

Optimalisasi Penyelesaian Knapsack Problem Dengan Algoritma Genetika

I Wayan Supriana^{a1}

^aJurusan Ilmu Komputer, Fakultas MIPA, Universitas Udayana, Indonesia
Jalan Kampus Bukit Jimbaran, Bali, Indonesia
¹iwayansupriana@gmail.com

Abstrak

Permasalahan knapsack merupakan permasalahan yang sering kita temukan dalam kehidupan sehari-hari. Knapsack problem sendiri adalah sebuah permasalahan dimana seseorang dihadapkan pada permasalahan optimasi pada pemilihan benda yang dapat dimasukkan ke dalam wadah yang memiliki keterbatasan ruang atau daya tampung. Permasalahan knapsack problem dapat diselesaikan dengan berbagai algoritma optimasi, salah satunya menggunakan algoritma genetika. Algoritma genetika dalam penyelesaian masalah meniru teori evolusi makhluk hidup. Adapun komponen-komponen algoritma genetika tersusun dari populasi yang terdiri dari kumpulan individu-individu yang merupakan calon solusi dari permasalahan knapsack. Proses berjalannya evolusi dimulai dari proses seleksi, pindah silang dan mutasi pada setiap individu sehingga diperoleh populasi baru. Proses evolusi akan dilakukan secara berulang sampai memenuhi kriteria optimal dari solusi yang dihasilkan. Permasalahan yang ditekankan pada penelitian ini adalah bagaimana cara menyelesaikan permasalahan knapsack dengan menerapkan algoritma genetika. Hasil yang diperoleh berdasarkan pengujian dari sistem yang dibangun, bahwa knapsack problem dapat mengoptimalkan penempatan barang-barang dalam wadah atau daya tampung yang tersedia. Optimalisasi permasalahan knapsack dapat dimaksimalkan dengan inputan parameter yang sesuai.

Kata Kunci : Knapsack Problem, Algoritma Genetika, Optimalisasi, Populasi

Abstract

Knapsack problems is a problem that often we encounter in everyday life. Knapsack problem itself is a problem where a person faced with the problems of optimization on the selection of objects that can be inserted into the container which has limited space or capacity. Problems knapsack problem can be solved by various optimization algorithms, one of which uses a genetic algorithm. Genetic algorithms in solving problems mimicking the theory of evolution of living creatures. The components of the genetic algorithm is composed of a population consisting of a collection of individuals who are candidates for the solution of problems knapsack. The process of evolution goes through the selection process, crossovers and mutations in each individual in order to obtain a new population. The evolutionary process will be repeated until it meets the criteria of an optimum of the resulting solution. The problems highlighted in this research is how to resolve the problem by applying a genetic algorithm knapsack. The results obtained by the testing of the system is built, that the knapsack problem can optimize the placement of goods in containers or capacity available. Optimizing the knapsack problem can be maximized with the appropriate input parameters.

Keywords : Knapsack Problem, Genetic Algorithm, Optimization, Population

1. Pendahuluan

Knapsack problem merupakan permasalahan yang sering kita temukan dalam kehidupan sehari-hari. Knapsack problem sendiri adalah sebuah permasalahan dimana seseorang dihadapkan pada permasalahan optimasi pada pemilihan benda yang dapat dimasukkan ke dalam wadah yang memiliki keterbatasan ruang atau daya tampung. Contoh permasalahan knapsack jika seorang pengepak barang di gudang harus menempatkan berbagai jenis barang

ke dalam wadah atau tempat yang memiliki kapasitas maksimum sehingga tidak memungkinkan menempatkan semua barang, oleh sebab itu bagaimana petugas harus memasukkan barang semaksimal mungkin ke dalam wadah atau tempat barang tersebut ketika menerima permintaan pengiriman ke konsumen. Jenis barang yang dimasukkan kedalam wadah memiliki berat, harga serta tingkat kepentingan dari barang lain. Petugas gudang akan memilih barang yang sesuai dengan tempat atau wadah dengan pertimbangan berat barang tidak melebihi kapasitas maksimum sehingga dapat mengoptimalkan tempat yang digunakan. Melalui proses memaksimalkan barang yang di tempatkan di wadah diharapkan pengiriman jumlah barang dapat dioptimalkan sehingga mendapatkan keuntungan yang sebesar-besarnya.[1]

Permasalahan knapsack problem dapat diselesaikan dengan berbagai algoritma optimasi, salah satunya menggunakan algoritma genetika. Berbagai studi pembahasan knapsack problem sudah pernah dilakukan, seperti penyelesaian knapsack problem dengan menggunakan algoritma genetika oleh Kartina Diah KW, Mardhiah Fadhli dan Carly Sutanto Jurusan Teknik Komputer Politeknik Caltex Riau Pekanbaru. Dalam pembahasannya bahwa knapsack problem tergantung dari penentuan parameter, data yang diinputkan serta besarnya kapasitas atau daya tampung dari wadah. Penelitian berikutnya dari Komang Setemen Jurusan Manajemen Informatika Fakultas Teknik dan Kejuruan Universitas Pendidikan Ganesha. Hasil yang diperoleh algoritma genetika mampu memberikan solusi optimal sesuai dengan yang diharapkan. [2]

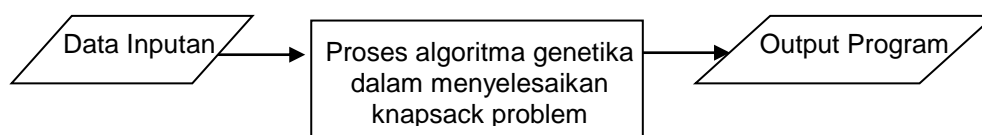
Algoritma genetika dalam penyelesaian masalah meniru teori evolusi mahluk hidup. Adapun komponen-komponen algoritma genetika tersusun dari populasi yang terdiri dari kumpulan individu-individu yang merupakan calon solusi dari permasalahan knapsack. Proses berjalannya evolusi dimulasi dari proses seleksi, pindah silang dan mutasi pada setiap individu sehingga diperoleh populasi baru. Proses evolusi akan dilakukan secara berulang sampai memenuhi kriteria optimal dari solusi yang dihasilkan. Permasalahan yang ditekankan pada penelitian ini adalah bagaimana cara menyelesaikan permasalahan knapsack dengan menerapkan algoritma genetika.

2. Metodologi Penelitian

Knapsack problem dalam penelitian ini menekankan barang yang di tempatkan pada wadah atau ruang yang digunakan semaksimal mungkin berdasarkan jenis, harga dan tingkat kepetingan barang yang dikirimkan. Pendekatan metode pengembangan sistem pada knapsack problem dengan algoritma genetika adalah SDLC (*System Development Life Cycle*). Metode pengembangan sistem dengan SDLC meliputi: identifikasi masalah, penentuan kebutuhan informasi, analisis kebutuhan sistem, disain sistem, pengembangan dan dokumentasi perangkat lunak, ujicoba dan evaluasi. [3]

2.1. Gambaran Umum Sistem

Proses yang terjadi pada sistem adalah menentukan barang-barang yang paling optimal dari sisi berat untuk ditempatkan ke dalam wadah atau tempat penampungan tetapi tidak melebihi dari kapasitas daya tampung. Metode optimasi menggunakan algoritma genetika. Gambar 1 dibawah ini adalah alur diagram.



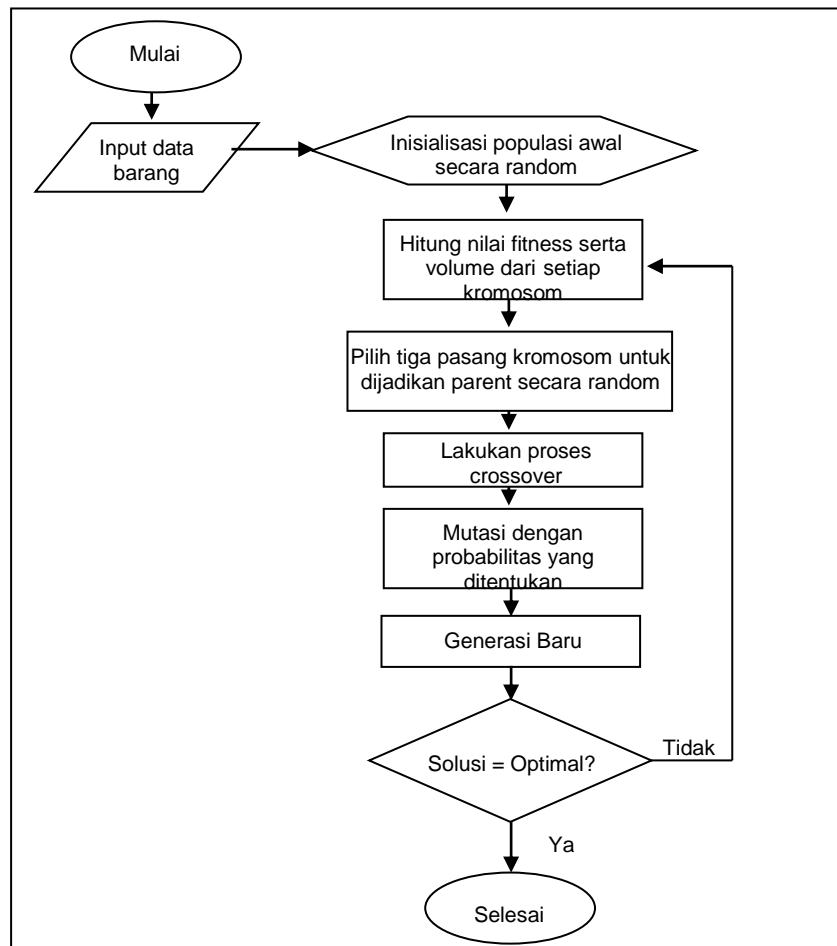
Gambar 1. Alur Diagram Sistem

2.2. Metode Perancangan Sistem

Knapsack problem merupakan permasalahan klasik yang sering terjadi dalam menempatkan barang-barang yang memiliki berbagai jenis bentuk serta bagaimana memaksimalkan ruang

yang tersedia. Tujuan adalah pengiriman barang dapat dimaksimalkan sehingga keuntungan dapat ditingkatkan. Ada beberapa parameter yang digunakan pada disain sistem yang dibangun untuk knapsack problem dengan algoritma genetika: terdapat berbagai jenis barang yang memiliki berat berbeda-beda serta harga dan tingkat kepentingan barang, berat barang yang di kemas tidak melebihi kapasitas dari knapsack, nilai parameter sistem dengan algoritma genetika dapat diubah sesuai dengan kebutuhan user seperti jumlah populasi dan probabilitas mutasi serta probabilitas crossover, hasil akhir yang disajikan adalah barang-barang yang memiliki nilai terbaik dan berat paling maksimal yang bisa di masukkan ke dalam wadah atau tempat penampungan.

Mekanisme penyelesaian knapsack problem dalam penelitian ini adalah dengan langkah-langkah yang terdapat pada algoritma berikut ini: langkah pertama inialisasi nilai awal fitness dari kromosom = 0 dimana ruang kosong minimum = 1 dan repairing = 0. Langkah kedua lakukan randomize terhadap barang untuk isi kromosom (alel), selama jumlah fitness kurang dari sama dengan luas wadah atau daya tampung dan ruang kosong lebih besar ruang kosong minimum maka nilai repairing = repairing + 1 jika jumlah fitness lebih besar dari luas gudang, dan jika nilai repairing lebih besar 3 maka alel tersebut dihapus dan random dihentikan. Langkah ketiga jika diperoleh nilai fitness lebih dari luas wadah atau daya tampung, maka dilakukan repairing dengan melakukan randomize pada alel ketiga. Gambar 2 dibawah ini menunjukkan jalannya proses pada *knapsack* problem dengan algoritma genetika.



Gambar 2. Flowchart Proses Sistem

2.3. Teknik Pengkodean Kromosom

Pengkodean kromosom diuraikan dengan gen-gen penyusun kromosom berupa abjad sesuai dengan urutan barang dan setiap gen memiliki berat dan harga. Panjang kromosom tergantung

jumlah total barang yang dapat ditempatkan pada wadah atau daya tampung. Sebagai contoh terdapat daya tampung barang sebesar 50. Adapun barang yang akan ditempatkan adalah sebagai berikut:

Tabel 1. Daftar Barang

Barang Ke-	Berat	Harga
A	15	Rp. 15.000
B	10	Rp. 10.000
C	25	Rp. 5.000
D	25	Rp. 25.000

Terdapat 3 barang yang dipilih sesuai dengan kapasitas atau daya tampung knapsack dari tabel 1 diatas, sehingga kromosom yang terbentuk dapat diuraikan sebagai berikut: kromosom[1]=ABC, kromosom[2]=CB, kromosom[3]=ABD, urutan abjad dilakukan secara acak. Pada proses crossover model yang digunakan adalah cutpoint serta mutasi dilakukan berdasarkan probabilitas mutasi dari nilai inputan parameter yang dilakukan oleh user.

2.4. Teknik Evaluasi Solusi

Teknik evaluasi terhadap solusi yang dihasilkan di setiap generasi untuk menentukan kesesuaian dengan fungsi tujuan dan constraint yang ada. Proses diawali dengan memeriksa berat atau total berat barang di masing-masing kromosom pada satu generasi, apabila melebihi daya tampung dari knapsack maka kromosom tersebut di repairing (diperbaiki). Fungsi obyektif dari penelitian ini adalah nilai fitness kromosom dari generasi yang terpilih, hal ini sesuai dengan fungsi tujuan yaitu memaksimalkan penempatan barang. Adapun fungsi obyektif yang digunakan adalah sebagai berikut:

$$F.Obj = \sum_{i=1}^n weight_i \tag{1}$$

dengan batasan sebagai berikut: jumlah nilai fitness kurang dari sama dengan luas wadah atau daya tampung dan ruang kosong lebih besar dari ruang kosong minimum, repairing (perbaikan) dilakukan apabila repairing+1 jika jumlah fitness lebih besar dari wadah atau daya tampung barang, jika nilai repairing yang dilakukan lebih besar dari 3 kali, maka alel atau gen tersebut dihapus dan random dihentikan.

2.5. Teknik Pembentukan Generasi

Proses terciptanya generasi baru melalui tiga tahap yaitu seleksi, persilangan serta mutasi. Terciptanya generasi baru dengan tujuan untuk mendapatkan kromosom dengan nilai fitness terbaik yang menunjukkan solusi maksimal dalam setiap generasinya. Berikut ini adalah contoh proses pembentukan generasi dengan menggunakan algoritma genetika. Diketahui barang dengan harga yang terdapat pada tabel 2 dibawah ini, barang-barang tersebut rencananya akan ditempatkan kedalam wadah atau tempat penampungan dengan kapasitas sebesar 150 Kg.

Tabel 2. Daftar Barang dan Harga

No	Barang	Berat (kg)	Harga	Fitness
1	A	10	Rp 1,500,000.00	10
2	B	20	Rp 2,800,000.00	20
3	C	40	Rp 3,200,000.00	40
4	D	30	Rp 2,300,000.00	30
5	E	60	Rp 5,000,000.00	60
6	F	35	Rp 2,000,000.00	35
7	G	45	Rp 2,800,000.00	45
8	H	25	Rp 3,500,000.00	25
9	I	5	Rp 900,000.00	5

10	J	50	Rp 3,500,000.00	50
----	---	----	-----------------	----

Langkah pertama adalah penentuan kromosom untuk generasi pertama, prosesnya adalah sebagai berikut:

Tabel 3. Proses Pembentukan Kromosom Pertama

KROMOSOM 1				
RANDOM	BARANG	Fitness	Ruang Kosong	Repairing
8	H	25	125	0
5	E	60	65	0
4	D	30	35	0
3	C	40	-5	1
2	B	20	15	2
9	I	5	10	2
7	G	45	-35	3
Kromosom	HEDBI	140		

Tabel 4. Proses Pembentukan Kromosom Kedua

KROMOSOM 2				
RANDOM	BARANG	Fitness	Ruang Kosong	Repairing
4	D	30	120	0
3	C	40	80	0
9	J	50	30	0
7	G	45	-15	1
1	A	10	20	2
2	B	20	0	2
Kromosom	DCJAB	150		

Tabel 5. Proses Pembentukan Kromosom Ketiga

KROMOSOM 3				
RANDOM	BARANG	Fitness	Ruang Kosong	Repairing
1	C	40	110	0
3	J	50	60	0
9	D	30	30	0
10	I	5	25	0
7	H	25	0	1
Kromosom	CJDIH	150		

Tabel 6. Proses Pembentukan Kromosom Keempat

KROMOSOM 4				
RANDOM	BARANG	Fitness	Ruang Kosong	Repairing
6	F	35	115	0
4	D	30	85	0
3	C	40	45	0
8	H	25	20	0
7	G	45	-25	1

10	J	50	-75	2
KROMOSOM 4				
RANDOM	BARANG	Fitness	Ruang Kosong	Repairing
9	I	5	-80	3
Kromosom	FDCH	130		

Tabel 7. Proses Pembentukan Kromosom Kelima

KROMOSOM 5				
RANDOM	BARANG	Fitness	Ruang Kosong	Repairing
7	G	45	105	0
5	E	60	45	0
8	H	25	20	0
2	B	20	0	0
Kromosom	GEHB	150		

Tabel 8. Proses Pembentukan Kromosom Kelima

KROMOSOM 6				
RANDOM	BARANG	Fitness	Ruang Kosong	Repairing
5	E	60	90	0
2	B	20	70	0
9	I	5	65	0
6	F	35	30	0
1	A	10	20	0
7	G	45	-25	-1
10	J	50	-70	2
8	H	25	-100	3
Kromosom	EBIFA	130		

Tabel 9 dibawah ini adalah susunan kromosom yang terbentuk dari proses diatas dengan nilai fitness masing-masing.

Tabel 9. Susunan Kromosom

KROMOSOM		GEN				FITNESS
P1	H	E	D	B	I	140
P2	D	C	J	A	B	150
P3	C	J	D	I	H	150
P4	F	D	C	H		140
P5	G	E	H	B		150
P6	E	B	I	F	A	130

Langkah kedua adalah proses pindah silang atau crossover dengan menggunakan metode cutpoint berdasarkan randomize (panjang kromosom terpendek-1). Pada kromosom tabel 9 cutpoint = randomize (4-1) = 3. Adapun syarakat pada proses crossover ini adalah: parent yang melakukan crossover dilakukan secara acak dan hanya menghasilkan 6 child saja, Apabila nilai fitness lebih besar dari tempat atau daya tampung maka child tersebut dibuang, jika pada kromosom child terdapat dua alel yang sama maka direpairing dengan alel lain dari parent lain yang diajak crossover.

Tabel 10. Proses Crossover Parent P1 dan P2

KROMOSOM	GEN					Fitness
P1	H	E	D	B	I	140
P2	D	C	J	A	B	150
Cutpoint						
CHILD1	H	E	D	A	B	145
CHILD2	D	C	J	B	I	145

Tabel 11. Proses Crossover Parent P3 dan P4

KROMOSOM	GEN					Fitness
P3	C	J	D	I	H	150
P4	F	D	C	H		130
Cutpoint						
CHILD3	C	J	D	H		145
CHILD4	F	D	C	I	H	135

Tabel 12. Proses Crossover Parent P5 dan P6

KROMOSOM	GEN					Fitness
P5	G	E	H	B		150
P6	E	B	I	F	A	130
Cutpoint						
CHILD5	G	E	H	F	A	175(dibuang)
CHILD6	E	B	I	B		repairing
Proses repairing di child 2						
CHILD6	E	B	I	G		130

Dari proses crossover tersebut dilakukan seleksi terhadap parent dan child yang memiliki fitness terbaik, sehingga didapat hasil seleksi kromosom pada generasi pertama adalah:

Tabel 13. Kromosom Generasi Pertama

KROMOSOM	GEN					FITNESS
Child1	H	E	D	A	B	145
P2	D	C	J	A	B	150
P3	C	J	D	I	H	150
Child2	D	C	J	B	I	145
P5	G	E	H	B		150
Child3	C	J	D	H		145

Langkah ketiga adalah proses mutasi dengan probabilitas mutasi sebesar 0.1. Jumlah gen yang bermutasi adalah $0.01 \times 150 = 1.5$ atau sebesar 2 gen yang dimutasi. Proses mutasi dilakukan dengan pembangkitan bilangan acak sejumlah gen dalam satu generasi. Bilangan acak yang dibangkitkan sebanyak dua kali untuk melakukan pertukaran alel. Misalkan bilangan acak yang pertama dibangkitkan adalah (21, 9) dan bilangan acak kedua adalah (25, 20). Pertukaran yang dilakukan adalah posisi 21 ditukar dengan posisi 25 dan sebaliknya, serta posisi 9 ditukar dengan posisi 20 dan sebaliknya. Syarat mutasi yang dilakukan adalah jika dalam satu kromosom terdapat dua alel yang sama pada saat mutasi, maka lakukan pengulangan pembangkitan bilangan random. Perulangan mutasi juga dilakukan jika nilai fitness yang dihasilkan oleh satu kromosom karena proses mutasi melebihi dari wadah atau daya tampung.

Table 14. Mutasi Generasi Pertama

KROMOSOM		GEN				FITNESS
Child1	H	E	D	A	B	145
P2	D	C	J	I	B	145
P3	C	J	D	I	H	150
Child2	D	C	J	B	A	150
P5	C	E	H	B		145
Child3	G	J	D	H		150

3. Kajian Pustaka

Dalam penelitian ini terdapat berbagai sumber referensi yang digunakan untuk mendukung penelitian yang dilakukan. Diawah ini adalah pemaparan subbab-subbab yang digunakan dalam penelitian.

3.1. Knapsack Problem

Knapsack problem merupakan permasalahan optimalisasi untuk memaksimalkan barang-barang yang ditempatkan kedalam sebuah wadah. Barang yang dimaksud memiliki karakteristik dari sisi bentuk, ukuran, berat serta profitnya. Sehingga bagaimana memaksimalkan penempatan barang dengan karakteristik yang dimaksud dengan jumlah yang ditempat di sebuah wadah semaksimal mungkin dengan tujuan profit yang diperoleh sebesar-sebesaranya. Sebagai contoh seorang penjual peralatan rumah tangga atau kebutuhan sehari-hari yang menggunakan sepeda, penjual harus mampu menempatkan semaksimal mungkin barang yang akan dijual pada sepedanya dengan kapasitas yang terbatas tentunya selain kapasitas tempat berat barang yang di bawa juga perlu diperhatikan supaya si penjual bisa mengayuh sepedanya. Selain itu barang-barang yang dibawa adalah barang yang memiliki prioritas profit tinggi. [2]

Knapsack problem mempunyai total ukuran atau kapasitas yang disimbolkan dengan V , ada n jenis barang berbeda yang dimasukkan dalam knapsack. Barang ke- i mempunyai bobot v_i serta profit b_i . X_i merupakan total barang ke- i yang ditempatkan pada knapsack. Fungsi tujuan dari permasalahan knapsack adalah:

$$\text{Maksimum } \sum_{i=1}^n b_i \cdot X_i ; \text{ dengan constraint } \sum_{i=1}^n v_i \cdot X_i \leq V_i$$

Terdapat beberapa jenis dari knapsack problem diantaranya adalah sebagai berikut: (1) 0/1 yaitu masing-masing barang terdapat 1 bagian saja. (2) fractional knapsack problem yaitu produk atau barang boleh dimuat sebagai, hal ini sering kita temukan dalam kehidupan sehari-hari misalnya beras, gula dan lain sebagainya. (3) bounded knapsack problem yaitu masing-masing barang terdapat n bagian. (4) unbounded knapsack problem yaitu masing-masing barang tersedia lebih dari satu bagian serta jumlah tak terbatas. [4]

3.2. Algoritma Genetika

Algoritma genetika merupakan penyelesaian permasalahan dengan teknik komputasi. Awal terbentuknya berpedoman pada mekanisme proses seleksi alam atau yang sering disebut proses evolusi. Pada mekanisme evolusi, individu berkesinambungan mengalami perbaikan gen untuk beradaptasi pada lingkungan sekitar. Sehingga Individu-individu terbaiklah yang dapat bertahan hidup. Mekanisme seleksi alamiah menyebabkan perubahan gen pada individu-individu dengan perkembangbiakan didalam setiap generasinya. Pada algoritma genetika mekanisme perkembangbiakan merupakan proses dasar yang menjadi titik fokus, sehingga bagaimana memperoleh generasi keturunan yang terbaik. Dalam algoritma genetika proses pencarian dilakukan pada setiap generasinya yang berupa populasi. Populasi terdiri dari beberapa individu atau yang disebut dengan kromosom, kromosom inilah sebagai calon solusi yang dicari nilai

fitness terbaik di setiap generasinya. Kromosom tersusun atas gen dan nilai gen disebut sebagai alel. [5]

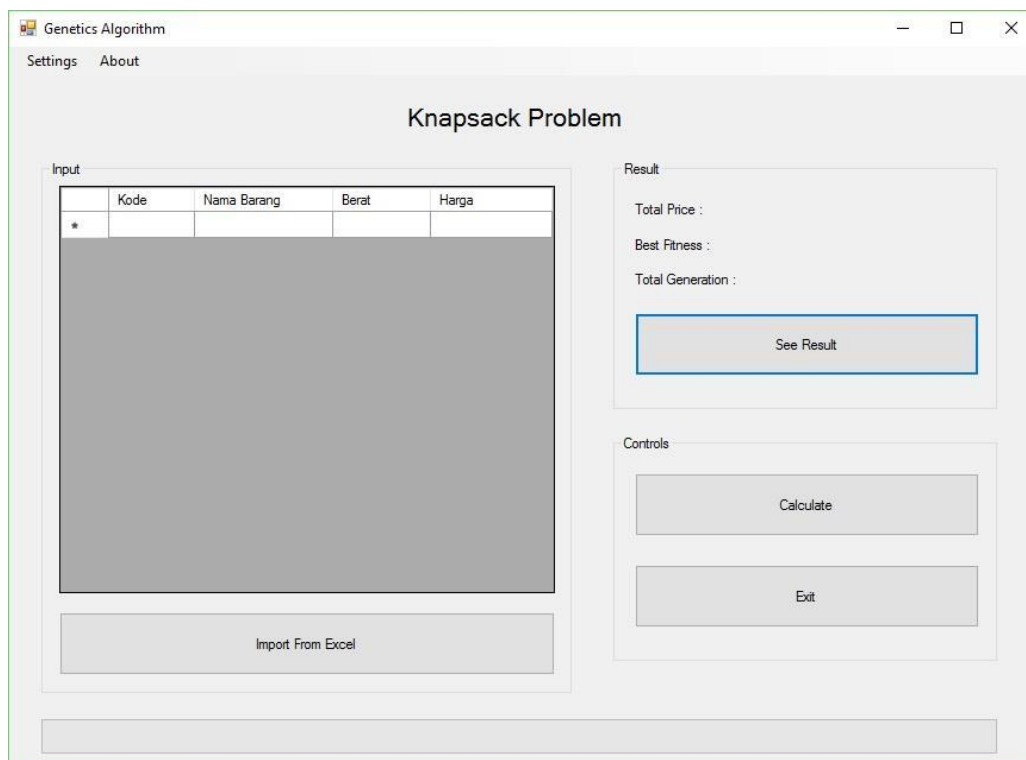
Proses algoritma genetika secara umum terbagi menjadi 5 tahapan yaitu: tahap pembentukan generasi pertama, tahap menentukan nilai fitness pada masing-masing kromosom, tahap seleksi, tahap regenerasi (crossover serta mutasi), tahapan pembentukan generasi baru. Langkah-langkah yang dikerjakan dalam algoritma genetika dapat dijelaskan sebagai berikut: (1) Mendeklarasikan bentuk kromosom. (2) menentukan fungsi fitness. (3) menentukan teknik pembangkitan populasi pertama. (4) melakukan reproduksi. (5) melakukan crossover. (6) melakukan mutasi. [5]

4. Pembahasan

Hasil yang diperoleh setelah melakukan implementasi berdasarkan metodologi dan perancangan sistem disajikan dalam screen shoot dan analisa output dari sistem yang dibangun.

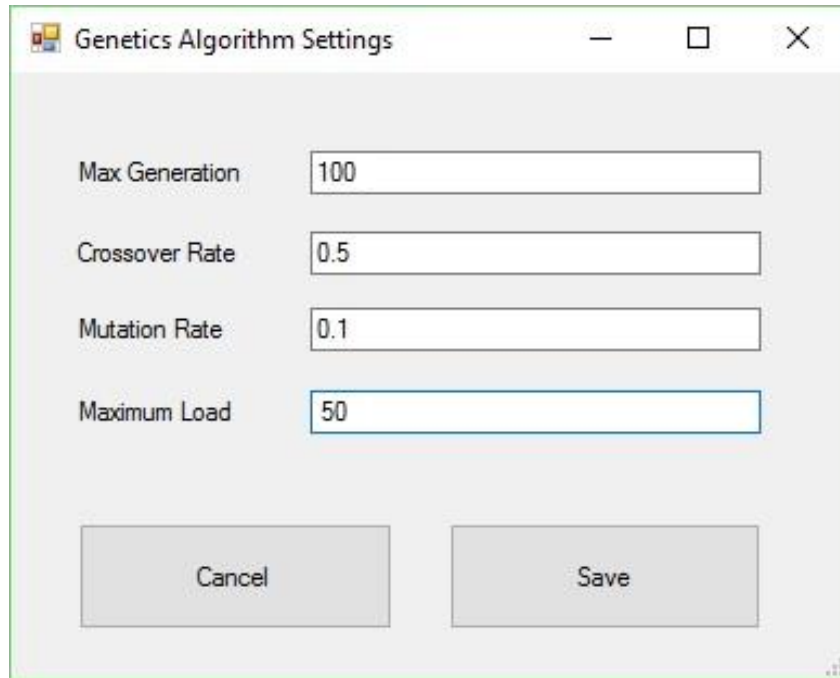
4.1. Tampilan Awal Program

Pada saat program dijalankan akan tampil seperti Gambar 3 dibawah ini, pada tampilan awal program terdapat menu setting dan about. Menu setting berfungsi untuk melakukan pengaturan variabel pada algoritma genetika dan menu about berfungsi untuk menampilkan cara menggunakan dan alur dari program.



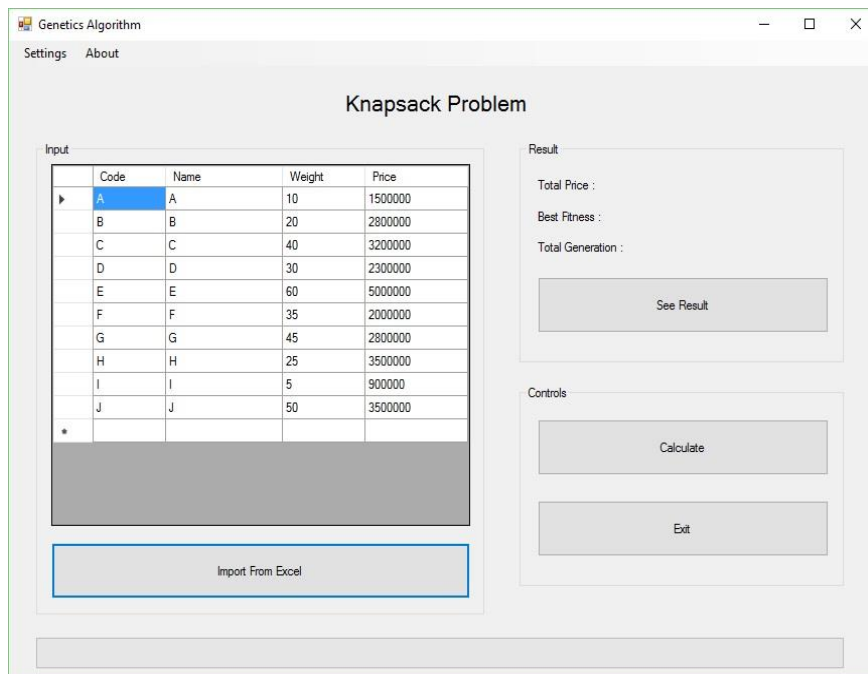
Gambar 3. Halaman Utama

Berikut contoh nilai variabel yang dimasukkan pada sistem dengan max generasi sebesar 100, crossover rate sebesar 0.5, mutation rate sebesar 0.1 dan maximum load sebesar 50.



Gambar 4. Menu Settings

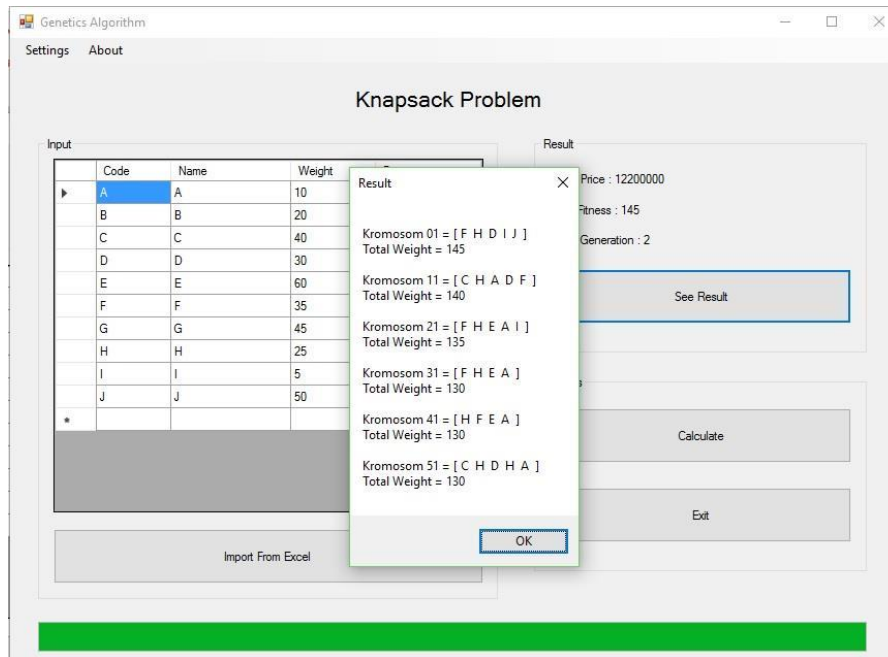
Pada menu settings terdapat pengaturan untuk variabel maksimal generasi, crossover rate, mutation rate, dan juga maksimal beban. Setelah melakukan penyimpanan maka proses selanjutnya adalah memasukkan data barang-barang mulai dari kode, name weight dan price. Gambar 5 dibawah ini adalah data barang yang sudah dimasukkan pada program aplikasi.



Gambar 5. Import Data Barang Dari Excel

Data barang pada Gambar 5 diatas adalah data yang di import dari excel yang berjumlah 10 data. Setelah proses input data langkah selanjutnya yang dilakukan adalah menjalankan program algoritma genetika dengan mengklik tombol calculate. Gambar 6 dibawah ini adalah

hasil akhir yang diperoleh.



Gambar 6. Import Data Barang Dari Excel

Gambar 6 diatas merupakan hasil dari algoritma genetika yaitu berupa total harga, fitness terbaik dan juga total generasi yang dilewati. Sistem yang dibangun memiliki toleransi error sebesar 5% untuk menghentikan iterasi algoritma genetika dan menggunakan popsize sebesar 6 individu/kromosom. Untuk melihat seluruh individu pada generasi terakhir dapat dilakukan dengan mengklik tombol "see result". Pada tampilan ini dapat dilihat seluruh kromosom yang terdapat pada generasi terakhir beserta total weight yang dimiliki oleh setiap individu.

5. Kesimpulan

Berdasarkan output dari sistem yang dibangun mengenai permasalahan knapsack dengan menggunakan algoritma genetika didapat bahwa kromosom, seleksi, crossover serta mutasi sangat menentukan proses dan hasil yang diperoleh. Dari output yang dihasilkan berdasarkan data barang yang digunakan dan tingkat error sebesar 5% diperoleh kromosom terbaik dengan kode [F H D I J] yang memiliki nilai fitness sebesar 145. Generasi optimum terjadi pada generasi ke 2. Sehingga dari data uji yang digunakan permasalahan knapsack problem dapat dioptimalkan penyelesaiannya.

Daftar Pustaka

- [1] K. Setemen, "Implementasi Algoritma Genetika Pada Knapsack Problem Untuk Optimasi Pemilihan Buah Kemasan Kotak," *Seminar Nasional Aplikasi Teknologi Informasi*, 2010, pp. 21–25.
- [2] K. D. KW, M. Fadhli, and C. Sutanto, "Penyelesaian Knapsack Problem Menggunakan Algoritma Genetika," *Seminar Nasional Informatika*, 2010, pp. 28–33.
- [3] Suyanto, *Evolutionary Computation: Komputasi Berbasis Evolusi dan Genetika*. Bandung: Informatika, 2008.
- [4] M. Hristakeva and D. Shrestha, "Solving the 0-1 knapsack problem with genetic algorithms," *Proceedings of the 37 Midwest Instruction and Computing Symposium*, 2004, Morris, MN.
- [5] S. Kusumadewi, *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu, 2003.