

**PERBANDINGAN ALGORITMA GENETIKA DAN ALGORITMA BRUTE
FORCE DALAM MASALAH PEMILIHAN MEDIA PROMOSI**

(Skripsi)

Oleh

**SHIFA TRISKA AZIZA
NPM. 2117031092**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2025

ABSTRACT

COMPARISON OF GENETIC ALGORITHM AND BRUTE FORCE ALGORITHM IN DETERMINING PROMOTIONAL MEDIA PROBLEM

By

Shifa Triska Aziza

The Knapsack Problem is an optimization problem used to minimize or maximize an objective function without exceeding a given capacity. In the context of marketing, promotional media play a crucial role in delivering product or service information to the targeted audience. Selecting the appropriate and efficient promotional media is essential to ensure that limited resources, such as budget and advertising time, are utilized optimally. This study employs two different algorithms to select promotional media: the genetic algorithm and the brute force algorithm. The data used consist of media promotion attributes, including cost and advertisement duration. The results of both algorithms show that the brute force algorithm is better than the genetic algorithm at solving the problem of selecting promotional media. The selected media are TikTok, Instagram, X, and YouTube with the cost is Rp233.148,00, the advertising time is 5 days, and the audience is 27026 people.

Keywords: Knapsack Problem, Optimization, Genetic Algorithm, Brute Force Algorithm.

ABSTRAK

PERBANDINGAN ALGORITMA GENETIKA DAN ALGORITMA BRUTE FORCE DALAM MASALAH PEMILIHAN MEDIA PROMOSI

Oleh

Shifa Triska Aziza

Knapsack Problem atau Masalah *Knapsack* merupakan permasalahan optimasi untuk meminimalisasi atau memaksimalkan suatu fungsi tujuan tanpa melebihi kapasitas yang diberikan. Media promosi memiliki peran penting sebagai sarana untuk menyampaikan informasi produk atau layanan kepada audiens yang dituju. Pemilihan media promosi yang tepat dan efisien sangat diperlukan agar sumber daya yang terbatas, seperti anggaran dan waktu, dapat digunakan secara optimal. Dalam penelitian ini digunakan dua algoritma yang berbeda untuk memilih media promosi, yaitu algoritma genetika dan algoritma *brute force*. Data yang digunakan merupakan data promosi media yang terdiri dari biaya dan waktu iklan. Hasil dari kedua algoritma tersebut menunjukkan bahwa algoritma *brute force* lebih baik daripada algoritma genetika dalam menyelesaikan masalah pemilihan media promosi. Media yang terpilih adalah *TikTok*, *Instagram*, *X*, dan *YouTube* dengan biaya yang dikeluarkan sebesar Rp233.148,00, waktu tayang iklan selama 5 hari, dan jumlah penonton sebanyak 27026 orang.

Kata kunci: *Knapsack Problem*, Optimasi, Algoritma Genetika, Algoritma *Brute Force*.

**PERBANDINGAN ALGORITMA GENETIKA DAN ALGORITMA BRUTE
FORCE DALAM MASALAH PEMILIHAN MEDIA PROMOSI**

SHIFA TRISKA AZIZA

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA MATEMATIKA

Pada

Jurusan Matematika

Fakultas Matematika dan Ilmu Pengetahuan Alam



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2025

Judul Skripsi

PERBANDINGAN ALGORITMA GENETIKA DAN ALGORITMA *BRUTE FORCE* DALAM MASALAH PEMILIHAN MEDIA PROMOSI

Nama Mahasiswa

Shifa Triska Agiza

Nomor Pokok Mahasiswa

2117031092

Program Studi

Matematika

Fakultas

Matematika dan Ilmu Pengetahuan Alam



1. Komisi Pembimbing

Dr. Notiragayu, S.Si., M.Si.

NIP. 197311092000122001

Drs. Tiryono Ruby, M.Sc., Ph.D.

NIP. 196207041988031002

2. Ketua Jurusan Matematika

Dr. Aang Nuryaman, S.Si., M.Si.

NIP. 197403162005011001

MENGESAHKAN

1. Tim Penguji

Ketua

: Dr. Notiragayu, S.Si., M.Si.



Sekretaris

: Drs. Tiryono Ruby, M.Sc., Ph.D.



Penguji

Bukan Pembimbing : Dr. Agus Sutrisno, S.Si., M.Si.



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Heri Satria, S.Si., M.Si.

NIP 197110012005011002

Tanggal Lulus Ujian Skripsi: 25 Juni 2025

PERNYATAAN SKRIPSI MAHASISWA

Yang bertanda tangan di bawah ini:

Nama : **Shifa Triska Aziza**
Nomor Pokok Mahasiswa : **2117031092**
Jurusan : **Matematika**
Judul Skripsi : **Perbandingan Algoritma Genetika dan Algoritma *Brute Force* dalam Masalah Pemilihan Media Promosi**

Dengan ini menyatakan bahwa skripsi ini adalah hasil pekerjaan saya sendiri. Apabila kemudian hari terbukti bahwa skripsi ini merupakan hasil salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan akademik yang berlaku.

Bandar Lampung, 25 Juni 2025



Shifa Triska Aziza
NPM. 2117031092

RIWAYAT HIDUP

Penulis adalah anak pertama dari tiga bersaudara yang merupakan anak dari Bapak M. Barmawi, S.E. dan Ibu Ida Yulia Mega, S.Pd. Penulis memulai pendidikan formal di TK Ibnu Rusyd dan selesai pada tahun 2009. Kemudian penulis melanjutkan pendidikan di SDN 04 Tanjung Aman pada tahun 2009-2015. Penulis melanjutkan pendidikan menengah di SMP Negeri 7 Kotabumi lalu lulus pada tahun 2018 dan menyelesaikan pendidikan menengah di SMA Negeri 1 Kotabumi pada tahun 2021. Penulis melanjutkan pendidikan di perguruan tinggi melalui jalur SBMPTN (sekarang SNBT) di Universitas Lampung pada tahun 2021.

Di bangku perkuliahan, penulis mengikuti organisasi dan kegiatan kemahasiswaan untuk meningkatkan *softskill*. Penulis pernah menjadi Anggota Badan Eksekutif Mahasiswa (BEM) FMIPA Dinas Advokasi dan Kesejahteraan Mahasiswa periode 2023. Setelah itu, penulis melakukan kegiatan kerja praktik di Kantor Dinas Pendapatan Daerah (Dispenda) Kabupaten Lampung Utara pada 26 Desember 2023 sampai dengan 3 Februari 2024.

Penulis pernah mengikuti kegiatan asistensi untuk mata kuliah Algoritma dan Pemrograman pada semester genap tahun ajaran 2023/2024.

KATA INSPIRASI

"Jika kamu tidak sanggup menahan lelahnya belajar, maka kamu harus sanggup menahan perihnya kebodohan."

(Imam Syafi'i R.A.)

"If you don't have time to read, you don't have the time (or the tools) to write."

Simple as that.

(Stephen King)

"Tidak peduli seberapa jahat dunia menilai, jalani hari dengan hati yang baik."

(Shifa Triska Aziza)

PERSEMBAHAN

Dengan mengucapkan Alhamdulillah dan syukur kepada Allah SWT atas nikmat serta hidayah-Nya sehingga skripsi ini dapat terselesaikan dengan baik. Dengan rasa syukur dan bahagia, saya persembahkan rasa terimakasih saya kepada:

Mama, Papa, dan Adik-Adikku Tercinta

Terimakasih kepada keluarga atas segala pengorbanan, motivasi, doa dan ridho serta dukungannya selama ini. Terimakasih telah memberikan pelajaran berharga kepada anakmu ini tentang ikhlas pada apa yang sudah terjadi dan makna perjalanan hidup yang sebenarnya sehingga kelak bisa menjadi orang yang bermanfaat bagi orang lain.

Dosen Pembimbing dan Pembahas

Terimakasih kepada dosen pembimbing dan pembahas yang sudah membantu, memberikan motivasi, memberikan arahan, menunjukkan kesabaran yang luar biasa, serta ilmu yang berharga.

Wisely Salaka Khibban

Terimakasih kepada dirimu yang telah banyak membantu dan menunjukkan bahwa tidak selamanya kita hidup sendiri.

Sahabat-sahabatku

Terimakasih atas waktu, dukungan, dan momen yang kalian lalui bersamaku.

Almamater Yang Dibanggakan

Universitas Lampung

SANWACANA

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini yang berjudul **"Perbandingan Algoritma Genetika dan Algoritma *Brute Force* dalam Masalah Media Promosi"** dengan baik. Shalawat serta salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW.

Dalam proses penyusunan skripsi ini, tak lepas juga pihak yang telah membantu memberikan bimbingan, dukungan, arahan, motivasi serta saran sehingga skripsi ini dapat terselesaikan. Dengan demikian, penulis mengucapkan terimakasih kepada:

1. Ibu Dr. Notiragayu, S.Si., M.Si. selaku Pembimbing I yang telah banyak meluangkan waktu dan sabar dalam memberikan arahan, bimbingan, motivasi, saran serta dukungan kepada penulis sehingga dapat menyelesaikan skripsi ini. Semoga Allah SWT. membalas semua kebaikan yang Ibu berikan. *Aamiin Allahuma Aamiin.*
2. Bapak Drs. Tiryono Ruby, M.Sc., Ph.D. selaku Pembimbing II yang telah memberikan arahan, bimbingan, dan dukungan kepada penulis sehingga dapat menyelesaikan skripsi ini.
3. Bapak Dr. Agus Sutrisno, S.Si., M.Si. selaku Penguji yang telah bersedia memberikan kritik, saran, dan motivasi, serta evaluasi kepada penulis sehingga penulis dapat menjadi pribadi yang lebih baik lagi.
4. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
5. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

6. Ibu Dr. Fitriani, S.Si., M.Sc. selaku pembimbing akademik penulis.
7. Seluruh dosen, staf, dan karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
8. Mama, Papa, dan Adik-Adikku yang selalu mendoakan dan mendukung penulis secara material maupun emosional.
9. Wisely Salaka Khibban pasangan sekaligus sahabat yang selalu membantu penulis dalam menyelesaikan banyak masalah dan selalu menjadi *support system* untuk penulis.
10. Adik saya, Aliffiya pacar Renjun yang selalu memberi dukungan kepada penulis dalam menyelesaikan masalah yang ada.
11. Sahabat dari G2OM, yaitu Bidil, Imak, Intul, dan Zala yang selalu menyemangati penulis dalam menghadapi masalah.
12. Teman penulis yang banyak memberikan semangat, dukungan, serta motivasi yang tidak bisa penulis sebutkan satu per satu namanya, dan seluruh teman-teman Jurusan Matematika Angkatan 2021.

Semoga skripsi ini dapat bermanfaat bagi kita semua. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, sehingga penulis mengharapkan kritik dan saran yang membangun untuk menjadikan skripsi ini lebih baik lagi.

Bandar Lampung, 25 Juni 2025

Shifa Triska Aziza

DAFTAR ISI

DAFTAR TABEL	v
DAFTAR GAMBAR	vii
I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	2
II TINJAUAN PUSTAKA	4
2.1 Media Informasi	4
2.1.1 Media Informasi Digital	4
2.2 Program Linier	4
2.3 <i>Knapsack Problem</i>	5
2.4 Algoritma Genetika	7
2.5 Algoritma <i>Brute Force</i>	15
2.6 Bahasa Pemrograman Python	16
III METODE PENELITIAN	17
3.1 Waktu dan Tempat Penelitian	17
3.2 Data Penelitian	17
3.3 Metode Penelitian	18
IV HASIL DAN PEMBAHASAN	20
4.1 Formulasi Model Matematika <i>Knapsack Problem</i> Pemilihan Media Promosi	20
4.2 Penyelesaian <i>Knapsack Problem</i> Pemilihan Media Promosi Menggunakan Algoritma Genetika	21
4.2.1 Penyelesaian <i>Knapsack Problem</i> Pemilihan Media Promosi Menggunakan Algoritma Genetika Secara Manual	21
4.2.2 Penyelesaian <i>Knapsack Problem</i> Pemilihan Media Promosi Menggunakan Algoritma Genetika dengan Python	28
4.2.3 Perbandingan Solusi Masalah Pemilihan Media Promosi Menggunakan Algoritma Genetika dengan Manual dan Python	30

4.3	Masalah Pemilihan Media Promosi Menggunakan Algoritma <i>Brute Force</i>	30
4.3.1	Masalah Pemilihan Media Promosi Menggunakan Algoritma <i>Brute Force</i> Secara Manual	31
4.3.2	Masalah Pemilihan Media Promosi Menggunakan Algoritma <i>Brute Force</i> dengan Python	36
4.3.3	Perbandingan Solusi Masalah Pemilihan Media Promosi Menggunakan Algoritma <i>Brute Force</i> dengan Manual dan Python	37
4.4	Hasil Perbandingan Algoritma Genetika dan Algoritma <i>Brute Force</i> dalam Masalah Pemilihan Media Promosi	38
V	KESIMPULAN DAN SARAN	39
5.1	Kesimpulan	39
5.2	Saran	39
	DAFTAR PUSTAKA	40

DAFTAR TABEL

2.1	Kromosom yang diberikan dengan nilai (keuntungan)	9
2.2	Probabilitas masing-masing kromosom	9
2.3	Barang yang tersedia	12
2.4	Pembangkitan populasi untuk generasi pertama	13
2.5	Populasi untuk generasi kedua	14
2.6	Populasi untuk generasi ketiga	14
2.7	Hasil akhir algoritma genetika di generasi ketiga	14
2.8	Barang yang tersedia	15
2.9	Semua kombinasi dari barang yang tersedia	16
3.1	Data media promosi <i>online</i>	17
4.1	Inisialisasi populasi untuk generasi pertama	22
4.2	Hasil reproduksi generasi pertama untuk pembentukan generasi kedua	23
4.3	Inisialisasi populasi untuk generasi kedua	24
4.4	Hasil reproduksi generasi kedua untuk pembentukan generasi ketiga	24
4.5	Populasi untuk generasi ketiga	25
4.6	Hasil reproduksi generasi ketiga untuk pembentukan generasi ke-empat	25
4.7	Populasi untuk generasi keempat	26
4.8	Hasil reproduksi generasi keempat untuk pembentukan generasi kelima	26
4.9	Populasi untuk generasi kelima	27
4.10	Hasil reproduksi generasi kelima untuk pembentukan generasi keenam	27
4.11	Populasi untuk generasi keenam	28
4.12	Semua kemungkinan pilihan media yang dihasilkan oleh 7 media promosi	31
4.13	Kombinasi pemilihan media yang valid dari 7 media promosi	32
4.14	Kombinasi pemilihan media yang tidak valid dari 7 media promosi .	35

4.15 Hasil perbandingan algoritma genetika dan algoritma <i>brute force</i> dalam masalah pemilihan media promosi	38
--	----

DAFTAR GAMBAR

2.1	Roda rolet berdasarkan probabilitas pada Tabel 2.2.	10
2.2	Contoh sederhana penggunaan perintah <i>%%time</i>	16
3.1	Diagram alir penelitian.	19
4.1	<i>Script</i> program Python algoritma genetika untuk masalah pemilihan media promosi.	29
4.2	<i>Output</i> yang dihasilkan berdasarkan <i>input</i> pada Gambar 4.1.	29
4.3	<i>Script</i> program Python algoritma <i>brute force</i> untuk masalah pemilihan media promosi.	36
4.4	<i>Output</i> yang dihasilkan berdasarkan <i>script</i> program Python pada Gambar 4.3.	37

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Promosi merupakan upaya mengenalkan suatu barang atau jasa kepada publik yang dilakukan oleh individu, organisasi, atau perusahaan dengan tujuan meningkatkan popularitas dan penjualan. Promosi yang menarik dapat berpengaruh secara signifikan pada target audiens yang dituju. Oleh karena itu pemilihan media promosi yang tepat merupakan langkah penting untuk memastikan pesan promosi dapat menjangkau audiens secara efektif dan sesuai dengan kebutuhan target (Marbun dkk., 2022).

Penggunaan media promosi tidak terlepas dari keterbatasan yang perlu dipertimbangkan agar strategi promosi berjalan sesuai dengan tujuan yang ingin dicapai. Beberapa keterbatasan menggunakan media promosi di antaranya adalah keterbatasan anggaran, target audiens, serta proses produksi yang memakan waktu. Kemungkinan munculnya suatu masalah yaitu tidak efisiennya hasil promosi yang diinginkan dari segi jumlah audiens. Kendala pada pemilihan media promosi seperti ini dapat didekati oleh *Knapsack Problem* atau Masalah *Knapsack*, sebuah masalah kombinatorial dengan tujuan untuk mengoptimalkan pemilihan media promosi dengan tujuan memperoleh audiens yang maksimal tanpa melebihi biaya yang ada (Ammar, 2019).

Hingga saat ini, banyak penelitian yang membahas tentang algoritma yang digunakan untuk menyelesaikan *Knapsack Problem* ataupun membandingkan algoritma mana yang lebih baik digunakan dalam penyelesaian *Knapsack Problem*. Contoh algoritma yang pernah digunakan dalam penelitian *Knapsack Problem* yaitu algoritma *greedy*, algoritma *dynamic programming*, algoritma *brute force*, algoritma genetika, dan sebagainya. Beberapa penelitian sebelumnya yaitu seperti penelitian yang dilakukan oleh Rois dkk. (2019) membahas tentang perbandingan algoritma

greedy, *dynamic programming*, dan algoritma *brute force*. Hasilnya menunjukkan bahwa algoritma *brute force* cocok digunakan untuk mengolah data dalam jumlah kecil sedangkan untuk data dalam jumlah besar lebih cocok algoritma *greedy*. Contoh penelitian lain dilakukan oleh Iriany dkk. (2017) yang mencari solusi optimal dari pendistribusian produk menggunakan algoritma genetika serta penelitian lain yang dilakukan oleh Rija dkk. (2012) yaitu mencari solusi penjadwalan latihan menggunakan algoritma *brute force*.

Berdasarkan penelitian terdahulu, pada penelitian ini membahas tentang perbandingan algoritma genetika dan *brute force* dalam masalah pemilihan media promosi. Pada kasus ini, pencarian perbandingan solusi ini menggunakan bantuan *software Python* dengan acuan berapa lama waktu *running* program yang digunakan untuk mencari solusi optimal dari masalah pemilihan media promosi.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah perbandingan algoritma genetika dan algoritma *brute force* untuk pemilihan media promosi secara manual dan dengan bantuan *software Python*.

1.3 Tujuan Penelitian

Tujuan yang ingin didapat dari penelitian ini adalah,

1. menentukan penyelesaian *Knapsack Problem* menggunakan algoritma genetika dan algoritma *brute force* dengan bantuan pemrograman *Python*,
2. membandingkan hasil *Knapsack Problem* yang diperoleh dari algoritma genetika dan algoritma *brute force*.

1.4 Manfaat Penelitian

Manfaat yang ingin dituju dari penelitian ini adalah,

1. memberikan literasi kepada pembaca mengenai *Knapsack Problem* khususnya penyelesaian dengan menggunakan algoritma genetika dan algoritma

brute force,

2. menambahkan referensi untuk penelitian berikutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Media Informasi

Perkembangan media informasi menunjukkan perjalanan panjang dari komunikasi melalui lisan hingga pemanfaatan era digital saat ini. Setiap inovasi teknologi membawa perubahan dalam cara manusia berinteraksi dan mendistribusikan informasi, menciptakan masyarakat yang lebih terhubung. Media informasi dapat diartikan sebagai alat yang mengumpulkan dan menyusun informasi sehingga menjadi pengetahuan yang bermanfaat bagi penerima. Unsur-unsur yang termuat dalam media informasi seperti alat grafis, fotografis, atau elektronik yang digunakan untuk menangkap, memproses, dan menyusun kembali informasi (Hasan dkk., 2023).

2.1.1 Media Informasi Digital

Media informasi digital adalah sarana yang menggunakan teknologi digital untuk menyebarkan informasi kepada audiens. Menurut Syarifuddin dan Utari (2022) proses digital menggunakan logika algoritma, membuat program-program komputer dan perangkat lunak seperti media sosial, data dan database, serta buku elektronik. Contoh media sosial yang digunakan dalam beberapa tahun belakangan ini seperti *Facebook*, *X*, *Instagram*, *YouTube*, dan *Tiktok*.

2.2 Program Linier

Perkembangan program linier pada abad ke-20 merupakan kemajuan yang pesat di bidang ilmiah. Sebuah program linier adalah masalah optimasi di mana variabel-variabel dalam masalah dibatasi oleh sekumpulan persamaan dan atau pertidaksamaan linier. Dengan batasan-batasan tersebut, suatu fungsi (disebut fungsi objektif)

harus dimaksimalkan atau diminimalkan (Hilier dan Lieberman, 2009). Fungsi ini bergantung secara linier terhadap variabel-variabel tersebut. Terminologi dasar terkait program linier ada beberapa macam, yaitu:

1. Variabel keputusan

Variabel keputusan ini yang akan mempengaruhi nilai tujuan.

2. Fungsi kendala

Fungsi kendala yaitu suatu persamaan atau pertidaksamaan yang harus dipenuhi oleh variabel keputusan.

3. Fungsi tujuan atau objektif

Fungsi tujuan merupakan bentuk dari maksimisasi atau minimisasi yang nilainya bergantung berdasarkan kendala-kendala dan juga variabel keputusan.

4. Kendala batas

Kendala batas dalam program linier merujuk pada batasan-batasan matematis yang diterapkan pada variabel-variabel dalam suatu masalah optimasi.

Pada tahun 1947, George Dantzig, tokoh matematika yang mengembangkan salah satu metode dalam menyelesaikan permasalahan program linier yaitu metode simpleks. Metode simpleks merupakan penemuan yang berdampak besar untuk program linier. Bentuk umum dari program linier dapat ditulis sebagai berikut (Hilier dan Lieberman, 2009).

$$\text{maksimalkan/minimalkan } z = \sum_{j=1}^n p_j x_j \quad (2.2.1)$$

$$\text{dengan kendala } \sum_{j=1}^n a_{ij} x_j (\leq, =, \geq) b_i, \quad \forall i = 1, \dots, m \quad (2.2.2)$$

$$x_j \geq 0, \quad \forall j = 1, \dots, n. \quad (2.2.3)$$

2.3 Knapsack Problem

Knapsack Problem atau Masalah *Knapsack* merupakan permasalahan optimasi dengan minimalisasi atau maksimalisasi suatu fungsi tujuan tanpa melebihi kapasitas yang diberikan. Optimasi seringkali diterapkan dalam kehidupan sehari-hari, seperti pada bidang kesehatan, logistik, telekomunikasi, manufaktur, dan lain-lain. Opti-

masi dibagi menjadi beberapa macam, salah satunya yaitu optimasi kombinatorial (Aristoteles dkk., 2015).

Masalah *Knapsack* merupakan suatu permasalahan optimasi kombinatorial yang melibatkan n barang dengan masing-masing bobot w_i dan nilai p_i yang berbeda. Tujuannya adalah untuk memilih beberapa barang dengan nilai yang tinggi dengan total bobot yang dimiliki *knapsack* tidak melebihi kapasitas yang ada. Dalam bentuk persamaan matematika, Masalah *Knapsack* secara umum dapat diformulasikan sebagai berikut (Dantzig, 1963).

$$\max z = \sum_{i=1}^n p_i x_i \quad (2.3.4)$$

$$\sum_{i=1}^n w_i x_i \leq W \quad (2.3.5)$$

dengan,

$x_i = 1$ jika barang dimasukkan ke karung- i , dan 0 jika tidak.

$p_i = \text{profit}$ atau nilai dari barang pada karung- i .

$w_i =$ bobot barang yang dimasukkan ke karung- i .

$W =$ kapasitas maksimum dari *knapsack*.

Adapun bentuk dari Masalah *Knapsack Multiple Constraint* diformulasikan pada persamaan berikut (Chu dan Beasley, 1998).

$$\text{maksimalkan} \quad \sum_{j=1}^n p_j x_j \quad (2.3.6)$$

$$\text{dengan kendala} \quad \sum_{j=1}^n w_{ij} x_j \leq c_i, \quad \forall i = 1, \dots, m, \quad (2.3.7)$$

$$x_j \in \{0, 1\}, \quad \forall j = 1, \dots, n. \quad (2.3.8)$$

dengan,

p_j : *profit* atau keuntungan atau nilai dari barang j

w_{ij} : pemakaian kapasitas barang j pada sumber i

c_i : kapasitas dari sumber i

$$x_j = \begin{cases} 1, & \text{jika barang } j \text{ terpilih,} \\ 0, & \text{selainnya.} \end{cases}$$

Diasumsikan juga tanpa menghilangkan keumuman, p_j , w_{ij} , dan c_i adalah bilangan bulat positif.

2.4 Algoritma Genetika

Algoritma Genetika atau *Genetic Algorithm* (GA), yang dapat diterapkan dalam bioinformatika, filogenetik, ilmu komputasi, teknik, ekonomi, kimia, manufaktur, matematika, fisika, farmakometri, dan bidang lainnya didasarkan pada algoritma pencarian seleksi alam dan genetika. Secara umum, proses awal algoritma genetika adalah mencari calon solusi-solusi (kromosom) dengan tujuan mendapatkan solusi yang optimal (Berberler dkk., 2013).

Proses penentuan solusi optimal menggunakan algoritma genetika dilakukan dengan menggabungkan solusi yang mungkin dari suatu generasi ke generasi-generasi berikutnya. Operator dasar algoritma genetika ada 3, yaitu *crossover*, mutasi, dan seleksi (Holland, 1992).

Beberapa istilah dasar yang penting dalam algoritma genetika dijelaskan sebagai berikut (Holland, 1992).

1. Gen

Gen direpresentasikan sebagai suatu solusi dalam bentuk *string* biner (kromosom), setiap posisi di *string* tersebut adalah gen, yang bisa bernilai 0 atau 1.

2. Kromosom

Gabungan dari beberapa gen yang memiliki nilai.

3. Individu

Individu juga disebut sebagai salah satu solusi yang mungkin atau kromosom itu sendiri.

4. Populasi

Sekumpulan individu yang akan menjalani proses evolusi

5. Generasi

Generasi merupakan putaran siklus dalam evolusi atau iterasi pada algoritma genetika.

Evolusi biasanya dimulai dari populasi individu yang dihasilkan secara acak dan terjadi secara turun-temurun. Di setiap generasi, *fitness* setiap individu dalam populasi dievaluasi. Banyak individu yang *fit* dipilih secara stokastik dari populasi saat ini, dan gen masing-masing individu diubah (dikombinasikan ulang dan mungkin bermutasi secara acak) untuk membentuk populasi baru. Populasi baru tersebut kemudian digunakan dalam iterasi berikutnya. Umumnya algoritma berakhir ketika jumlah generasi maksimum telah dihasilkan atau tingkat *fitness* yang terpenuhi (Iriany dkk., 2017). Berikut ini rincian langkah-langkah dalam penyelesaian algoritma genetika khususnya dalam menyelesaikan *Knapsack Problem*.

1. Inisialisasi populasi atau pembangkitan populasi.

Pembangkitan populasi dapat ditentukan jumlah kromosomnya sendiri, semakin banyak populasi yang dibangkitkan maka akan semakin bervariasi hasilnya tidak berhenti dalam solusi lokal. Biasanya jumlah populasi terbatas sampai dengan 2^n dengan n merupakan jumlah barang.

2. Menentukan *fitness* dari setiap kromosom yang dibangkitkan.

Dalam penyelesaian *Knapsack Problem*, cara menghitung nilai *fitness* bergantung dengan kapasitas dari masing-masing kendalanya. Pada kasus ini, perhitungan nilai *fitness* menggunakan fungsi berikut.

$$Fitness(x) = \begin{cases} \sum_{i=1}^n c_i \cdot x_i, & \text{jika } \sum_{i=1}^n k_{ij} \cdot x_i \leq K_j, \forall j \in \{1, 2, \dots, m\} \\ 0, & \text{selainnya.} \end{cases}$$

Dengan keterangan:

- $x_i \in \{0, 1\}, \forall i \in \{1, 2, \dots, n\}$: solusi dalam bentuk kromosom (0 = tidak dipilih, 1 = dipilih),
- c_i : nilai dari barang ke- i ,
- k_{ij} : pemakaian sumber daya ke- j oleh barang ke- i (misalnya biaya, waktu, dan kapasitas),
- K_j : batas maksimum untuk kendala ke- j ,
- m : jumlah banyaknya kendala,
- n : jumlah banyaknya barang.

3. Seleksi kromosom untuk menjadi *parent*.

Tahap seleksi kromosom ini dilakukan dengan cara memilih kromosom yang valid dalam langkah sebelumnya. Ada beberapa jenis cara seleksi kromosom, berikut ini yang lebih diketahui secara umum.

(a) *Roulette Wheel Selection* (Seleksi Roda Rolet)

Probabilitas pemilihan *parent* memiliki nilai yang proporsional terhadap jumlah *fitness*-nya.

Diberikan Tabel 2.1 yang menyajikan populasi dengan masing-masing nilai (keuntungan).

Tabel 2.1 Kromosom yang diberikan dengan nilai (keuntungan)

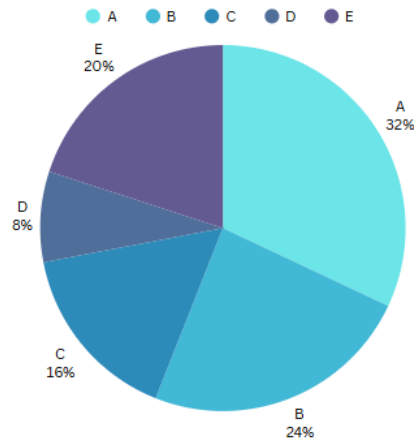
Kromosom	Nilai (keuntungan)
A	40
B	30
C	20
D	10
E	25

Contoh gambaran seleksi yang menggunakan sistem roda rolet sebagai berikut. Disajikan masing-masing probabilitas kromosom secara proporsional dalam Tabel 2.2.

Tabel 2.2 Probabilitas masing-masing kromosom

Kromosom	Nilai (keuntungan)	Probabilitas	Kumulatif Probabilitas
A	40	0.32	0.32
B	30	0.24	0.56
C	20	0.16	0.72
D	10	0.08	0.8
E	25	0.2	1

Diperoleh roda rolet yang sesuai berdasarkan probabilitas masing-masing kromosom pada Tabel 2.2.



Gambar 2.1 Roda rolet berdasarkan probabilitas pada Tabel 2.2.

(b) *Tournament Selection* (Seleksi Turnamen)

Mengambil beberapa individu acak, lalu pilih yang terbaik dari *offspring* (keturunan) yang dihasilkan. Contoh seleksi dengan menggunakan *tournament selection* seperti berikut. Misalnya individu yang akan diseleksi untuk menjadi *parent* adalah sebanyak 2 kromosom.

Turnamen 1:

- i. Ambil dua individu secara acak, misal pilih C dan E.
- ii. Bandingkan nilai *fitness* kedua individu terpilih, karena keuntungan $C < E$ maka individu E yang terpilih menjadi *parent*.

Turnamen 2:

- i. Pilih dua individu secara acak, misal pilih A dan D.
- ii. Bandingkan nilai *fitness* kedua individu terpilih, karena keuntungan $A > D$ maka individu A yang terpilih menjadi *parent*.

Setelah diperoleh dua *parent* dari dua seleksi turnamen, maka selanjutnya dapat dilakukan proses *crossover*. Seleksi turnamen dilakukan sebanyak *parent* yang dibutuhkan berdasarkan parameter yang ditentukan.

(c) *Elitism* (Elitisme)

Berikutnya ada seleksi dengan elitisme, dimana individu yang memiliki nilai objektif yang paling tinggi dan tidak melebihi batas *knapsack* dipastikan dipilih untuk generasi berikutnya. Contoh untuk proses seleksi elitisme, misalkan algoritma genetika digunakan untuk mencari hasil yang terbaik dengan ketentuan jumlah generasinya hanya 3 dan hanya

1 elit yang dipilih dari setiap generasinya. Dari hasil generasi pertama yang disajikan pada Tabel 2.1, nilai (keuntungan) yang paling tinggi dimiliki oleh individu A dengan nilai 40. Oleh karena itu, individu A akan dipilih untuk menjadi salah satu individu pada populasi generasi kedua. Berlaku hal yang sama untuk generasi kedua dan ketiga.

4. Melakukan *crossover* (penyilangan).

Crossover merupakan proses menggabungkan gen dari kedua *parent* dengan tujuan menghasilkan individu baru yang disebut *offspring* (keturunan). Biasanya *crossover* dilakukan berdasarkan parameter *rate* (kemungkinan) yang ditentukan sendiri (dinotasikan sebagai P_c), *offspring* yang dihasilkan diharapkan memiliki genetika yang lebih baik daripada *parent*-nya. Secara umum, teknik *crossover* dilakukan dengan cara *1-point crossover* yaitu dengan menyilangkan gen dimulai dari urutan gen tertentu. Pemilihan letak terjadinya *crossover* dilakukan dengan cara membangkitkan bilangan acak, seperti lempar dadu. Contoh dari proses *1-point crossover* dapat dilihat di bawah ini.

Parent 1 : [1, 0, 1, 0, 1]

Parent 2 : [0, 1, 0, 0, 0]

Misalkan *crossover* dilakukan pada gen kedua, maka menghasilkan proses sebagai berikut.

Parent 1 : [1, 0 | 1, 0, 1]

Parent 2 : [0, 1 | 0, 0, 0]

Offspring yang dihasilkan sebagai berikut.

Offspring 1 : [1, 0, 0, 0, 0]

(Hasil penyilangan dari dua gen pertama *Parent 1* dan tiga gen terakhir *Parent 2*)

Offspring 2: [0, 1, 1, 0, 1]

(Hasil penyilangan dari dua gen pertama *Parent 2* dan tiga gen terakhir *Parent 1*)

Selain teknik *1-point crossover*, adapun contoh penggunaan teknik *crossover* yang lain yaitu seperti *K-point crossover*. Cara kerja dari *K-point crossover* sama seperti *1-point crossover*, namun terdapat sebanyak K titik dalam proses persilangan.

5. Terjadi mutasi gen.

Mutasi terjadi dengan tujuan menambahkan keberagaman atau variasi dari kromosom yang sudah ada agar populasi tidak berhenti pada pencarian solusi lokal saja. Cara kerja mutasi yang paling umum yaitu mengubah salah satu gen dari kromosom dari 0 menjadi 1 atau sebaliknya (dikenal sebagai *bit flip mutation*) berdasarkan *rate* yang ditentukan. Untuk pemilihan terjadinya kromosom dapat dilakukan dengan cara yang sama seperti *crossover* yaitu dengan melempar dadu. Misalkan Kromosom *offspring 1*: $[1, 0, 0, 0, 0]$ dengan *mutation rate* (dinotasikan sebagai P_m) sebesar 0.2. Misalkan terjadi mutasi pada gen ke-3 dan ke-4, sehingga kromosom berubah menjadi $[1, 0, 1, 1, 0]$.

6. Evaluasi terhadap kromosom yang dihasilkan.

Evaluasi dilakukan untuk menjaga nilai *fitness* agar individu-individu yang dihasilkan tetap pada kategori valid, individu yang tidak valid akan dibuang. Individu yang valid akan dipilih untuk seleksi pada generasi berikutnya.

Kompleksitas waktu diperlukan untuk mengetahui berapa banyak waktu komputasi yang dibutuhkan suatu algoritma untuk menyelesaikan suatu masalah. Kompleksitas waktu algoritma genetika bergantung pada 3 parameternya, yaitu jumlah populasi per generasi (P), jumlah generasi (G), dan panjang kromosom atau banyaknya data (n). Dengan demikian kompleksitas waktu algoritma genetika dapat dinyatakan dengan $\mathcal{O}(P \cdot G \cdot n)$ atau bersifat heuristik (Reeves dan Rowe, 2002).

Contoh 2.4.1

Misalkan tersedia 3 barang yang akan dimasukkan ke dalam sebuah tas dengan kapasitas 5 kg. Selesaikan dengan menggunakan algoritma genetika. Barang yang tersedia sebagai berikut.

Tabel 2.3 Barang yang tersedia

Barang	Berat	Nilai
A	2	3
B	3	4
C	4	5

Sebelum mencari penyelesaian, parameter untuk algoritma genetika akan ditentukan terlebih dahulu. Pada Contoh 2.4.1 parameter yang dipakai adalah sebagai berikut.

- Jumlah populasi awal : 3

- *Crossover rate* : 0.6
- *Mutation rate* : 0.1
- Jumlah generasi : 3

Setelah parameter ditentukan, maka dimulai dengan inisialisasi populasi, misalkan diberikan kromosom yang dibangkitkan secara acak dengan nilai *fitness* juga sebagai berikut.

Tabel 2.4 Pembangkitan populasi untuk generasi pertama

Kromosom	Barang yang dipilih	Berat (kg)	<i>Fitness</i>	Nilai
[1, 0, 0]	A	2	2	3
[0, 0, 1]	C	4	4	5
[1, 1, 1]	A, B, C	9	9	0

Kemudian diperoleh individu yang valid adalah yang memiliki kromosom [1, 0, 0] dan [0, 0, 1]. Langkah berikutnya adalah melakukan proses seleksi, metode yang digunakan adalah *elitism* yaitu kromosom terbaik yang valid akan diambil sebagai calon individu di generasi berikutnya. Kromosom yang valid yaitu dilihat berdasarkan kromosom yang tidak melebihi batas kapasitas yang telah ditentukan, dalam hal ini berat individu tersebut. Parameter yang digunakan adalah sebanyak 1 elit yang diambil dari kromosom yang valid. Berdasarkan Tabel 2.4, dipilih kromosom terbaik yaitu kromosom [0, 0, 1] sebagai salah satu individu pada generasi berikutnya.

Setelah proses seleksi selesai, langkah berikutnya yaitu melakukan *crossover*. Kromosom yang akan dipilih menjadi *parent* adalah [1, 0, 0] dan [0, 0, 1], sedangkan teknik *crossover* yang digunakan adalah *1-point crossover*. Berikut ini adalah proses *crossover* yang dilakukan *parent* untuk menghasilkan *offspring*.

Parent 1 : [1, 0, 0]

Parent 2 : [0, 0, 1]

1-point crossover terjadi pada gen pertama, sehingga terjadi *crossover*

Parent 1 : [1 | 0, 0]

Parent 2 : [0 | 0, 1]

yang menghasilkan *offspring* sebagai berikut.

Offspring 1 : [1, 0, 1]

Offspring 2 : [0, 0, 0]

Setelah proses *crossover* selesai, terjadi proses mutasi yang memungkinkan *offspring* berubah gen. Misalkan terjadi proses mutasi pada *offspring 2* pada gen kedua sehingga dari 0 ke 1 kromosomnya menjadi [0, 1, 0].

Offspring 1 : [1, 0, 1]

Offspring 2 : [0, 1, 0]

Sehingga, terbentuk populasi baru yang disebut generasi kedua sebagai berikut.

Tabel 2.5 Populasi untuk generasi kedua

Kromosom	Berat (kg)	<i>Fitness</i>	Nilai
[0, 0, 1] (dari seleksi <i>elitism</i>)	4	4	5
[1, 0, 1] (<i>offspring 1</i>)	6	6	0
[0, 1, 0] (<i>offspring 2</i>)	3	3	4

Dengan melakukan proses yang sama seperti pada generasi pertama, diperoleh generasi ketiga sebagai berikut.

Tabel 2.6 Populasi untuk generasi ketiga

Kromosom	Berat (kg)	<i>Fitness</i>	Nilai
[0, 0, 1] (dari seleksi <i>elitism</i>)	4	4	5
[1, 1, 0] (<i>offspring 1</i>)	5	5	7
[0, 0, 1] (<i>offspring 2</i>)	4	4	5

Berdasarkan populasi generasi ketiga diperoleh bahwa ada dua individu yang memiliki kromosom sama yaitu [0, 0, 1]. Jika *parent* yang memiliki kromosom yang sama dilakukan penyilangan maka kedua *offspring* dari *parents* tersebut akan menghasilkan kromosom yang sama untuk *offspring*-nya. Hasil akhir dari generasi ketiga setelah dilakukan oleh operator genetika (*crossover* dan mutasi), yang diperoleh adalah sebagai berikut.

Tabel 2.7 Hasil akhir algoritma genetika di generasi ketiga

Kromosom	Berat (kg)	<i>Fitness</i>	Nilai
[1, 1, 0] (dari seleksi <i>elitism</i>)	5	5	7
[1, 1, 0] (<i>offspring 1</i> yang bermutasi pada gen ke-3)	5	5	7
[1, 1, 0] (<i>offspring 2</i> yang bermutasi pada gen ke-1 dan ke-2)	5	5	7

Diperoleh bahwa hasil akhir dari generasi ke-3 yaitu menghasilkan kromosom yang sama dari semua populasi yang ada. Maka yang dapat disimpulkan solusinya adalah terpilihnya kromosom $[1, 1, 0]$ atau terpilihnya barang A dan barang B dengan berat 5 kg yang memiliki nilai 7.

2.5 Algoritma *Brute Force*

Algoritma *brute force* merupakan algoritma yang menggunakan penalaran sederhana untuk menyelesaikan sebuah masalah, terutama pada penelitian ini yaitu *Knapsack Problem*. Cara kerja dari algoritma *brute force* sangat mudah dipahami, algoritma ini bekerja dengan mencari semua kombinasi solusi yang mungkin dengan tujuan mencari solusi yang paling optimal dari semua kemungkinan (Rois dkk., 2019). Adapun langkah-langkah dalam penyelesaian *Knapsack Problem* menggunakan algoritma *brute force*, yaitu:

1. Tuliskan semua kombinasi yang mungkin dari data yang diberikan.
2. Evaluasi semua nilai dari semua kombinasi yang sudah ditulis. Perhatikan kapasitas *knapsack* juga.
3. Pilih kombinasi yang mempunyai nilai terbesar.
4. Selesai.

Kompleksitas waktu yang diperlukan komputasi untuk menyelesaikan algoritma *brute force* yaitu bergantung dengan banyaknya data yang digunakan. Kompleksitas waktu untuk algoritma *brute force* dinotasikan dengan $O(2^n)$.

Contoh 2.5.1

Misalkan tersedia 3 barang yang akan dimasukkan ke dalam sebuah tas dengan kapasitas 5 kg. Selesaikan dengan menggunakan algoritma *brute force*. Barang yang tersedia sebagai berikut.

Tabel 2.8 Barang yang tersedia

Barang	Berat	Nilai
A	2	3
B	3	4
C	4	5

Buat semua kemungkinan kombinasi dari barang yang ada sebanyak 2^n dengan n merupakan banyak barang yang tersedia, maka diperoleh $2^3 = 8$ kombinasi. Semua kemungkinan kombinasi barang disajikan pada Tabel 2.9.

Tabel 2.9 Semua kombinasi dari barang yang tersedia

Kombinasi	Kode biner	Berat (kg)	Nilai	Keterangan
1	[0, 0, 0]	0	0	Valid
2	[0, 0, 1]	4	5	Valid
3	[0, 1, 0]	3	4	Valid
4	[0, 1, 1]	7	9	Tidak valid
5	[1, 0, 0]	2	3	Valid
6	[1, 0, 1]	6	8	Tidak Valid
7	[1, 1, 0]	5	7	Valid
8	[1, 1, 1]	9	12	Tidak Valid

Diperoleh bahwa kombinasi terbaik yaitu kombinasi ke-7 dengan kode biner [1, 1, 0] atau terpilihnya barang A dan barang B dengan berat 5 kg yang memiliki nilai 7.

2.6 Bahasa Pemrograman Python

Python merupakan bahasa pemrograman tingkat tinggi yang dibuat oleh Guido Van Rossum dan dirilis pada tahun 1991. Python merupakan bahasa yang populer karena mudah dipahami (Awangga dkk., 2019). Python dipilih sebagai alat bantu penelitian karena Python memiliki pemahaman sintaks yang cukup mudah, selain itu Python juga memiliki *library* yang lengkap dan memiliki dukungan komunitas yang kuat karena Python bersifat *open source*.

Untuk pengukuran waktu eksekusi program atau *running time* program Python, digunakan perintah `%%time` yang mencatat total waktu proses kode dalam satu sel. Berikut adalah contoh sederhana penggunaan *command* atau perintah `%%time` di Python dengan kasus menghitung bilangan kuadrat dari 0 sampai 1.000.000.

```
In [2]: %%time
for i in range(1000000):
    _ = i ** 2

Wall time: 656 ms
```

Gambar 2.2 Contoh sederhana penggunaan perintah `%%time`.

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan pada semester genap tahun ajaran 2024/2025 di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang beralamatkan di Jalan Prof. Dr. Ir. Soemantri Brojonegoro, Gedong Meng, Kecamatan Rajabasa, Kota Bandar Lampung, Lampung.

3.2 Data Penelitian

Data yang digunakan dalam penelitian ini yaitu media promosi online. Data ini diperoleh pada hari Senin, 24 Maret 2025. Komponen yang digunakan dalam data ini melibatkan nama media promosi, biaya sewa media promosi dalam satuan rupiah, dan waktu yang tersedia untuk pengerjaan proyek dalam satuan hari sebagai kendala dan audiens atau peserta sebagai objektif dari *Knapsack Problem* ini. Rincian dari data diberikan pada Tabel 3.1.

Tabel 3.1 Data media promosi *online*

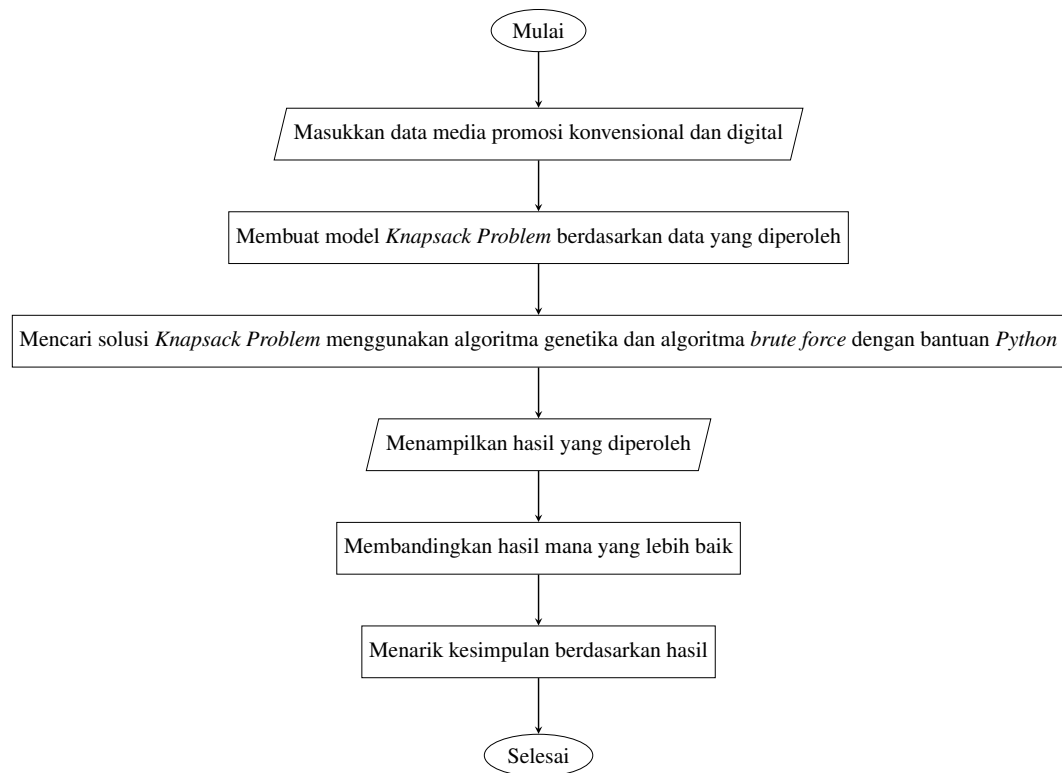
No	Nama Media	Biaya (rupiah)	Waktu (hari)	Audiens (orang)
1	<i>TikTok</i>	27952	1	2770
2	<i>Instagram</i>	55196	2	4400
3	<i>Facebook</i>	40000	2	2300
4	<i>X (dulu Twitter)</i>	50000	1	9856
5	<i>Snack Video</i>	33300	1	1810
6	<i>Snapchat</i>	83950	1	6500
7	<i>YouTube</i>	100000	1	10000

3.3 Metode Penelitian

Penelitian ini menggunakan studi literatur dengan media-media yang memberikan informasi seperti buku dan artikel ilmiah untuk acuan utama serta laman-laman pada peramban yang memberikan informasi tambahan. Langkah-langkah yang dilakukan dalam penelitian ini adalah:

1. Mencari serta memahami materi tentang *Knapsack Problem* terutama penyelesaian dengan menggunakan algoritma genetika dan algoritma *brute force*.
2. Mengumpulkan data media promosi konvensional maupun digital.
3. Menyelesaikan *Knapsack Problem* menggunakan algoritma genetika dengan bantuan pemrograman *Python*. Berikut ini langkah-langkahnya.
 - (a) Pembangkitan populasi.
 - (b) Evaluasi nilai *fitness*.
 - (c) Seleksi terhadap populasi dengan menggunakan gabungan seleksi turnamen dan elitisme.
 - (d) Lakukan proses *crossover* dan atau mutasi.
 - (e) Pemilihan generasi baru.
4. Menyelesaikan *Knapsack Problem* menggunakan algoritma *brute force* dengan bantuan pemrograman *Python*. Berikut ini langkah-langkahnya.
 - (a) Tuliskan semua himpunan bagian yang mungkin dari daftar data yang diberikan.
 - (b) Evaluasi semua nilai dari semua himpunan bagian yang sudah ditulis. Perhatikan kapasitas *knapsack* juga.
 - (c) Pilih himpunan bagian yang mempunyai nilai terbesar.
 - (d) Selesai.
5. Membandingkan hasil yang lebih baik di antara kedua algoritma yang digunakan.
6. Mengambil kesimpulan berdasarkan hasil yang didapatkan.

Adapun diagram alir penelitian sebagai berikut.



Gambar 3.1 Diagram alir penelitian.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian diperoleh bahwa algoritma *brute force* memberikan hasil yang lebih baik dari segi waktu yang dibutuhkan program Python untuk berjalan (2.72 milidetik) dibandingkan dengan algoritma genetika (46 milidetik) dalam masalah media promosi. *Running time* program Python akan berbeda saat data yang digunakan berukuran besar karena algoritma *brute force* mencari semua kemungkinan solusi, hal ini menyebabkan *running time* program Python meningkat. Sedangkan algoritma genetika mampu memberikan solusi mendekati optimal dengan waktu komputasi yang lebih efisien. Hal ini dikarenakan algoritma genetika tidak mencari semua kemungkinan kombinasi, melainkan menggunakan mekanisme seleksi, *crossover*, dan mutasi untuk mencari solusi terbaik dalam data dengan ukuran besar.

Media terpilih yang memberikan audiens paling banyak dengan biaya maksimal Rp250.000,00 dan waktu pengiklanan selama 6 hari adalah *TikTok*, *Instagram*, *X*, dan *YouTube* dengan total audiens yang diperoleh adalah 27026 orang dengan anggaran yang digunakan sebanyak Rp233.148,00 dan lama waktu pengiklanan adalah selama 5 hari.

5.2 Saran

Saran yang dapat diberikan pada penelitian ini adalah menambahkan jumlah data atau jenis data yang lain dengan jumlah yang lebih banyak (dapat menggunakan algoritma yang sama atau berbeda) agar waktu yang dibutuhkan untuk menjalankan program Python memiliki waktu yang berbeda-beda karena setiap algoritma memiliki *running time* yang berbeda bergantung pada jumlah data yang digunakan.

DAFTAR PUSTAKA

- Ammar, M. (2019). Implementasi Algoritma Greedy dalam Menyelesaikan Kasus Knapsack Problem pada Jasa Pengiriman PT. Citra Van Titipan Kilat (TIKI) Kota Makassar. *Jurnal Matematika dan Aplikasinya*, 1(2), 26-32.
- Aristoteles, A., Wardiyanto, W., dan Dwiastuti, A. (2015). Evaluasi Kinerja Genetic Algorithm dengan Strategi Perbaikan Kromosom Studi Kasus: *Knapsack Problem*. *Jurnal Komputasi*, 3(2), 162-168.
- Awangga, R. M., Prastya, R., Simamora, T. A. O., Majesty, D., dan Fadilah, I. (2019). *Dasar-dasar Python*. Kreatif, Indonesia.
- Berberler, M. E., Guler, A., dan Nuriyev, U. G. (2013). A Genetic Algorithm to Solve the Multidimensional Knapsack Problem. *Mathematical and Computational Applications*, 18(3), 486-494.
- Bustami, H. (2005). *Fundamental Operation Research*. Gramedia Pustaka Utama, Bandung.
- Chu, P. C. dan Beasley, J. E. (1998). A Genetic Algorithm for the Multidimensional Knapsack Problem. *Journal of Heuristics*, 4(4), 63-86.
- Dantzig, G. B. (1963). *Linear Programming and Extensions*. Princeton University Press, New Jersey.
- Hasan, K., Utami, A., Eni, S., Izzah, N., dan Ramadhana, S. C. (2023). Komunikasi di Era Digital: Analisis Media Konvensional vs New Media pada Kalangan Mahasiswa Ilmu Komunikasi Universitas Malikussaleh Angkatan 2021. *Jurnal Komunikasi Pemberdayaan*, 2(1), 56-63.
- Hilier, F. S. dan Lieberman, G. J. (2009). *Introduction to Operations Research*. McGraw-Hill Science/Engineering/Math, Iowa.
- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems*. Bradford Books, Leeds.

- Iriany, R., Widodo, A. W., dan Mahmudy, W. F. (2017). Pemanfaatan Algoritma Genetika Untuk Optimasi 0/1 Multi-Dimensional Knapsack Problem Dalam Pendistribusian Produk (Studi Kasus UD.TOSA). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 1(4), 257-269.
- Marbun, M.B., Ali, H., dan Dwikoco, F. (2022). Pengaruh Promosi, Kualitas Pelayanan dan Keputusan Pembelian Terhadap Pembelian Ulang (Literature Review Manajemen Pemasaran). *Jurnal Manajemen Pendidikan dan Ilmu Sosial*, 3(2), 716-727.
- Prasetyowati, M. I. dan Wicaksana, A. (2013). Implementasi Algoritma Dynamic Programming untuk Multiple Constraints Knapsack Problem. *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, 3(4), 6-13.
- Reeves, C. R. dan Rowe, J. E. (2002). *Genetic Algorithms: Principles and Perspectives*. Kluwer Academic Publishers, Dordrecht.
- Rija, H., Fadhli, M., dan Sikumbang, W. N. (2012). Aplikasi Agenda Latihan di Pusat Kebugaran dengan Pendekatan Algoritma Brute Force berbasis Android. *Jurnal Aksara Komputer Terapan*, 1(1), 1-10.
- Rois, M. A., Maslihah, S., dan Cahyono, B. (2019). Penyelesaian Integer Knapsack Problem Menggunakan Algoritma Greedy, Dynamic Programming, Brute Force dan Genetic. *Telematika*, 12(2), 87-97.
- Syarifuddin, S. dan Utari, E.D. (2022). *Media Pembelajaran: Dari Masa Konvensional Hingga Masa Digital*. Bening Media Publishing, Palembang.