	2	80	1	50	1	55
3	3	100	2	70	3	80
4	-	-	4	90	-	-

Tabel 3.b.1 Visualisasi persoalan uji kasus

Prediksi waktu bernilai nol sengaja dicantumkan yang berarti tidak ada alokasi waktu untuk mata kuliah tersebut. Apabila waktu maksimal yang dialokasikan adalah 6 jam, maka berikut merupakan visualisasi setiap tahapnya.

1. Tahap 1

$$f_1(x_1) = \max\{R_1(p_1)\}$$

$x_1$	$R_1(p_1)$			Solusi Optimal	
	$p_1 = 1$	$p_1 = 2$	$p_1 = 3$	$f_1(x_1)$	$p_1^*$
0	<b>0</b>	-	-	0	1
1	<b>0</b>	-	-	0	1
2	0	<b>80</b>	-	80	2
3	0	80	<b>100</b>	100	3
4	0	80	<b>100</b>	100	3
5	0	80	<b>100</b>	100	3
6	0	80	<b>100</b>	100	3

Tabel 3.b.2 Visualisasi tahap 1

2. Tahap 2

$$f_2(x_2) = \max\{R_2(p_2) + f_1[x_2 - c_2(p_2)]\}$$

$x_2$	$R_2(p_2) + f_1[x_2 - c_2(p_2)]$				Solusi Optimal	
	$p_2=1$	$p_2=2$	$p_2=3$	$p_2=4$	$f_2(x_2)$	$p_2^*$
0	<b>0</b>	-	-	-	0	1
1	0	<b>50</b>	-	-	50	2
2	<b>80</b>	50	70	-	80	1
3	100	<b>130</b>	70	-	130	2
4	100	<b>150</b>	<b>150</b>	90	150	2, 3
5	100	150	<b>170</b>	90	170	3
6	100	150	<b>170</b>	<b>170</b>	170	3, 4

Tabel 3.b.3 Visualisasi tahap 2

3. Tahap 3

$$f_3(x_3) = \max\{R_3(p_3) + f_2[x_3 - c_3(p_3)]\}$$

$x_3$	$R_3(p_3) + f_2[x_3 - c_3(p_3)]$			Solusi Optimal	
	$p_3 = 1$	$p_3 = 2$	$p_3 = 3$	$f_3(x_3)$	$p_3^*$
0	<b>0</b>	-	-	0	1
1	50	<b>55</b>	-	55	2
2	80	<b>105</b>	-	105	2
3	130	<b>135</b>	80	135	2
4	150	<b>185</b>	130	185	2
5	170	<b>205</b>	160	205	2
6	190	<b>225</b>	210	225	2

Tabel 3.b.4 Visualisasi tahap 3

4. Solusi

Setelah seluruh tahap diproses, maka dilakukan rekonstruksi solusi.

$x_3$	$p_3^*$	$x_2$	$p_2^*$	$x_1$	$p_1^*$	$p_1^*, p_2^*, p_3^*$
6	2	5	3	3	3	3, 3, 2

Tabel 3.b.5 Rekonstruksi solusi

Maka, didapatkan solusi dari persoalan di atas untuk mencapai nilai optimal dengan maksimal waktu alokasi 6 jam adalah

1. Strategi Algoritma: 3 jam => 100
2. Sistem Operasi: 2 jam => 70
3. Probabilitas & Statistika: 1 jam => 55

Berikut merupakan hasil uji kasus dalam implementasi program.

```
Strategi Algoritma
  Prediksi 1:
    Jam belajar: 2
    Nilai: 80
  Prediksi 2:
    Jam belajar: 3
    Nilai: 100
Sistem Operasi
  Prediksi 1:
    Jam belajar: 1
    Nilai: 50
  Prediksi 2:
    Jam belajar: 2
    Nilai: 70
  Prediksi 3:
    Jam belajar: 4
    Nilai: 90
Probabilitas & Statistika
  Prediksi 1:
    Jam belajar: 1
    Nilai: 55
  Prediksi 2:
    Jam belajar: 3
    Nilai: 80
Jam belajar maksimal yang Anda miliki: 6
SOLUSI:
1. Strategi Algoritma: 3 jam belajar => 100
2. Sistem Operasi: 2 jam belajar => 70
3. Probabilitas & Statistika: 1 jam belajar => 55
```

Gambar 3.b.2 Solusi uji kasus

#### IV. KESIMPULAN

Keterbatasan waktu untuk belajar merupakan permasalahan umum yang dilalui manusia. Salah satu solusi yang dapat diterapkan untuk mengoptimalkan belajar adalah dengan menentukan jadwal belajar.

Berdasarkan analisis dan uji kasus yang telah dilakukan sebelumnya, dapat disimpulkan bahwa algoritma *Dynamic Programming* cukup efektif untuk memberikan hasil yang optimal dalam menentukan jadwal belajar. Diharapkan makalah ini dapat memotivasi pembaca yang ingin belajar dan mengembangkan algoritma yang ada untuk membuat solusi lain pada permasalahan yang lebih kompleks.

#### VIDEO LINK AT YOUTUBE

<https://youtu.be/ekHTU5yK6WE>

#### UCAPAN TERIMA KASIH

Saya mengucapkan syukur kepada Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan tugas makalah Strategi Algoritma yang berjudul “Aplikasi *Dynamic Programming* pada Penjadwalan Waktu berdasarkan Prediksi Waktu”. Ucapan terima kasih juga saya sampaikan kepada dosen pengampu mata kuliah IF2211 Strategi Algoritma, Bapak Ir. Rinaldi Munir, Ibu Dr. Nur Ulfa Maulidevi, dan secara khusus kepada Ibu Dr. Masayu Leylia Khodra selaku dosen Kelas 01 yang telah mengajar mata kuliah ini dan membuat saya dapat menyelesaikan tugas makalah ini dengan baik. Saya juga mengucapkan terima kasih kepada keluarga dan seluruh pihak yang senantiasa mendukung saya dalam pembuatan makalah ini.

#### REFERENCES

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Program-Dinamis-2020-Bagian1.pdf> (diakses pada 15 Mei, 13.00 WIB)
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Program-Dinamis-2020-Bagian2.pdf> (diakses pada 15 Mei, 15.00 WIB)
- [3] <https://www.maxmanroe.com/vid/umum/pengertian-belajar.html> (diakses pada 16 Mei, 10.00 WIB)
- [4] <https://glosarium.org/arti-jadwal/> (diakses pada 16 Mei 11.00 WIB)
- [5] <https://kbbi.web.id/waktu> (diakses pada 16 Mei 12.00 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, 20 Mei 2022

Willy Wilsen - 13520160