Penerapan Pemrograman Dinamis untuk Penentuan Jenis Ikan pada Budidaya Ikan Air Tawar

Doddy Aditya Wiranugraha - 13517008
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13517008@std.stei.itb.ac.id

***Abstrak* — Indonesia adalah negara dengan potensi alam yang cukup luas untuk berbagai bidang usaha, salah satunya adalah potensi budidaya ikan air tawar. Indonesia sangat potensial untuk mengembangkan budidaya ikan air tawar. Untuk mengurangi resiko gagal panen dan mengoptimalkan keuntungan, banyak petani yang kemudian memilih untuk budidaya ikan secara diversifikasi. Pembudidaya ikan memilih untuk membagi-bagi ukuran tambaknya dan kemudian membudidayakan jenis ikan yang berbeda. Hal ini dilakukan agar ketika suatu jenis ikan yang dibudidayakan gagal panen masih ada jenis ikan yang lain untuk menutupi kerugian dari ikan yang gagal panen. Namun ada batasannya adalah setiap jenis ikan memiliki biaya, nilai jual, dan waktu panen yang berbeda-beda. Oleh karena itu, dibutuhkan suatu cara untuk menentukan jenis ikan apa yang akan dibudidayakan sehingga dalam jangka waktu tertentu, pembudidaya ikan mendapatkan keuntungan yang didapatnya. Salah satu penyelesaian masalah ini adalah dengan memodelkan permasalahan ini sebagai *knapsack problem* dan menyelesaikannya dengan pemrograman dinamis.**

Kata Kunci* — *knapsack; pemrograman dinamis; budidaya; ikan; air tawar; pembudidaya

I. PENDAHULUAN

Gambar 1 – Tambak Ikan[1]

Indonesia dikenal sebagai negara terkaya dalam hal keanekaragaman hayati setelah negara Brazil. Di dalam bidang perikanan,, Indonesia adalah negara dengan keanekaragaman ikan terkaya pertama. Dengan jumlah penduduk yang sangat besar merupakan pasar yang sangat potensial bagi produk perikanan. Selain itu kekayaan perikanan menjadikan devisa negara melalui ekspor perikanan yang dilakukan oleh pembudidaya ikan.

Berbagai potensi perikanan di Indonesia ini didukung oleh kondisi geografis yang sangat strategis yaitu di titik silang perdaganan dunia. Namun pemerintah baru menangkap potensi ini pasca reformasi, setelah Presiden Abdurrahman Wahid membentuk Kementrian Kelautan dan Perikanan RI sebagai bagian dari Kabinet Persatuan Nasional.

Tidak heran jika potensi budidaya perikanan di Indonesia membuat banyak orang akhirnya memilih budidaya ikan sebagai mata pencahariannya sehari-hari. Oleh karena itu Indonesia seharusnya bisa mengoptimalkan potensi alam yang sangat besar ini. Namun ada berbagai macam kendala dalam mengembangkan potensi alam di Indonesia melalui budidaya ikan ini. Dalam hal budidaya ikan, para pembudidaya ikan harus berani mengambil resiko seperti gagal panen, penurunan kualitas perairan, besarnya porsi biaya, tingginya harga pakan, dan kondisi pasar yang berubah mempengaruhi nilai jual ikan.

Untuk mengurangi resiko tersebut, beberapa pembudidaya ikan telah menerapkan metode budidaya ikan terdiversifikasi, yaitu budidaya ikan dengan membudidayakan berbagai macam jenis ikan dimana satu jenis ikan pada satu tambak. Hal ini dilakukan agar meminimalkan resiko kerugian yang ditanggung oleh pembudidaya ikan. Metode diversifikasi ini dapat memberikan jaminan bahwa usaha budidaya ikan yang dilakukan akan berkelanjutan dan perkembangan usaha akan efektif dan efisien.

Namun dengan menerapkan metode diversifikasi dapat timbul sebuah permasalahan baru bagi pembudidaya ikan. Banyak faktor yang harus dipertimbangkan mengenai jenis ikan apa saja yang harus dibudidayakan dan harus dipertimbangkan dengan matang karena akan mempengaruhi keuntungan dan keberhasilan dari metode ini. Jenis perikanan yang akan dibahas pada makalah ini adalah ikan air tawar. Dalam hal ini ikan air tawar terdiri dari berbagai macam dan jenis ikan, oleh karena itu dalam mempertimbangkan pemilihan jenis ikan diberikan batasan bahwa setiap jenis ikan memiliki harga modal, harga jual, dan waktu panen. Ketiga hal inilah yang akan menjadi faktor yang mempengaruhi keuntungan

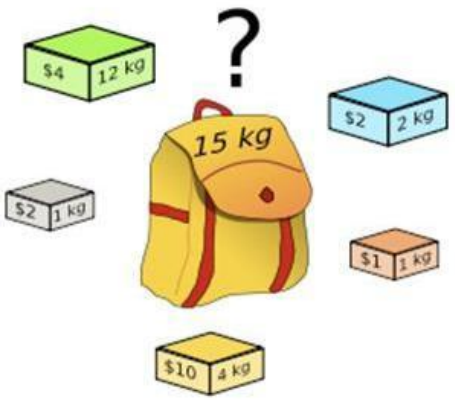
pembudidaya ikan air tawar. Berdasarkan hal ini, pembudidaya dapat menentukan ikan apa yang sebaiknya dipilih untuk kemudian dibudidayakan di tambak.

Makalah ini akan mengaplikasian pemrograman dinamis (*dynamic programming)* yang merupakan salah satu topik mata kuliah "Strategi Algoritma" untuk menyelesaikan permasalahan penentuan ikan apa saja yang sebaiknya dibudidayakan oleh seorang pembudidaya ikan dengan metode diversifikasi. Permasalahan ini dapat dimodelkan seperti permasalahan *integer (1/0) knapsack problem* yang memiliki batasan (*constraint)* yaitu harga modal, harga jual, dan waktu panen. Dengan pemrograman dinamis ini diharapkan pembudidaya ikan dapat dengan mudah menentukan ikan apa saja yang dapat memberikan keuntungan sebanyak mungkin dari beberapa pilihan ikan yang ada untuk dibudidayakan di tambaknya.

II. DASAR TEORI

A. *Integer (1/0) Knapsack Problem*[2]

Integer (1/0) knapsack problem adalah permasalahan memilih n buah objek untuk dimasukkan ke dalam sebuah karung (*knapsack*) yang memiliki kapasitas bobot *K* sehingga objek-objek yang dipilih tidak akan melebihi kapasitas *knapsack*. Tujuan yang dicapai dari *knapsack problem* ini adalah keuntungan yang diperoleh maksimal. Setiap objek yang akan dipilih memiliki bobot (*weight*) *w* tertentu dan keuntungan atau nilai (*value*) *v* tertentu. Jadi pada *Integer (1/0) knapsack problem*, pada setiap tahap, barang yang diambil harus memperhatikan kapasitas *knapsack* yang dimiliki dan keuntungan yang diperoleh harus maksimal. Berikut adalah bentuk matematis dari persoalan *Integer (1/0) knapsack problem.*



Gambar 2 – *Knapsack Problem*[2]

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

Integer (1/0) knapsack problem dapat digunakan untuk memodelkan berbagai persoalan optimasi yang solusinya berupa *subset* yang memiliki batasan (*constraint*) tertentu. Contoh dari permasalahan yang dapat dimodelkan dengan persoalan *knapsack problem* adalah penyimpanan barang di gudang dan pengaturan pengiriman barang dengan efisien untuk memperoleh keuntungan maksimal.

Dalam teori komputasi, *Integer (1/0) knapsack problem* adalah suatu persoalan yang termasuk ke dalam persoalan NP-Complete. Hal ini berarti bahwa belum ditemukan suatu algoritma yang dapat menyelesaikan persoalan *knapsack problem* dengan waktu polinomial. Meskipun belum ditemukan algoritma yang mangkus untuk menyelesaikan persoalan ini dalam, setidaknya terdapat algoritma lain yang dapat menyelesaikan persoalan ini dengan waktu pseudo-polinomial yaitu pemrograman dinamis (*dynamic programming)*. Selain dengan menggunakan pemrograman dinamis (*dynamic programming*), persoalan semacam ini juga dapat diselesaikan dengan strategi algoritma yang lain seperti *branch and bound* , *exhaustive search* dengan cara mengenumerasikan semua kemungkinan solusi dan memilih yang optimum, dan *greedy* dengan cara mengambil solusi optimum lokal dengan harapan mendapatkan solusi optimum global.

Integer (1/0) knapsack problem memiliki beberapa variasi. Variasi ini timbul dengan mengubah parameter permasalahan yang semula seperti jumlah *knapsack* yang digunakan, jumlah batasan (*constraint*), atau jumlah tujuan. Beberapa variasi dari *knapsack problem* diantaranya adalah *bounded knapsack problem* dimana setiap objek dapat dipilih sebanyak c, *unbounded knapsack problem* dimana tidak ada batasan untuk pemilihan setiap objek, *multi-objective knapsack problem* dimana tujuan atau hal yang ingin dioptimasi lebih dari satu, *multi-dimensional knapsack problem* dimana batasan (*constraint*) lebih dari satu, dan *multiple knapsack problem* dimana jumlah *knapsack* lebih dari satu.

B. *Program Dinamis (Dynamic Programming)*[3]

Pemrograman Dinamis (*dynamic programming*) merupakan salah satu strategi pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan tahapan (*stage*) sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berakitan. Istilah pemrograman dinamis ini 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 yang isinya terisi secara dinamis. Pemrograman dinamis umumnya digunakan untuk menyelesaikan persoalan optimasi.

Pemrograman dinamis (*dynamic programming*) ini sebenarnya mirip dengan strategi algoritma lain yaitu *greedy* dimana solusi yang dibentuk secara bertahap dan mengambil langkah terbaik untuk setiap langkahnya. Perbedaan kedua strategi ini terletak pada rangkaian keputusan yang dipertimbangkan dalam penyelesaian masalah. Pada metode

greedy, hanya satu rangkaian keputusan saja yang dipertimbangkan, yaitu rangkaian keputusan yang dapat memberikan hasil terbaik pada setiap tahapnya (optimum lokal) dengan harapan dapat mengarah ke solusi yang optimum global. Pada metode *greedy*, keputusan yang dipilih pada tahap tertentu tidak mempertimbangkan rangkaian keputusan untuk tahapan yang telah dipilih sebelumnya dan tahapan yang akan dipilih berikutnya. Contoh masalah yang tidak selalu menghasilkan solusi optimum global adalah permasalahan *Integer (1/0) knapsack problem.*

Berbeda dengan metode *greedy*, metode pemrograman dinamis mempertimbangkan lebih dari satu rangkaian keputusan, dimana rangkaian-rangkaian keputusan tersebut dibangun 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 yang telah dijelaskan, maka dapat dirumuskan bahwa ongkos pada tahap *k* + 1 adalah sama dengan (ongkos yang dihasilkan pada tahap *k*) ditambah dengan (ongkos dari tahap *k* ke tahap *k* + 1).

Pemrograman dinamis (*dynamic programming*) dapat diaplikasikan pada persoalan yang memiliki karakteristik sebagai berikut[3] :

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 mengidentifikasikan 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 digunakan untuk penyelesaian suatu permasalah dengan pemrograman dinamis (*dynamic programming*), yaitu sebagai berikut[3]:

1. Pemrograman dinamis maju (*forward* atau *up-down*). Pemrograman dinamis bergerak mulai dari tahap 1, lalu ke tahap 2, 3, dan seterusnya sampai tahap ke-*n*. Rangkaian peubah keputusannya adalah $x_1, x_2, \ldots, x_n$.
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 peubah keputusannya adalah $x_n, x_{n-1}, \ldots, x_1$.

Kedua pendekatan ini menghasilkan solusi yang sama dan keduanya merupakan solusi yang optimum.

Pada umumnya, untuk mengembangkan sebuah algoritma pemecahan masalah melalui strategi pemrograman dinamis (*dynamic programming*) dilakukan dengan langkah berikut[3]:

1. Karakteristikkan 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 untuk melakukan penyimpanan hasil kalkulasi dengan nilai optimum pada setiap tahap. Implementasi secara rekursif akan membangun solusi dari atas ke bawah (*top-down*) dan membentuk sebuah pohon dan hanya akan mengunjungi status yang dibutuhkan pada setiap tahapnya untuk mencapai solusi optimal tanpa menelusuri semua status yang mungkin pada setiap tahapnya. Implementasi secara iteratif akan membangun solusi dari bawah ke atas (*bottom-up*) dan menelusuri semua status yang mungkin untuk setiap tahap.

III. PENYELESAIAN MULTI-DIMENSIONAL KNAPSACK PROBLEM DENGAN MENGGUNAKAN PEMROGRAMAN DINAMIS

Penyelesaian *multi-dimensional knapsack problem* dengan menggunakan pemrograman dinamis akan dilakukan dengan cara mengikuti empat langkah yang telah dijelaskan sebelumnya. Pada persoalan ini, tahap *k* adalah proses memasukkan objek ke-*k* ke dalam *knapsack* dan status *y* menyatakan kapasitas muat *knapsack* yang tersisa setelah memasukkan barang pada *knapsack* ditahap sebelumnya. Kemudian kapasitas *knapsack* (M) untuk permasalahan seperti ini dapat dimisalkan dalam bentuk vektor multidimensi tergantung banyaknya batasan (*constraint*) yang teridentifikasi. Misal terdapat *n* batasan (*constraint*) maka akan ada $y = (y_1, y_2, \ldots, y_n)$. Jadi maksudnya adalah jika ada *n* batasan maka akan ada juga sebanyak *y*n* yang menyatakan ada sebanyak *n* kapasitas batasan.

Solusi optimal untuk persoalan semacam ini dapat dibentuk melalui proses seperti berikut. Misalkan ketika berada pada tahap *k*, maka akan ada dua opsi untuk mempertimbangkan apakah objek tersebut akan dimasukkan pada *knapsack* atau tidak. Perhitungan yang dilakukan pada saat memasukkan objek ke dalam *knapsack* adalah dengan menggunakan rumus $y - w_k$, dimana w_k merupakan bobot untuk setiap batasan. Pada

setiap tahap ketika akan mengisi kapasitas yang tersisa pada *knapsack* maka dilakukan dengan prinsip optimalitas dengan mengacu pada nilai optimal pada tahap sebelumnya, yaitu $f_{k-1}(y - w_k)$. Kemudian nilai keuntungan dari pemilihan objek pada tahap ke-k (p_k) ditambahkan dengan nilai $f_{k-1}(y - w_k)$ dibandingkan dengan keuntungan tanpa melakukan pemilihan objek pada tahap ke-k yaitu $f_{k-1}(y)$ sehingga akan dipilih solusi yang menghasilkan keuntungan terbesar, apakah mengambil objek ke-k tersebut atau tidak.

Persoalan ini dapat dijelaskan secara ringkas dengan relasi rekurens sebagai berikut.

$f_0(y) = 0, y_i = 0, 1, 2, \ldots, M_i$ *(basis)*
untuk $1 \le i \le m$

$f_k(y) = -\infty$, jika terdapat $y_i < 0$ *(basis)*
untuk $1 \le i \le m$

$f_k(y) = \max \{ f_{k-1}(y), p_k + f_{k-1}(y - w_k)\}$, (*rekurens*)

$k = 1, 2, \ldots, n$ dan $y_i \ge 0$ untuk $1 \le i \le m$

$f_k(y)$ adalah keuntungan optimal dari persoalan *multi-dimensional knappsack problem* pada tahap ke-k untuk kapasitas y. Untuk nilai yang $f_k(y) = -\infty$ merupakan nilai untuk kapasitas yang bernilai negatif, sedangkan nilai $f_0(y) = 0$ merupakan nilai yang mendefinisakan bahwa tidak ada objek yang akan dipilih, atau merupakan inisialisasi awal. $fn(M)$ merupakan solusi optimum dari persoalan *multi-dimensional knapsack problem*. Kompleksitas algoritma ini adalah $O(nM)$.

Berikut ini akan diberikan contoh mengenai penerapan pemrograman dinamis untuk menyelesaikan persoalan *multi-dimensional knapsack problem*. Dimisalkan terdapat dua batasan yaitu kapasitas bobot dan volume maksimal yang dapat ditampung oleh *knapsack* sehingga m=2 dengan bobot dan volume maksimal masing-masing adalah 3 dan 2 serta jumlah objek yang akan dimasukkan kedalam *knapsack* ada 3 buah dengan rincian sebagai berikut.

Tabel I Rincian objek

Objek ke-i	bobot (w_i)	volume (v_i)	nilai (p_i)
1	1	1	55
2	3	2	70
3	2	2	65

Berikut adalah perhitungan pada setiap tahapnya.

1. Tahap 1:

$f_1(y_1, y_2) = \max \{f_0(y_1, y_2), p_1 + f_0(y_1 - w_1, y_2 - v_1)\}$
$= \max \{f_0(y_1, y_2), 55 + f_0(y_1 - 1, y_2 - 1)\}$

Tabel II Tahapan I

y_1	y_2	$f_0(y1, y2)$	$55 + f_0(y_1 - 1, y_2 - 1)$	$f_1(y1, y2)$	(x_1^*, x_2^*, x_3^*)
0	0	0	-∞	0	(0, 0, 0)
0	1	0	-∞	0	(0, 0, 0)
0	2	0	-∞	0	(0, 0, 0)
1	0	0	-∞	0	(0, 0, 0)
1	1	0	55	55	(1, 0, 0)
1	2	0	55	55	(1, 0, 0)
2	0	0	-∞	0	(0, 0, 0)
2	1	0	55	55	(1, 0, 0)
2	2	0	55	55	(1, 0, 0)
3	0	0	-∞	0	(0, 0, 0)
3	1	0	55	55	(1, 0, 0)
3	2	0	55	55	(1, 0, 0)

2. Tahap 2:

$f_2(y_1, y_2) = \max \{f_1(y_1, y_2), p_2 + f_1(y_1 - w_2, y_2 - v_2)\}$
$= \max \{f_1(y_1, y_2), 70 + f_1(y_1 - 3, y_2 - 2)\}$

Tabel III Tahapan 2

y_1	y_2	$f_1(y1, y2)$	$70 + f_1(y_1 - 3, y_2 - 2)$	$f_2(y1, y2)$	(x_1^*, x_2^*, x_3^*)
0	0	0	70 - ∞ = -∞	0	(0, 0, 0)
0	1	0	70 - ∞ = -∞	0	(0, 0, 0)
0	2	0	70 - ∞ = -∞	0	(0, 0, 0)
1	0	0	70 - ∞ = -∞	0	(0, 0, 0)
1	1	55	70 - ∞ = -∞	55	(1, 0, 0)
1	2	55	70 - ∞ = -∞	55	(1, 0, 0)
2	0	0	70 - ∞ = -∞	0	(0, 0, 0)
2	1	55	70 - ∞ = -∞	55	(1, 0, 0)
2	2	55	70 - ∞ = -∞	55	(1, 0, 0)
3	0	0	70 - ∞ = -∞	0	(0, 0, 0)
3	1	55	70 - ∞ = -∞	55	(1, 0, 0)
3	2	55	70 + 0 = 70	70	(0, 1, 0)

3. Tahap 3:

$f_3(y_1, y_2) = \max \{f_2(y_1, y_2), p_3 + f_2(y_1 - w_3, y_2 - v_3)\}$
$= \max \{f_2(y_1, y_2), 65 + f_2(y_1 - 2, y_2 - 2)\}$

Tabel IV Tahapan 3

y_1	y_2	$f2(y1, y2)$	$65 + f_2(y_1 - 2, y_2 - 2)$	$f_3(y1, y2)$	(x_1^*, x_2^*, x_3^*)
0	0	0	65 - ∞ = -∞	0	(0, 0, 0)
0	1	0	65 - ∞ = -∞	0	(0, 0, 0)
0	2	0	65 - ∞ = -∞	0	(0, 0, 0)
1	0	0	65 - ∞ = -∞	0	(0, 0, 0)
1	1	55	65 - ∞ = -∞	55	(1, 0, 0)
1	2	55	65 - ∞ = -∞	55	(1, 0, 0)
2	0	0	65 - ∞ = -∞	0	(0, 0, 0)
2	1	55	65 - ∞ = -∞	55	(1, 0, 0)
2	2	55	65 + 0 = 65	65	(0, 0, 1)
3	0	0	65 - ∞ = -∞	0	(0, 0, 0)
3	1	55	65 - ∞ = -∞	55	(1, 0, 0)
3	2	70	65 + 0 = 65	70	(0, 1, 0)

Dari tahapan terakhir ditemukan solusi optimum dari persoalan ini yaitu $X = (0, 1, 0)$ dengan $\sum p = f_3(3, 2) = 70$.

IV. PENERAPAN PEMROGRAMAN DINAMIS UNTUK MENENTUKAN JENIS IKAN PADA BUDIDAYA IKAN AIR TAWAR

Dapat dimisalkan suatu pasar pada tahun 2019 dengan fokus pada sektor perikanan memiliki keadaan pasar sebagai berikut.

Ikan	Modal (Juta Rp)	Harga Jual (Juta Rp)	Waktu Panen (Bulan)
Ikan Mas	5	11	4
Ikan Lele	2	7	3
Ikan Patin	3	20	5
Ikan Nila	8	17	6
Ikan Gurame	9	32	10
Ikan Gabus	5	15	6
Ikan Bandeng	4	16	5

Dengan data tersebut maka pembudidaya ikan air tawar harus dapat menentukan jenis ikan yang akan dipilih untuk dibudidayakan pada tahun tersebut. Dari serangkaian data tersebut tujuan yang akan dicapai adalah mengoptimalkan keuntungan yang dihasilkan dari budidaya ikan yang telah dipilih. Dengan persoalan *multi-dimensional knapsack problem* yang telah dijelaskan sebelumnya, maka persoalan ini juga dapat diselesaikan menggunakan metode tersebut. Pada kasus ini yang akan menjadi batasan *y* adalah modal dan waktu panen. Terdapat 7 jenis ikan yang akan dipilih sehingga akan ada 7 buah tahapan untuk memperoleh solusi optimum berupa keuntungan.

Untuk penerapan pemrograman dinamis pada persoalan ini akan mirip seperti yang telah dilakukan pada bagian III. Pada program yang telah saya buat, pengguna akan memasukkan batasan-batasan pada program, dengan jumlah batasan maksimal adalah dua buah, kemudian program akan menampilkan ikan apa saja yang akan dipilih beserta jumlah keuntungan yang diperoleh, yang dimana jumlah keuntungan tersebut adalah keuntungan optimal.

Pada program yang telah dibuat oleh penulis, akan dilakukan beberapa uji kasus yang akan menyelesaikan permasalahan diatas. Berikut beberapa uji kasus dan hasil analisisnya.

1. Uji kasus 1

   Batasan 1 : Modal ≤ 20(Juta)

   Batasan 2 : Waktu panen ≤ 8(Bulan)

```
C:\Users\Doddy Aditya\Desktop>python tes.py
Batasan 1 : Modal 20 Juta
Batasan 2 : Waktu panen 8 Bulan
Keuntungan maksimum :  27
[0, 1, 1, 0, 0, 0, 0]
Pilihan jenis ikan yang menghasilkan keuntungan maksimum :
Ikan Lele
Ikan Patin
```

Gambar 3. Hasil uji kasus pertama

Pada hasil uji kasus pertama didapatkan nilai keuntungan sebesar 27 juta dimana dengan batasan pertama yaitu modal sebesar 20 juta dan batasan kedua yaitu waktu panen sebesar 8 bulan. Nilai 0 menandakan bahwa ikan tersebut tidak dipilih, sedangkan nilai 1 menandakan bahwa ikan tersebut dipilih. Dari hasil uji kasus tersebut didapatkan jenis ikan yang memberikan nilai keuntungan maksimum adalah ikan lele dan ikan patin.

2. Uji kasus 2

   Batasan 1 : Modal ≤ 25(Juta)

   Batasan 2 : Waktu panen ≤ 10(Bulan)

```
C:\Users\Doddy Aditya\Desktop>python tes.py
Batasan 1 : Modal 25 Juta
Batasan 2 : Waktu panen 10 Bulan
Keuntungan maksimum :  36
[0, 0, 1, 0, 0, 0, 1]
Pilihan jenis ikan yang menghasilkan keuntungan maksimum :
Ikan Patin
Ikan Bandeng
```

Gambar 4. Hasil uji kasus kedua

Pada hasil uji kasus kedua didapatkan nilai keuntungan sebesar 36 juta dimana dengan batasan pertama yaitu modal sebesar 25 juta dan batasan kedua yaitu waktu panen sebesar 10 bulan. Nilai 0 menandakan bahwa ikan tersebut tidak dipilih, sedangkan nilai 1 menandakan bahwa ikan tersebut dipilih. Dari hasil uji kasus tersebut didapatkan jenis ikan yang memberikan nilai keuntungan maksimum adalah ikan patin dan ikan bandeng.

3. Uji kasus 3

   Batasan 1 : Modal ≤ 25(Juta)

   Batasan 2 : Waktu panen ≤ 5(Bulan)

```
C:\Users\Doddy Aditya\Desktop>python tes.py
Batasan 1 : Modal 25 Juta
Batasan 2 : Waktu panen 5 Bulan
Keuntungan maksimum :  20
[0, 0, 1, 0, 0, 0, 0]
Pilihan jenis ikan yang menghasilkan keuntungan maksimum :
Ikan Patin
```

Gambar 5. Hasil uji kasus ketiga

Pada hasil uji kasus ketiga didapatkan nilai keuntungan sebesar 20 juta dimana dengan batasan pertama yaitu modal sebesar 25 juta dan batasan kedua yaitu waktu panen sebesar 5 bulan. Nilai 0 menandakan bahwa ikan tersebut tidak dipilih, sedangkan nilai 1 menandakan bahwa ikan tersebut dipilih. Dari hasil uji kasus tersebut didapatkan jenis ikan yang memberikan nilai keuntungan maksimum adalah ikan patin.

Dari hasil pengujian yang telah dilakukan terhadap 3 kasus diatas dapat dilihat bahwa jenis ikan yang didapatkan dengan memodelkan persoalan ini kedalam *knapsack problem* dan menggunakan pemrograman dinamis (*dynamic programming*) merupakan hasil yang paling maksimal dari semua kemungkinan solusi yang ada.

Pada persoalan ini hanya sebatas menentukan jenis ikan dengan faktor yang dipertimbangkan adalah modal, keuntungan, dan lama waktu panen budidaya tersebut. Hal inilah yang membuat program ini kurang dapat diimplementasikan dalam dunia nyata. Permasalahan lain yang dapat ditemui jika persoalan ini diselesaikan dengan pemrograman dinamis adalah batasan memori yang dapat digunakan oleh program karena jika jumlah batasan semakin bertambah maka jumlah kemungkinan kombinasi akan juga bertambah. Misalnya kita ambil kasus uji 1 yaitu batasan pertama bernilai 20 dan batasan kedua bernilai 8, jika dilakukan penyelesaian dengan pemrograman dinamis maka akan ada 20*8 kombinasi, belum lagi berapa banyak jumlah objek yang ada, oleh karena itu program yang dibuat ini hanya dibatasi oleh sebanyak dua buah batasan dan tujuh buah objek.

V. KESIMPULAN

Persoalan *integer (1/0) knapsack* yang dapat diselesaikan dengan pemrograman dinamis merupakan metode yang dapat digunakan untuk menyelesaikan persoalan yang dimana memiliki hubungan dengan pemilihan objek-objek dan ada batasan-batasan yang harus diperhatikan dalam pemilihan objek tersebut, sehingga tujuan akhir dari metode ini adalah untuk mengoptimasi apa yang diinginkan pada persoalan tersebut. Contoh dari penerapan metode ini adalah untuk memilih jenis ikan pada budidaya ikan air tawar, dimana batasan-batasannya adalah modal dan waktu panen. Tujuan dari pemilihan jenis ikan tersebut adalah mengoptimalkan keuntungan yang didapat oleh pembudidaya. Dari hasil uji kasus yang telah dibahas pada bagian IV, hasil yang dicapai pada setiap uji kasus adalah hasil yang optimal. Pada program yang telah dibuat, mungkin belum sepenuhnya dapat diimplementasikan di dunia nyata, karena ada beberapa faktor lain yang belum dianalisis pada program yang telah dibuat, oleh karena itu untuk harapan kedepannya, semoga program yang telah dibuat ini dapat dikembangkan lebih lanjut sehingga lebih bermanfaat bagi orang lain.

VI. UCAPAN TERIMA KASIH

Penulis ingin mengucapkan puji syukur kepada Tuhan Yang Maha Esa karena dengan rahmat dan karunia-Nya, penulis dapat menyelesaikan makalah dengan judul "Penerapan Pemrograman Dinamis untuk Penentuan Jenis Ikan pada Budidaya Ikan Air Tawar" ini tepat waktu. Penulis ingin berterima kasih kepada kedua orang tua penulis, yaitu Ayah dan Ibu yang selama ini telah mendukung saya dalam keberjalan pengerjaan tugas makalah ini. Penulis juga ingin berterima kasih kepada rekan-rekan yang telah memberikan semangat kepada penulis, sehingga penulis dapat memiliki motivasi untuk menyelesaikan makalah ini.

REFERENSI

[1] http://www.bibitikan.net/menggali-potensi-budidaya-ikan-air-tawar-di-indonesia/ [diakses 21 April 2019 pukul 11.00 WIB].

[2] R. Munir, Algoritma Brute Force, 2016. [Online] Tersedia dalam http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Brute-Force-(2016).pdf. [diakses 21 April 2019 pukul 12.00 WIB].

[3] 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. [diakses 21 April 2019 pukul 12.00 WIB].

[4] https://news.kkp.go.id/index.php/potensi-usaha-budidaya-ikan-air-tawar/ [diakses 22 April 2019 pukul 18.00 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, 24 April 2019

Doddy Aditya Wiranugraha
13517008