

Seleksi Penerimaan Mahasiswa Baru Dengan Metode Pemecahan Masalah Algoritma Greedy Menggunakan Python

Muhammad Fauzan Nur Ilham¹, Anggi Saputra²

^{1,2}Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Muhammadiyah Kalimantan Timur

Jl. Ir. H. Juanda No.15, Sidodadi, Kec. Samarinda Ulu, Kota Samarinda, Kalimantan Timur 75124

Email : ¹2111102441149@umkt.ac.id, ²2111102441046@umkt.ac.id

Abstrak

Optimasi algoritma Greedy merupakan metode pemecahan masalah maksimasi ataupun minimalisasi. Optimasi sangat bermanfaat untuk meningkatkan performa serta produktifitas kinerja. Permasalahan Knapsack merupakan pertanyaan bagaimana memilih objek dari sejumlah besar atau berapa banyak objek yang akan disimpan agar diperoleh hasil kapasitas media penyimpanan yang optimal. Universitas melakukan penerimaan mahasiswa baru menggunakan metode seleksi. Penelitian ini difokuskan pada seleksi penerimaan mahasiswa baru. Metode yang digunakan dalam menentukan algoritma dengan hasil seleksi penerimaan mahasiswa baru yaitu metode pemecahan masalah algoritma greedy dengan python. Hasil penelitian ini menunjukkan bahwa pada mahasiswa yang belum memiliki kemampuan untuk masuk ke universitas akan didiskualifikasi dari seleksi penerimaan mahasiswa baru. Dengan hasil yang optimal dan tepat maka dapat mempermudah bagian Penerimaan Mahasiswa Baru (PMB) dalam menyeleksi mahasiswa yang pantas atau tidak diterima di universitas.

Kata kunci—Algoritma Greedy, Mahasiswa, PMB

1. PENDAHULUAN

Meskipun dapat melakukan perhitungan dibandingkan dengan orang pada umumnya, tetapi tidak dapat menyelesaikan masalah dengan cara yang sama tanpa diajari oleh seseorang dalam langkah-langkah yang telah ditentukan (algoritma). Sebuah algoritma adalah urutan langkah-langkah yang tertata untuk memecahkan permasalahan dengan memproses input menjadi output.

Tujuan analisis dari sebuah algoritma pada penelitian ini akan membuktikan metode Greedy bisa membantu dalam mengambil keputusan. Dalam hal seleksi penerimaan mahasiswa baru, dengan metode Greedy akan memberikan data yang pasti dan bisa digunakan pada penelitian selanjutnya yang mempunyai topik penelitian yang sama [1]. Seperangkat metode atau langkah-langkah untuk memecahkan masalah bertujuan untuk mencapai tujuan tertentu sering disebut dengan istilah Algoritma Strategi, di mana deskripsi metode atau angka disajikan sebagai beberapa serangkaian langkah eksekusi. Klasifikasi strategi algoritma meliputi, strategi solusi langsung (*direct solution strategy*) pemecahan masalah algoritma Brute Force dan algoritma Greedy, strategi basis searching pemecahan masalah Algoritma Backtracking dan algoritma Branch and Bound, strategi solusi *bottom-up* (strategi solusi *top-down*) membagi dan menaklukkan algoritma dan mengurangi dan menaklukkan algoritma, strategi solusi *bottom-up* pemrograman dinamis.

Beberapa algoritma dapat digunakan untuk menyelesaikan masalah tersebut, salah satunya adalah Algoritma Greedy yaitu algoritma yang merupakan metode penyelesaian masalah

optimasi yang diselesaikan secara bertahap (*step by step*). Hanya ada dua jenis masalah optimasi, yaitu maksimisasi dan minimisasi. Oleh karena itu, untuk penyelesaian persoalan optimasi yang diselesaikan secara bertahap (*step by step*) kami ingin menguji pemecahan masalah algoritma menggunakan algoritma Greedy.

2. LANDASAN TEORI

2.1. Algoritma

Algoritma ditulis dalam notasi khusus yang mudah dipahami dan notasi tersebut dapat diterjemahkan ke dalam sintaks bahasa pemrograman [2]. Algoritma ini pertama mengambil input tertentu dan kemudian menghasilkan output tertentu. Algoritma memiliki sifat tertentu yang masing-masing algoritma memiliki kekuatan dan kelemahan, sehingga perlu dipertimbangkan prosedur mana yang paling tepat untuk menyelesaikan masalah. Saat mencari algoritma perlu harus memahami:

1. Masalah apa yang akan kamu selesaikan
2. Gagasan tentang apa itu algoritma.
3. Berapa lama waktu yang dibutuhkan untuk menyelesaikan permasalahan?
4. Berapa banyak data yang dapat ditangani oleh algoritma.

2.2. Algoritma Greedy

Greedy Algorithm merupakan metode penyelesaian masalah optimasi yang diselesaikan secara bertahap dengan harapan hasil yang diperoleh akan menjadi hasil terbaik yang diperoleh [3]. Algoritma Greedy melibatkan pencarian subset kandidat yang harus memenuhi beberapa kriteria yang ditentukan, seperti optimalisasi solusi dan objek pelaporan [4], [5]. Tujuan dari algoritma serakah adalah untuk menemukan solusi yang mendekati optimal dan memaksimalkan atau meminimalkan solusi yang dihasilkan. Metode pencarian jalur terpendek rakus sangat berguna dalam menentukan rute terpendek menuju suatu tujuan [6]. Prinsip algoritma Greedy adalah "ambil apa yang bisa Anda dapatkan!", atau ambil apa yang bisa didapatkan sekarang. Algoritma Greedy memiliki beberapa langkah untuk menyelesaikan tugas [7]–[9] yaitu:

1. Memilih opsi terbaik saat ini tanpa mempertimbangkan konsekuensi di masa depan.
2. Dengan memilih optimum lokal untuk setiap langkah, kami berharap untuk mencapai optimum global.
3. Salah satu langkah saat menggunakan algoritma Greedy adalah pemilihan optimal lokal dengan harapan langkah selanjutnya mengarah ke solusi optimal global.

2.3. Knapsack Problem

Knapsack berasal dari bahasa Inggris yang berarti ransel atau karung. Knapsack adalah tas atau karung. Tas digunakan untuk membawa sesuatu. Tentu saja, tidak semua item akan muat di dalam tas [10]. Tas hanya dapat memuat beberapa item dengan ukuran total (berat) kurang dari atau sama dengan kapasitas tas [11]. Masalah Knapsack adalah metode yang dapat digunakan untuk memecahkan masalah memilih produk yang tepat ketika produk yang berbeda memiliki bobot dan keuntungan yang berbeda [12]. Tujuan tas punggung adalah untuk memilih barang yang tepat dan memaksimalkannya. Strategi untuk mengatasi masalah ransel adalah memasukkan semua barang (item) satu per satu dan mencegah barang yang sudah ada di dalam ransel dikeluarkan lagi. Strategi yang dapat digunakan untuk menempatkan barang di ransel [13].

2.4. Bahasa Pemrograman Python

Bahasa pemrograman yang bersifat interpretatif (interpreter) yang berfokus pada keterbacaan kode dan juga salah satu dari bahasa pemrograman yang terkait dengan ilmu data, Internet of Things (IoT), dan pembelajaran mesin. IoT merupakan suatu teknologi yang

mengusungkan adanya sebuah jaringan internet sebagai wadah untuk menghubungkan benda-benda di sekitar. Pada IoT, semua benda akan terhubung satu sama lain dalam satu jaringan. Dalam pengembangan sistemnya, internet membutuhkan suatu bahasa pemrograman. Suatu bahasa pemrograman akan sangat dibutuhkan untuk pengembangan sistem IoT karena internet memerlukan suatu bahasa pemrograman [14]. Beberapa keunggulan dari bahasa pemrograman Python [15]:

1. Kode dalam bahasa pemrograman Python dapat dengan mudah dipahami karena memiliki tingkat keterbacaan kode yang tinggi.
2. Library yang dimiliki bahasa pemrograman Python juga luas dan banyak.
3. Bahasa pemrograman Python juga memiliki peran penting dalam pengembangan Internet of Things .

3. METODE PENELITIAN

Bagian ini menjelaskan metode dalam penelitian ini dimana penerapan algoritma Greedy dalam seleksi Penerimaan Mahasiswa Baru (PMB) menggunakan bahasa Python. Adapun tampilan program Python ditampilkan pada Gambar 1 berikut.

```

1 # Knapsack Greedy #
2 # function def knapsack #
3
4 def knapsackGreedy():
5     print("KNAPSACK GREEDY 0/1")
6
7
8     # Tampung data barang #
9     arr = []
10
11
12     # User Input #
13
14     jum_pes = int(input("Jumlah Peserta: "))
15     for i in range(jum_pes):
16         print("Peserta ke-", i+1)
17         nam_pes = input("Nama Peserta : ")
18         ni_psi = float(input("Nilai Psikotes : "))
19         ni_akh = float(input("Nilai Akhir: "))
20         value = (ni_psi/100)*ni_akh
21         j = i
22         print("")
23         arr.append({
24             "Nama Peserta": nam_pes,
25             "Nilai Psikotes": ni_psi,
26             "Nilai Akhir": ni_akh,
27             "Value":value
28         })
29
30
31     # Urutan Peserta #
32
33     arr.sort(key=lambda x:x["Value"], reverse=True)
34     print("-"*50)
35     print("No.\tNama Peserta\tNilai Psikotes\tNilai Akhir\tValue")
36     for i in arr:
37         print("{}\t{}\t{}\t{}\t{}".format(arr.index(i)+1, i["Nama Peserta"],
38             i["Nilai Psikotes"], i["Nilai Akhir"], i["Value"]))
39
40     print("-"*50)
41
42
43     # Operasi Knapsack #
44
45     kapasitas = int(input("Kapasitas kuota peserta : "))
46     for i in arr:
47         if kapasitas >= jum_pes:
48             if i['Value'] >= 75.0:
49                 kapasitas -= j
50                 arr.append(i)
51             for c in arr:
52                 if c['Value'] < 75.0:
53                     arr.pop()
54                 arr.pop()
55
56
57     # Hasil #
58     p = len(arr)
59     print("Hasil Knapsack")
60     print("No.\tNama Peserta\tNilai Psikotes\tNilai Akhir\tValue")
61     for i in arr:
62         print("{}\t{}\t{}\t{}\t{}".format(arr.index(i)+1, i["Nama Peserta"],
63             i["Nilai Psikotes"], i["Nilai Akhir"], i["Value"]))
64     print(f"Kapasitas peserta yang masuk memenuhi nilai \t\t: {p} orang")
65     print("-"*50)
66
67
68
69 knapsackGreedy()

```

Gambar 1. Kode Program

Berdasarkan kode program pada Gambar 1 diberi penjelasan sebagai berikut:

- a. *Import*
Tahapan pertama adalah melakukan *Import os* yang berfungsi untuk membersihkan layar terminal yang sudah dijalankan sebelumnya kembali menjadi kosong. Dengan mengosongkan terminal akan memudahkan untuk melihat output dengan jelas.
- b. *Function def knapsack*
Dengan enkapsulasi program ke dalam *method* bisa mengefesienkan pengekseskuan program yang buat. Pada kode ini bisa berfikir secara modular dengan cara memasukkan kode ke dalam sebuah *method*.
- c. User Input
Pada tahap ketiga ini dilakukan input sejumlah peserta dengan memasukkan data Nama Peserta, Nilai Psikotes, dan Nilai Akhir. Kedua nilai peserta tersebut akan dihitung menjadi nilai (*value*) yang dijadikan parameter apakah peserta tersebut pantas atau tidak memenuhi nilai syarat. Lalu akan ditambahkan kedalam array list.
- d. Urutan Peserta
Pada tahap keempat ini program mengurutkan *value* secara descending dengan cara memanggil *method sort* lalu membuat fungsi lamda untuk membalikan *value* dengan mereverse urutannya. Selanjutnya, dilakukan perulangan sehingga menghasilkan semua peserta yang ada di dalam himpunan (array) yang telah dimasukkan sebelumnya.
- e. Operasi Knapsack
Pada bagian kelima ini dilakukan input kapasitas kuota peserta yang diseleksi. Peserta dengan nilai 75,0 ke atas akan ditambahkan ke dalam array dan yang memiliki nilai di bawah 75,0 akan didiskualifikasi dari seleksi penerimaan mahasiswa baru.
- f. Hasil
Pada tahap terakhir ini sama seperti tahap keempat dimana hasil yang ditampilkan hanya peserta yang memenuhi nilai syarat yang ditentukan oleh pihak Panitia PMB.

4. HASIL DAN PEMBAHASAN

Bagian ini membahas tentang struktur analisis penelitian untuk menjawab masalah penelitian. Adapun penjelasannya sebagai berikut:

Jumlah Peserta : 10
Kapasitas Peserta : 24

Tabel 1. Input Daftar Peserta

No	Nama Peserta	Nilai Psikotes	Nilai Akhir	Value
1	Fauzan	100	100	100.0
2	Anggi	100	100	100.0
3	Hafiz	100	100	100.0
4	Wahyu	100	50	50.0
5	Sultan	98	80	78.4
6	Vinna	80	86	68.8
7	Jamal	50	75	37.5
8	Udin	75	70	52.5
9	Nurul	50	60	30.0
10	Harits	100	100	100.0

Kapasitas Jumlah Peserta Yang Memenuhi Nilai : 5

Tabel 1. Output Peserta Masuk

No	Nama Peserta	Nilai Psikotes	Nilai Akhir	Value
1	Fauzan	100	100	100.0
2	Anggi	100	100	100.0
3	Hafiz	100	100	100.0
4	Harits	100	100	100.0
5	Sultan	98	80	78.4

Data pada Tabel 2 diimplementasikan ke dalam program python yang telah dibuat pada Gambar 1 sehingga memiliki output yang ditampilkan pada Gambar 2 berikut.

```
KNAPSACK GREEDY 0/1
Jumlah Peserta: 10
Peserta ke- 1
Nama Peserta : fauzan
Nilai Psikotes : 100
Nilai Akhir : 100
```

```
Peserta ke- 2
Nama Peserta : anggi
Nilai Psikotes : 100
Nilai Akhir : 100
```

```
Peserta ke- 3
Nama Peserta : hafiz
Nilai Psikotes : 100
Nilai Akhir : 100
```

```
Peserta ke- 4
Nama Peserta : Wahyu
Nilai Psikotes : 100
Nilai Akhir : 50
```

```
Peserta ke- 5
Nama Peserta : Sultan
Nilai Psikotes : 98
Nilai Akhir : 80
```

```
Peserta ke- 6
Nama Peserta : Vinna
Nilai Psikotes : 80
Nilai Akhir : 86
```

```
Peserta ke- 7
Nama Peserta : Jamal
Nilai Psikotes : 50
Nilai Akhir : 75
```

```
Peserta ke- 8
Nama Peserta : Udin
Nilai Psikotes : 75
Nilai Akhir : 70
```

```
Peserta ke- 9
Nama Peserta : Nurul
Nilai Psikotes : 50
Nilai Akhir : 60
```

```
Peserta ke- 10
Nama Peserta : Harits
Nilai Psikotes : 100
Nilai Akhir : 100
```

No.	Nama Peserta	Nilai Psikotes	Nilai Akhir	Value
1.	fauzan	100.0	100.0	100.0
2.	anggi	100.0	100.0	100.0
3.	hafiz	100.0	100.0	100.0
4.	Harits	100.0	100.0	100.0
5.	Sultan	98.0	80.0	78.4
6.	Vinna	80.0	86.0	68.8
7.	Udin	75.0	70.0	52.5
8.	Wahyu	100.0	50.0	50.0
9.	Jamal	50.0	75.0	37.5
10.	Nurul	50.0	60.0	30.0

Kapasitas kuota peserta : 24

Hasil Knapsack				
No.	Nama Peserta	Nilai Psikotes	Nilai Akhir	Value
1.	fauzan	100.0	100.0	100.0
2.	anggi	100.0	100.0	100.0
3.	hafiz	100.0	100.0	100.0
4.	Harits	100.0	100.0	100.0
5.	Sultan	98.0	80.0	78.4

Kapasitas peserta yang masuk memenuhi nilai : 5 orang

Gambar 2. Output Hasil Program

4. KESIMPULAN

Berdasarkan hasil penelitian ini maka dapat disimpulkan bahwa:

1. Metode algoritma Greedy adalah metode yang tepat diterapkan dalam pencarian mahasiswa baru dan untuk melakukan seleksi awal bagi para mahasiswa baru
2. Hasil analisis algoritma Greedy dapat membantu pihak PMB dalam membuat keputusan dalam seleksi penerimaan mahasiswa baru dengan cepat dimana hasil yang cepat diperlukan karena kekurangan mahasiswa bisa membuat universitas tidak maksimal dan tidak dapat target yang diinginkan.

5. SARAN

Penelitian selanjutnya perlu menerapkan algoritma Greedy pada sebuah universitas berdasarkan efektifitas algoritma Greedy yang bisa diandalkan dalam membuat suatu sistem penunjang keputusan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada bapak Sayekti Harits Suryawan, S.Kom., M.Kom. yang telah memberi bimbingan dalam penelitian ini.

DAFTAR PUSTAKA

- [1] Tukadi, R. Arief, and W. A. Rosyadi, "Reservasi Area Parkir Berbasis Internet Of Things," *Jurnal JE-UNISLA: Electronic Control, Telecommunication, Computer Information and Power System*, vol. 5, no. 2, pp. 370–375, 2020.
- [2] N. K. Dewi and A. S. Putra, "Sistem penunjang keputusan penerimaan karyawan baru dengan algoritma greedy," *Jurnal Visualika*, vol. 6, no. 2, pp. 154–160, 2020.
- [3] S. Rianti, H. Lubis, and R. Pahlevi, "Sistem Penunjang Keputusan Optimalisasi Barang Dengan Algoritma Greedy Pada PT Sentralindo Teguh Gemilang," *JSI (Jurnal sistem Informasi) Universitas Suryadarma*, vol. 7, no. 2, pp. 123–132, 2020.
- [4] E. Lutfina, N. Inayati, and G. W. Saraswati, "Analisis Perbandingan Kinerja Metode Rekursif dan Metode Iteratif dalam Algoritma Linear Search," *Komputika: Jurnal Sistem Komputer*, vol. 11, no. 2, pp. 143–150, 2022.

- [5] M. Syahlan, A. Syam, F. M. Faqi, and N. A. G. Basri, "Sistem Informasi Jasa Wedding Organizer Dengan Fitur Pemilihan Paket Otomatis Menggunakan Algoritma Greedy Pada Koya Wedding," in *SISITI: Seminar Ilmiah Sistem Informasi dan Teknologi Informasi*, 2020, pp. 190–202.
- [6] B. H. Lukmana, "Penerapan Algoritma Crochemore-Perri Pada Pencarian Jurnal Berbasis Mobile," *Pelita Informatika: Informasi dan Informatika*, vol. 6, no. 3, pp. 291–296, 2018.
- [7] M. R. Muttaqin and R. N. Hamrianti, "Perancangan dan Pembuatan Sistem Informasi Pemesanan Kamar Untuk Rawat Inap Pasien Di Rumah Sakit Kabupaten Purwakarta Secara Online".
- [8] S. Rianti, H. Lubis, and R. Pahlevi, "Sistem Penunjang Keputusan Optimalisasi Barang Dengan Algoritma Greedy Pada PT Sentralindo Teguh Gemilang," *JSI (Jurnal sistem Informasi) Universitas Suryadarma*, vol. 7, no. 2, pp. 123–132, 2020.
- [9] H. Sunandar and P. Pristiwanto, "Optimalisasi Implementasi Algoritma Greedy dalam Fungsi Penukaran Mata Uang Rupiah," *Jurnal Teknik Informatika UNIKA Santo Thomas*, vol. 4, no. 2, pp. 193–201, 2019.
- [10] F. Siahaan, *Desain Dan Analisis Algoritma*. Klaten, Jawa Tengah: Penerbit Lakeisha, 2020.
- [11] D. W. Nugraha, "Penerapan Kompleksitas Waktu Algoritma Prim untuk Menghitung Kemampuan Komputer dalam Melaksanakan Perintah," *Foristek*, vol. 2, no. 2, 2012.
- [12] S. Basriati, E. Safitri, and M. Ermanita, "Aplikasi Algoritma Greedy Terhadap Permasalahan Integer Knapsack pada Toko Surya Muda Pekanbaru," *Jurnal Sains Matematika dan Statistika*, vol. 6, no. 2, pp. 97–103, 2020.
- [13] Y. E. Yaremchuk, V. S. Katayev, and V. V Sinyugin, "Дослідження комбінаційних характеристик вітчизняних радіонепрозорих тканин М1, М2 та М3," *Реєстрація, зберігання і обробка даних*, vol. 17, no. 3, pp. 56–64, 2015.
- [14] R. T. Handayanto and H. Herlawati, "Machine Learning Berbasis Desktop dan Web dengan Metode Jaringan Syaraf Tiruan Untuk Sistem Pendukung Keputusan," *Jurnal Komtika (Komputasi dan Informatika)*, vol. 4, no. 1, pp. 15–26, 2020.
- [15] A. Abrar and A. Armin, "Rancang Bangun Robot Cerdas Menggunakan Raspberry PI dan Python," *JST (Jurnal Sains Terapan)*, vol. 6, no. 1, pp. 33–37, 2020.