

Implementasi *Greedy by Weight* dalam Sistem Pemesanan Rekomendasi Wisata Berdasarkan *budget* pada *Travel agent*

Gilang Yoga Pratama¹, Purwono Hendradi², Ardhin Primadewi³, Uky Yudatama⁴, Endah Ratna Arumi⁵

¹²³⁴⁵ Fakultas Teknik, Program Studi Teknik Informatika, Universitas Muhammadiyah Magelang, Indonesia

Email: ¹gilangy538@gmail.com, ²p_hendra@ummg.ac.id, ³ardhin@ummg.ac.id, ⁴uky@mail.com, ⁵arumi@ummg.ac.id

Email Penulis Korespondensi: p_hendra@ummg.ac.id

Abstrak— Yogyakarta adalah salah satu daerah dengan tempat wisata paling populer di Indonesia. Terlepas dari itu, kebanyakan wisatawan tertarik untuk datang sering kali bingung menentukan destinasi yang sesuai dengan budget mereka. Penelitian ini bertujuan untuk membantu wisatawan dalam memilih destinasi yang tepat tanpa harus melakukan perbandingan harga dan kualitas secara manual. Untuk itu, dikembangkan sebuah sistem yang digunakan untuk melakukan pemesanan berbasis web yang dilengkapi dengan fitur rekomendasi. Pada sistem ini memanfaatkan algoritma *Greedy by Weight* dengan memilih destinasi berdasarkan rating yang disebut *value* dan harga yang disebut *cost*. Sehingga, sistem akan merekomendasikan tempat-tempat dengan nilai terbaik sesuai budget yang dimasukkan pengguna. Metode yang digunakan dalam mengembangkan sistem ini adalah prototyping, di mana sistem dikembangkan dan diperbaiki berdasarkan masukan dari pengguna. Proses pengujian dilakukan dengan pendekatan *black box*, memastikan semua fitur bekerja sesuai fungsi yang diharapkan. Hasilnya, sistem dapat mempermudah wisatawan dalam menentukan destinasi, melakukan pemesanan, hingga pembayaran. Selain itu, agen *travel* juga mendapat keuntungan karena sistem membantu mengelola data pemesanan dengan lebih terstruktur. Dengan adanya sistem ini, proses merencanakan wisata jadi lebih cepat, praktis, dan sesuai kebutuhan.

Kata Kunci: Rekomendasi wisata, pemesanan *online*, *Greedy by Weight*, *budget* wisatawan, *prototyping*.

Abstract— Yogyakarta is one of the most popular tourist destinations in Indonesia. Despite from this, many interested tourists often struggle to choose a destination that fits their budget. This research aims to help travelers choose the right destination without having to manually compare prices and quality. To achieve this, a web based booking system equipped with a recommendation feature was developed. This system utilizes the Greedy by Weight algorithm, selecting destinations based on ratings and prices. The system recommends the best value destinations according to the user's budget. The system was developed using prototyping, where the system was developed and refined based on user feedback. Testing was conducted using a black box approach, ensuring all features functioned as expected. As a result, the system simplifies destination selection, booking, and payment. Furthermore, travel agents benefit from the system's ability to manage booking data in a more structured manner. This system makes travel planning faster, more practical, and more tailored to their needs.

Keywords: *Travel recommendations*, *online booking*, *Greedy by Weight*, *traveler budget*, *prototyping*.

1. PENDAHULUAN

Sektor pariwisata di Indonesia merupakan salah satu penggerak utama pertumbuhan ekonomi nasional yang didukung oleh kekayaan budaya dari lebih dari 1.300 kelompok etnis [1]. Keunikan ini menjadikan pariwisata sebagai prioritas pembangunan dalam RPJMN IV 2020–2024, dengan fokus pada peningkatan investasi, produktivitas tenaga kerja, dan daya saing ekonomi. Daerah Istimewa Yogyakarta (DIY) menjadi salah satu destinasi unggulan yang berkontribusi terhadap pendapatan daerah [2], dengan jumlah kunjungan wisatawan domestik mencapai 31.399.435 orang dan wisatawan mancanegara sebanyak 7.828 orang pada Oktober 2024 [3]. Popularitas ini mendorong perlunya inovasi dalam pelayanan wisata, khususnya bagi agen seperti Jogja *Tour & Travel* yang masih mengandalkan metode pemesanan manual berbasis WhatsApp, yang kurang efisien dan tidak menyediakan rekomendasi destinasi secara otomatis.

Pemanfaatan teknologi informasi telah terbukti meningkatkan minat wisatawan. Menurut salah satu penelitian menemukan bahwa pariwisata menjadi topik kedua yang paling banyak diakses di media sosial [4]. Beberapa penelitian sebelumnya telah menggarisbawahi pentingnya sistem rekomendasi wisata berbasis anggaran. Salah satu hal yang mendukung adalah banyaknya pengembangan sistem perencanaan wisata berbasis web yang memungkinkan pengguna memilih kota tujuan dan layanan tambahan sesuai *budget*, namun belum mengintegrasikan algoritma optimasi [5]. Sehingga bisa didukung dengan menerapkan algoritma *greedy* dan analisis hierarki untuk perencanaan rute wisata multi-kota berdasarkan waktu, biaya, dan jumlah atraksi, yang efisien tetapi belum difokuskan pada destinasi lokal [6]. Hal ini dibuktikan dengan efektivitas *greedy by weight* dalam sistem rekomendasi sembako berbasis rasio *value/cost* untuk memilih kombinasi barang terbaik tanpa melebihi anggaran [7]. Konsep itu juga digunakan untuk rekomendasi wisata di Bali berdasarkan jarak dan biaya dengan menerapkan *greedy by weight*, namun belum terhubung dengan sistem pemesanan berbasis web [8]. Beberapa penelitian lain juga menyoroti peran wisata dalam perekonomian daerah dan pentingnya digitalisasi untuk meningkatkan daya saing destinasi [9], [10].

Secara konseptual, sistem rekomendasi yang efektif harus mampu mempersonalisasi saran berdasarkan preferensi dan keterbatasan pengguna [11], [12]. Dalam konteks optimasi pemilihan, algoritma *greedy* menjadi salah satu pendekatan yang relevan karena mengutamakan solusi lokal terbaik pada setiap tahap pemilihan [13]. Pendekatan *greedy by weight* memanfaatkan perhitungan rasio *value/cost* untuk menentukan prioritas destinasi [6], [7], di mana *value* dapat berupa *rating* atau kualitas destinasi, sedangkan *cost* berupa harga tiket atau biaya perjalanan. Pemilihan destinasi dilakukan secara iteratif hingga batas anggaran tercapai dengan mempertahankan keragaman kategori destinasi [7]. Pendekatan ini dinilai sederhana, efisien, dan mampu memberikan rekomendasi cepat bagi pengguna.

Dari sisi pengembangan, metode *prototyping* merupakan pendekatan yang sesuai karena memungkinkan pengujian sistem secara bertahap dan iteratif berdasarkan masukan pengguna [14], [15]. Beberapa penelitian lain [16],



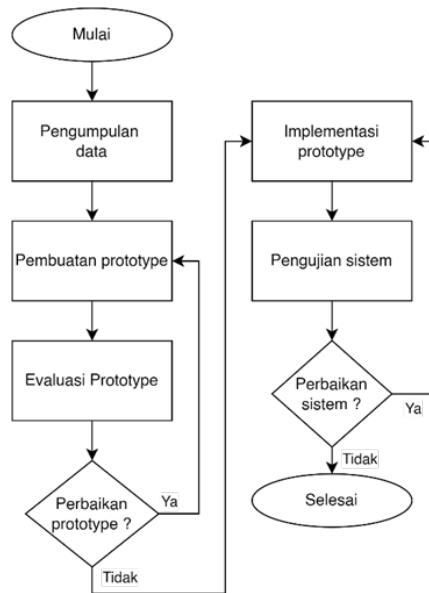
[17] menunjukkan bahwa sistem pemesanan wisata berbasis *web* dapat meningkatkan transparansi dan kepercayaan pelanggan. Hal ini bahwa *prototyping* mampu mengakomodasi kebutuhan pengguna secara dinamis melalui iterasi perbaikan yang cepat [18]. Untuk memastikan kualitas sistem, metode *black box testing* dapat digunakan untuk memverifikasi fungsionalitas tanpa melihat kode program [19]. Berdasarkan beberapa tinjauan pustaka, terdapat kesenjangan penelitian yang jelas. Terdapat penelitian yang telah mempertimbangkan faktor *budget*, namun belum mengintegrasikan algoritma optimasi [5]. Selain itu, penelitian lain menggunakan *greedy* pada skala multi-kota dan bukan destinasi lokal [6]. Terdapat penelitian yang membuktikan efektivitas *greedy by weight* di bidang non-pariwisata [7]. Namun juga terdapat platform yang mengaplikasikan *greedy* untuk wisata, tetapi tanpa sistem pemesanan terintegrasi [8]. Pada aspek metodologi, beberapa penelitian berhasil menerapkan *prototyping*, namun belum menggabungkannya dengan optimasi berbasis anggaran [14], [15].

Oleh karena itu, penelitian ini bertujuan membangun sistem pemesanan wisata berbasis web untuk Jogja *Tour & Travel* yang mengintegrasikan algoritma *Greedy by Weight* dalam memberikan rekomendasi destinasi sesuai *budget* pengguna, dikembangkan melalui metode *prototyping*, dan diuji dengan *black box testing*. Sistem ini diharapkan dapat mempermudah wisatawan dalam memilih destinasi berkualitas dengan harga terjangkau, meningkatkan efisiensi operasional agen perjalanan, serta menjadi rujukan untuk pengembangan sistem rekomendasi wisata berbasis optimasi di masa depan.

2. METODOLOGI PENELITIAN

2.1 Tahapan penelitian

Penelitian ini menggunakan pendekatan *prototyping*. Metode ini dipilih karena memungkinkan pengembangan sistem dilakukan secara iteratif, dengan melibatkan pengguna sejak tahap awal perancangan hingga implementasi akhir [20], [14]. *Prototyping* dinilai sesuai untuk membangun aplikasi yang menuntut kesesuaian tinggi dengan kebutuhan pengguna, seperti sistem pemesanan wisata berbasis *budget*.



Gambar 1. Tahapan penelitian

Berdasarkan pada Gambar 1, prosedur penelitian ini terdiri dari 5 tahapan, yaitu pengumpulan data, pembuatan *prototype*, evaluasi *prototype*, implementasi *prototype*, dan pengujian sistem. Berikut ini penjelasan dari tahapan yang dilakukan pada metode *prototyping*. Tahapan penelitian ini mengacu pada kerangka kerja *prototyping* [15] yang meliputi:

1. Pengumpulan Data

Data dikumpulkan melalui studi literatur dan wawancara. Studi literatur dilakukan untuk mengidentifikasi metode rekomendasi wisata berbasis *budget* dan implementasi algoritma *Greedy by Weight* pada berbagai bidang [6], [7]. Wawancara dilakukan dengan pengelola Jogja *Tour & Travel* serta calon pengguna untuk merumuskan kebutuhan sistem.

2. Perancangan *Prototype*

Pada tahap ini dibuat rancangan *user interface*, *user flow*, dan struktur basis data menggunakan Figma. Prinsip desain mengacu pada teori *user-centered design* untuk memastikan kenyamanan pengguna.

3. Evaluasi *Prototype*

Prototype diuji coba oleh pengguna untuk mengumpulkan umpan balik terkait tata letak, kemudahan penggunaan, dan kelengkapan informasi. Masukan dari pengguna menjadi dasar perbaikan iteratif.

4. Implementasi Sistem

Sistem dikembangkan menggunakan Laravel sebagai *backend*, MySQL untuk basis data, serta HTML, CSS, dan JavaScript untuk *frontend* [18]. Desain antarmuka mengacu pada *framework* AdminLTE agar responsif pada berbagai perangkat.

5. Pengujian Sistem

Pengujian dilakukan dengan metode *black box testing* untuk memverifikasi fungsionalitas sistem [19].

Pengujian meliputi *input budget*, rekomendasi destinasi, proses pemesanan, dan verifikasi pembayaran.

2.2 Penerapan Algoritma *Greedy by Weight*.

Greedy by Weight adalah penerapan algoritma *greedy* pada masalah *Knapsack*, di mana item diurutkan berdasarkan bobot atau biaya (*cost*) terkecil, lalu dipilih bertahap hingga kapasitas atau *budget* tercapai, sambil memaksimalkan nilai (*value*). Nilai rasio dihitung dengan persamaan (1).

$$Rasio = \frac{value}{cost} \quad (1)$$

Pemilihan destinasi dilakukan mulai dari rasio tertinggi dengan harga terendah, selama total biaya tidak melebihi *budget* dan kategori destinasi bervariasi. Hasil tersebut ditunjukkan pada Tabel 1 yang digunakan untuk melakukan pemilihan destinasi wisata berdasarkan rasio tertinggi hingga terendah.

Tabel 1. Hasil pencarian rasio

No	Destinasi	Kategori	Harga (Cost)	Rating (Value)	Rasio (Value/Cost)
1	Candi Borobudur	Budaya	Rp.50000	4.7	0.000094
2	Candi Mendut	Budaya	Rp.20000	4.8	0.00024
3	Candi Pawon	Budaya	Rp.20000	4.5	0.000225
4	Candi Prambanan	Budaya	Rp.50000	4.7	0.000094
5	Parangtritis	Laut	Rp.20000	4.6	0.00023
6	Kraton Yogyakarta	Budaya	Rp.20000	4.6	0.00023
7	Taman Sari	Budaya	Rp.20000	4.6	0.00023
8	Lavatour	Alam	Rp.20000	4.7	0.000235
9	Breksi	Alam	Rp.15000	4.5	0.0003
10	Mangunan	Alam	Rp.25000	4.6	0.000184
11	Pantai Indrayanti	Laut	Rp.10000	4.5	0.00045
12	Pantai Sadranan	Laut	Rp.10000	4.7	0.00047
13	Pantai Baron	Laut	Rp.10000	4.4	0.00044
14	Heha ocean view	Laut	Rp.30000	3.9	0.00013
15	Heha sky view	Alam	Rp.30000	4.4	0.000147

Proses pemilihan destinasi ini dilakukan dengan persamaan (2) untuk melakukan pembatasan banyaknya destinasi berdasarkan *budget* yang dimiliki.

$$(total\ biaya \leq budget) \wedge (kategori \notin K) \quad (2)$$

Pendekatan ini untuk menghasilkan kombinasi destinasi bernilai tinggi dalam batas *budget* yang ada, seperti pada penelitian [6] dan [7] yang menunjukkan keberhasilan metode ini dalam memaksimalkan jumlah item atau destinasi tanpa melanggar batas anggaran dan menjaga variasi kategori.

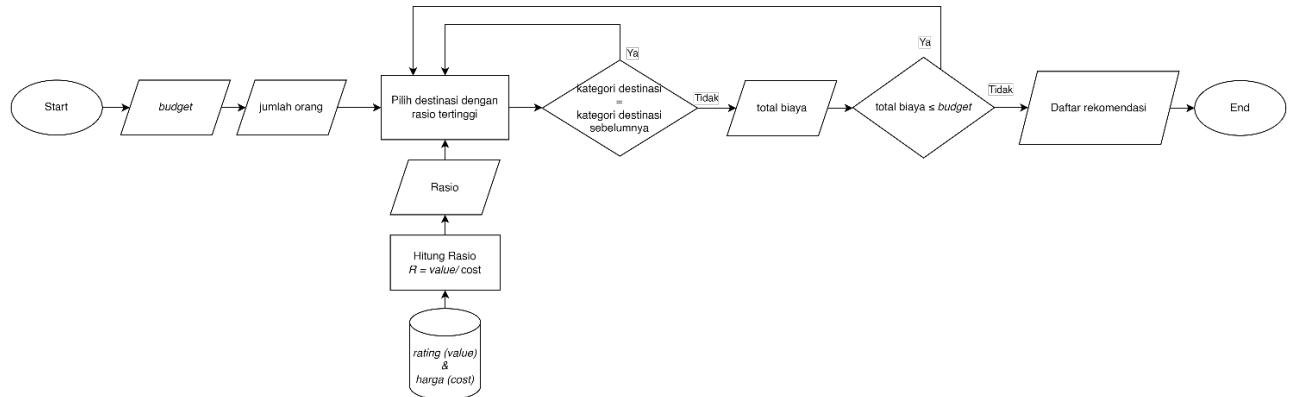
3. HASIL DAN PEMBAHASAN

Konsep yang akan dibangun dimulai ketika pengguna menginputkan nilai *budget* dan jumlah wisatawan yang akan berpartisipasi pada perjalanan tersebut. Sistem kemudian akan membagi nilai *budget* dengan jumlah orang untuk mendapatkan batas maksimum biaya per individu. Selanjutnya, sistem melakukan perhitungan nilai rasio setiap destinasi wisata dengan membagi nilai *rating* atau *value* yang merujuk pada persamaan (1). Setelah nilai rasio dari setiap destinasi dihitung, sistem memulai proses pemilihan destinasi dengan menggunakan algoritma *Greedy by Weight*.

Proses pemilihan ini dilakukan dengan mencari rasio tertinggi dengan beberapa ketentuan, seperti biaya yang tidak melebihi *budget* wisatawan dan memiliki kategori yang beragam. Proses ini dilakukan dengan menerapkan persamaan (2). Setiap kali destinasi terpilih, sistem akan memperbarui sisa *budget* dan menambahkan destinasi ke daftar rekomendasi. Apabila tidak ditemukan lagi destinasi yang memenuhi syarat rasio, harga, dan kategori, sistem akan mengizinkan pemilihan dari kategori yang sudah digunakan sebelumnya, selama masih sesuai dengan *budget*. Apabila



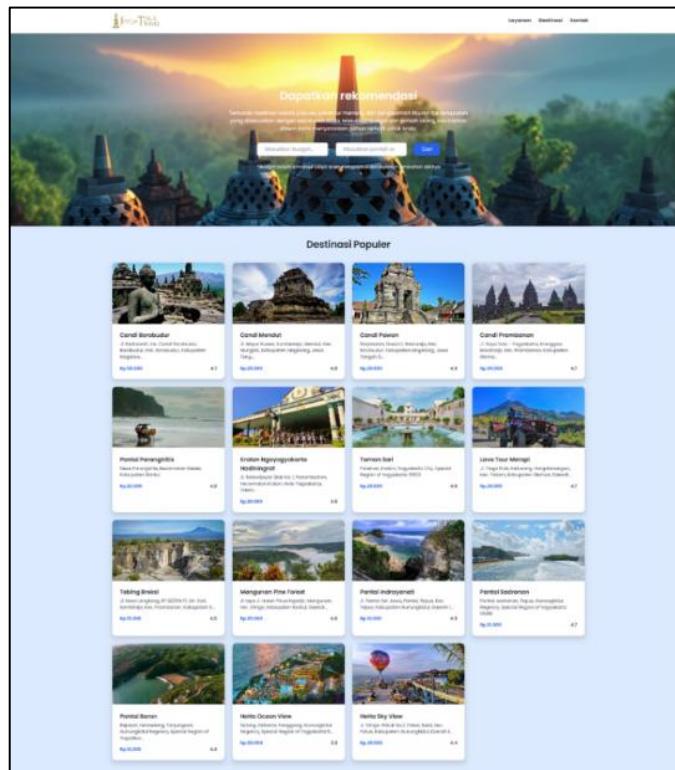
penambahan biaya destinasi berikutnya menyebabkan total biaya melebihi *budget*, maka sistem akan menghentikan proses seleksi dan menampilkan daftar rekomendasi akhir. Konsep ini dijelaskan pada Gambar 2.



Gambar 2. Konsep sistem yang dibangun

3.1 Implementasi Rancang bangun

Proses rekomendasi wisata pada sistem dimulai ketika pengguna memasukkan *budget* dan jumlah orang melalui form di `rekomendasi.blade.php`. Data ini diproses menggunakan algoritma *Greedy by Weight* dengan menghitung rasio antara rating destinasi (*value*) dan harga tiket (*cost*). Sistem kemudian memilih kombinasi destinasi dengan rasio terbaik yang masih sesuai batas *budget*, sekaligus menjaga keberagaman kategori wisata. Hasil rekomendasi ditampilkan dalam bentuk kartu berisi informasi nama, kategori, rating, harga, dan tombol pemesanan, dengan data diambil dari tabel destinasi dan `kategori_destinasi` melalui relasi model Laravel.



Gambar 3. Halaman Rekomendasi

3.1 Implementasi Algoritma *Greedy by Weight*

Fitur rekomendasi wisata merupakan inti dari sistem *JogjaTourTravel* yang bertujuan memberikan rekomendasi destinasi berdasarkan *budget* dan jumlah orang yang dimasukkan oleh pengguna. Proses implementasi fitur ini dilakukan pada `method rekomendasi()` yang berada di dalam `RekomendasiController.php`. Metode ini menerima *input* dari pengguna, mengolah data destinasi, dan menghasilkan daftar destinasi yang direkomendasikan berdasarkan algoritma *greedy* yang mempertimbangkan rasio antara *rating* dan harga. Tahap pertama dalam algoritma ini adalah proses validasi *input*. Sistem memverifikasi bahwa *field budget* dan *jumlah_orang* harus diisi dengan nilai numerik dan minimal bernilai satu. Hal ini dilakukan untuk memastikan bahwa *input* valid sebelum dilanjutkan ke tahap perhitungan.

Setelah validasi berhasil, sistem menyimpan *input budget* dan *jumlah_orang* ke dalam variabel lokal. Kemudian dihitung *budgetPerOrang*, yaitu hasil pembagian antara total *budget* dengan jumlah orang, sebagai dasar pengambilan keputusan pemilihan destinasi wisata. Selanjutnya, seluruh data destinasi diambil dari model Destinasi beserta relasi kategorinya menggunakan *method with('kategori')*. Setiap destinasi kemudian diberikan atribut baru yaitu *rasio*, yang dihitung dari pembagian antara nilai *rating* dan harga destinasi atau dengan persamaan (1). Jika harga destinasi bernilai nol, maka rasionalnya diatur menjadi nol untuk menghindari pembagian tak terdefinisi.

Untuk menghindari adanya *input* tidak valid atau kosong maka dilakukan pengecekan terlebih dahulu terhadap *budget* yang diinputkan oleh wisatawan. Apabila *budget* yang dimasukkan kurang dari harga destinasi yang ada, ataupun *budget* yang dimasukkan adalah 0, maka akan memunculkan *alert* atau peringatan agar memasukkan *budget* yang lebih sesuai. Setelah pengecekan destinasi slesai, sistem menyiapkan daftar kategori yang tersedia dari seluruh destinasi, serta variabel pendukung lain untuk memastikan bahwa pemilihan destinasi tetap memperhatikan kategori yang beragam serta tidak melebihi sisa *budget*. Sistem akan mencari rasio tertinggi pada setiap iterasi, dan hanya memilih destinasi jika memenuhi batasan *budget* dan memiliki kategori yang berbeda.

Proses utama rekomendasi dilakukan dalam struktur *while (true)*, yang akan terus berjalan selama masih ada destinasi yang memenuhi syarat untuk dipilih. Dalam setiap iterasi, sistem mencari destinasi dengan rasio tertinggi yang belum dipilih sebelumnya, memiliki kategori yang belum digunakan, dan memiliki harga tidak melebihi dari sisa *budget*. Proses ini mengimplementasikan persamaan (2). Proses pemilihan destinasi ini dimulai dengan inisialisasi variabel *\$rasioTertinggi* dan *\$selected* sebagai penampung sementara destinasi terbaik yang ditemukan dalam 1 iterasi. Proses pemilihan destinasi ini diimplementasikan pada Gambar 4.

```
43     while (true) {  
44         $rasioTertinggi = null;  
45         $selected = null;
```

Gambar 4. Kode perulangan

Selanjutnya, dilakukan iterasi kepada semua daftar destinasi untuk menemukan destinasi dengan rasio tertinggi yang belum dipilih sebelumnya, memiliki kategori yang belum dipakai, dan harga masih berada dalam sisa *budget*. Jika ditemukan destinasi dengan rasio lebih tinggi dari yang sebelumnya tersimpan, maka destinasi tersebut akan dipilih. Proses *looping* ini diimplementasikan dengan Gambar 5.

```
48     foreach ($allDestinasi as $destinasi) {  
49         $kategoriDestinasi = $destinasi->kategori->nama_kategori ?? 'Tidak Ada';  
50  
51         if (  
52             $destinasi->harga <= $sisaBudget &&  
53             !in_array($destinasi->id, $sudahDipakaiId) &&  
54             !in_array($kategoriDestinasi, $kategoriYangSudahDipakai)  
55         ) {  
56             if (!$rasioTertinggi || $destinasi->rasio > $rasioTertinggi->rasio) {  
57                 $rasioTertinggi = $destinasi;  
58             }  
59         }  
60     }
```

Gambar 5. Kode implementasi persamaan (2)

Jika ditemukan destinasi yang memenuhi semua kriteria tersebut, maka destinasi ditambahkan ke dalam daftar *\$destinasiTerpilih*, dan informasi terkait seperti kategori dan ID destinasi dicatat agar tidak dipilih ulang. Selain itu, nilai *sisaBudget* juga dikurangi dengan harga dari destinasi yang dipilih. Proses ini diimplementasikan pada Gambar 6.

```
62         if ($rasioTertinggi) {  
63             $destinasiTerpilih[] = $rasioTertinggi;  
64             $kategoriYangSudahDipakai[] = $rasioTertinggi->kategori->nama_kategori ?? 'Tidak Ada';  
65             $sudahDipakaiId[] = $rasioTertinggi->id;  
66             $sisaBudget -= $rasioTertinggi->harga;
```

Gambar 6. Kode menyimpan destinasi

Namun jika tidak ditemukan destinasi yang memenuhi syarat kategori, maka sistem melakukan *fallback* atau memperbolehkan pemilihan dari kategori yang sudah digunakan sebelumnya, selama destinasi tersebut belum dipakai

dan harganya masih dalam sisa *budget*. Sistem kembali mencari rasio tertinggi dari destinasi yang tersisa. Proses ini diimplementasikan pada Gambar 7.

```
67 } else {
68     // Fallback jika tidak ditemukan dari kategori baru
69     foreach ($allDestinasi as $destinasi) {
70         if (
71             $destinasi->harga <= $sisaBudget &&
72             !in_array($destinasi->id, $sudahDipakaiId)
73         ) {
74             if (!$selected || $destinasi->rasio > $selected->rasio) {
75                 $selected = $destinasi;
76             }
77         }
78     }
}
```

Gambar 7. Kode *fallback*

Jika ditemukan destinasi yang sesuai pada proses *fallback*, maka destinasi tersebut dipilih dan diproses seperti sebelumnya. Namun, apabila tidak ditemukan sama sekali, maka proses dihentikan. Proses tersebut diimplementasikan pada Gambar 8.

```
80     if ($selected) {
81         $destinasiTerpilih[] = $selected;
82         $sudahDipakaiId[] = $selected->id;
83         $sisaBudget -= $selected->harga;
84     } else {
85         break;
86     }
87 }
```

Gambar 8. Kode pengambilan destinasi *fallback*

Setelah itu, sistem akan melakukan pengecekan ulang apakah masih ada destinasi lain yang belum dipakai dan masih sesuai dengan sisa *budget*. Jika tidak ditemukan, maka pengulangan dihentikan sepenuhnya. Proses ini diimplementasikan pada Gambar 9.

```
90     $masihAda = $allDestinasi->first(function ($d) use ($sisaBudget, $sudahDipakaiId) {
91         return $d->harga <= $sisaBudget && !in_array($d->id, $sudahDipakaiId);
92     });
93
94     if (!$masihAda) {
95         break;
96     }
}
```

Gambar 9. Kode *stop looping*

Jika semua kategori sudah digunakan minimal sekali, maka sistem akan me-reset array *kategoriYangSudahDipakai* untuk mengizinkan kategori yang sama dipilih kembali. Hal ini dilakukan agar pemilihan tetap berjalan selama *budget* mencukupi. Proses *looping* ini diimplementasikan dengan Gambar 10.

```
99     if (count($kategoriYangSudahDipakai) >= count($kategoriTersedia)) {
100         $kategoriYangSudahDipakai = [];
101     }
102 }
```

Gambar 10. Kode *reset* kategori

Setelah proses rekomendasi selesai, hasilnya disimpan dalam *session* agar dapat diakses di halaman selanjutnya, dan sistem akan menampilkan view rekomendasi. hasil beserta data rekomendasi, *budget* awal, sisa *budget*, dan jumlah orang. Dengan struktur seperti ini, sistem tetap mempertahankan prinsip *greedy*, yaitu memilih destinasi dengan nilai efisiensi tertinggi atau *value/cost*, namun tetap memperhatikan keterbatasan anggaran dan keberagaman destinasi dari berbagai kategori.

3.2 Pengujian

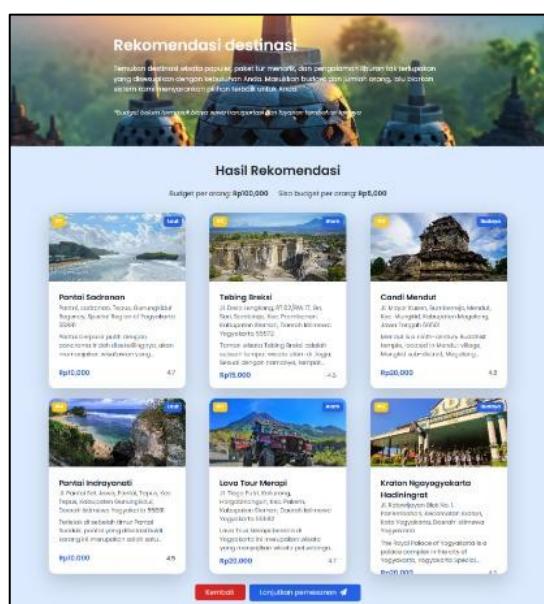
Proses pengujian berperan untuk memverifikasi kesesuaian algoritma *Greedy by Weight* yang dihitung secara manual dengan hasil implementasi pada sistem. Uji coba dilakukan menggunakan tiga responden dengan variasi *budget* rendah, sedang, dan tinggi. Pada tahap pengujian manual, setiap destinasi dihitung nilai rasio *value/cost*, kemudian dipilih secara bertahap mulai dari rasio tertinggi hingga batas *budget* tercapai, sambil menjaga keberagaman kategori destinasi. Selanjutnya, pada pengujian sistem, *input budget* dan jumlah orang dimasukkan ke aplikasi, lalu sistem secara otomatis melakukan proses perhitungan menggunakan algoritma yang sama.

Hasil pengujian menunjukkan bahwa *output* sistem selalu sama dengan hasil perhitungan manual. Sebagai contoh, Responden A dengan *budget* Rp500.000 menghasilkan 6 destinasi baik pada perhitungan manual maupun sistem. Pada *budget* sedang Rp1.200.000 diperoleh 14 destinasi, sedangkan pada *budget* tinggi Rp3.000.000 diperoleh 15 destinasi, dengan hasil yang sama pada kedua metode. Hasil pengujian manual dan sistem ditunjukkan pada Tabel 2 yang memperlihatkan bahwa destinasi yang dihasilkan sama antara manual dan sistem.

Tabel 2. Hasil pengujian manual dan sistem

Responden	Kategori	Budget input	Budget output	Destinasi manual	Destinasi sistem
A (Buruh berkeluarga)	Rendah	Rp500,000.00	Rp475,000.00	6	6
	Sedang	Rp1,200,000.00	Rp1,200,000.00	14	14
	Tinggi	Rp3,000,000.00	Rp1,050,000.00	15	15
B (Mahasiswa)	Rendah	Rp500,000.00	Rp450,000.00	9	9
	Sedang	Rp600,000.00	Rp570,000.00	11	11
	Tinggi	Rp1,500,000.00	Rp1,500,000.00	14	14
C (Buruh belum berkeluarga)	Rendah	Rp400,000.00	Rp380,000.00	6	6
	Sedang	Rp750,000.00	Rp750,000.00	13	13
	Tinggi	Rp1,000,000.00	Rp700,000.00	15	15

Pola kesesuaian ini juga konsisten pada Responden B dan Responden C dengan variasi *budget* yang berbeda. Dengan demikian, dapat disimpulkan bahwa algoritma *Greedy by Weight* telah diimplementasikan secara benar ke dalam sistem. Sistem mampu memberikan hasil rekomendasi yang sama dengan perhitungan manual, baik dari segi jumlah destinasi, total biaya, maupun keberagaman kategori. Contoh dari hasil pengujian sistem diperlihatkan pada Gambar 11 yang memperlihatkan bahwa wisatawan dengan *budget* Rp.500.000 yang berjumlah 5 orang dapat menghasilkan 6 destinasi wisata.



Gambar 11. Hasil pengujian sistem Rp.500.000 untuk 5 orang



3.3 Pembahasan

Pembahasan ini menguraikan bahwa penerapan algoritma *Greedy by Weight* dalam sistem rekomendasi destinasi wisata berbasis web bertujuan membantu wisatawan memperoleh kombinasi destinasi terbaik sesuai dengan keterbatasan *budget*. Proses dimulai ketika pengguna menginputkan *budget* total dan jumlah orang pada *form* sistem. Nilai *budget* tersebut kemudian dibagi dengan jumlah orang untuk menentukan batas pengeluaran per individu. Setelah itu, sistem mengambil data destinasi dari *database*, di mana setiap destinasi memiliki dua parameter utama, yaitu harga (*cost*) dan rating (*value*) sebagai indikator kualitas.

Selanjutnya, sistem menghitung rasio *value/cost* berdasarkan persamaan yang telah ditentukan. Rasio ini menjadi dasar penentuan prioritas destinasi, di mana destinasi dengan rasio tertinggi dipilih terlebih dahulu. Proses seleksi dilakukan secara bertahap menggunakan pendekatan *Greedy by Weight*, dengan aturan bahwa total biaya yang diakumulasi tidak boleh melebihi *budget* per orang. Selain fokus pada rasio terbaik, sistem juga mempertimbangkan keberagaman kategori destinasi seperti wisata laut, alam, dan budaya, agar hasil rekomendasi tidak monoton dan memberikan pengalaman wisata yang bervariasi bagi pengguna.

Implementasi algoritma ini dilakukan pada RekomendasiController menggunakan *framework* Laravel, dan hasil rekomendasi ditampilkan melalui *view* hasil_rekomendasi.blade.php dalam bentuk daftar destinasi yang dapat langsung dipesan. Lingkungan pengembangan memanfaatkan Laragon sebagai *web server* lokal yang mencakup PHP dan MySQL, sedangkan proses coding dilakukan melalui Visual Studio Code.

Pengujian sistem dilakukan terhadap tiga responden, masing-masing dengan tiga skenario *budget*, yaitu rendah, sedang, dan tinggi. Hasil pengujian menunjukkan bahwa *output* sistem identik dengan perhitungan manual menggunakan logika *Greedy by Weight*, baik dari sisi jumlah destinasi yang dipilih, total biaya yang tidak melebihi anggaran, urutan prioritas destinasi berdasarkan rasio, maupun keberagaman kategori wisata. Konsistensi ini membuktikan bahwa implementasi algoritma telah sesuai rancangan, mampu memberikan rekomendasi yang relevan, efisien, serta dapat menjadi solusi efektif bagi wisatawan dan penyedia jasa *travel* dalam pengelolaan pemesanan wisata berbasis *budget*.

4. KESIMPULAN

Penelitian ini menyimpulkan bahwa sistem pemesanan dan rekomendasi destinasi wisata berbasis *web* yang dibangun berhasil memenuhi kebutuhan wisatawan dengan keterbatasan *budget* melalui penerapan algoritma *Greedy by Weight*. Algoritma ini bekerja dengan menghitung rasio antara nilai (*value*) berupa *rating* destinasi dan biaya (*cost*) berupa harga tiket menggunakan persamaan (1), kemudian memilih destinasi dengan rasio tertinggi secara bertahap sambil memastikan total biaya tidak melebihi batas anggaran pengguna menggunakan persamaan (2). Selain mempertimbangkan efisiensi biaya, sistem juga mengakomodasi keberagaman kategori wisata seperti laut, alam, dan budaya, sehingga rekomendasi yang dihasilkan tidak monoton pada satu jenis wisata saja.

Hasil pengujian dengan berbagai skenario jumlah orang dan *budget* menunjukkan bahwa sistem mampu memberikan rekomendasi yang konsisten dan sesuai perhitungan manual menggunakan logika algoritma *Greedy by Weight*. Kesesuaian ini mencakup jumlah destinasi yang dipilih, urutan prioritas berdasarkan rasio, serta total biaya yang tetap berada dalam batas *budget*. Hal ini membuktikan bahwa implementasi algoritma dalam sistem berjalan sesuai dengan rancangan yang telah dibuat. Dengan demikian, sistem ini dapat menjadi solusi efektif untuk membantu wisatawan memilih kombinasi destinasi berkualitas sesuai anggaran, sekaligus menyediakan platform pemesanan jasa *travel* yang praktis dan relevan bagi penyedia layanan maupun pengguna akhir.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih yang sebesar-besarnya kepada dosen pembimbing yang telah memberikan arahan dan masukan selama proses penelitian hingga penyusunan artikel ini. Ucapan terima kasih juga ditujukan kepada Universitas Muhammadiyah Magelang yang telah memberikan fasilitas serta lingkungan akademik yang mendukung. Tidak lupa, penulis menghargai kontribusi rekan-rekan, keluarga, serta semua pihak yang telah membantu baik secara langsung maupun tidak langsung sehingga penelitian ini dapat terselesaikan dengan baik.

REFERENCES

- [1] A. Mun'im, "Penyempurnaan Pengukuran Kontribusi Pariwisata: Alternatif Percepatan Pertumbuhan Ekonomi Indonesia Improvement on the Measurement of Tourism Contribution: An Alternative to Accelerating Indonesia's Economic Growth," *J. Kepariwisataan Indones.*, vol. 16, no. 1, pp. 1–14, 2022.
- [2] A. Wicaksono, "New Normal Pariwisata Yogyakarta," *Kepariwisataan J. Ilm.*, vol. 14, no. 03, pp. 139–150, 2020, doi: 10.47256/kepariwisataan.v14i03.59.
- [3] Biro Pusat Statistik, "Perkembangan Pariwisata Daerah Isti mewa Yogyakarta, Oktober 2024," *Badan Pusat Statistik Provinsi Daerah Isti mewa*, 2024. <https://yogyakarta.bps.go.id/pressrelease/2024/12/02/1573/perkembangan-pariwisata-daerah-isti-mewa-yogyakarta-oktober-2024.html>
- [4] R. Setiawati and G. L. Pitalia, "Pemanfaatan Media Sosial sebagai Sarana Promosi Sektor Pariwisata," *KONSTELASI Konvergensi Teknol. dan Sist. Inf.*, vol. 3, no. 2, pp. 278–285, 2023, [Online]. Available: <https://ojs.uajy.ac.id/index.php/konstelasi/article/view/7255>

- [5] A. C. Ho, M. Lee, and I. B. Trisno, "Sistem Pemilihan Tempat Berwisata Berdasarkan Budget," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 6, pp. 1031–1037, 2022, doi: 10.32672/jnkti.v5i6.5472.
- [6] X. Gao and J. Zhang, "Greedy Algorithm Based Travel Route Planning under Multi - objective Decision Making," *Sci. J. Technol.*, vol. 7, no. 1, pp. 135–141, 2025, doi: <https://doi.org/10.54691/nx787h98>.
- [7] N. Widjiyati, "Penerapan Algoritma Greedy pada Query RDBMS Microsoft SQL Server sebagai Sistem Rekomendasi Pembelian Makanan Pokok," *Smart Comp Jurnalnya Orang Pint. Komput.*, vol. 11, no. 2, pp. 124–130, 2022, doi: 10.30591/smartcomp.v11i2.3507.
- [8] M. A. Ubaidillah and I. B. Gede Dwidasmara, "Tourism Recommendation System in Bali Using Topsis and Greedy Algorithm Methods," *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 8, no. 3, p. 277, 2020, doi: 10.24843/jlk.2020.v08.i03.p09.
- [9] Y. Z. Surentu, D. M. D. Warouw, and M. Rembang, "Pentingnya Website Sebagai Media Informasi Destinasi Wisata Di Dinas Kebudayaan Dan Pariwisata Kabupaten Minahasa," *Acta Diurna Komun.*, vol. 2, no. 4, pp. 1–17, 2020, [Online]. Available: <https://ejournal.unsat.ac.id/index.php/actadiurnakomunikasi/article/view/31117/29843>
- [10] Wijianto, "STRATEGI PENGEMBANGAN WISATA ALAMI DALAM ERA DIGITALISASI," *J. Ilm. EDUNOMIKA*, vol. 08, no. 02, 2024, [Online]. Available: https://jurnal.stie-aas.ac.id/index.php/jie/article/view/13101?utm_source=chatgpt.com
- [11] Yaya Suharya, Y. Herdiana, N. Indah Putri, and Z. Munawar, "Sistem Rekomendasi Untuk Toko Online Kecil Dan Menengah," *Tematik*, vol. 8, no. 2, pp. 176–185, 2021, doi: 10.38204/tematik.v8i2.683.
- [12] S. Devi Nurhayati and W. Widayani, "Sistem Rekomendasi Wisata Kuliner di Yogyakarta dengan Metode Item-Based Collaborative Filtering Yogyakarta Culinary Recommendation System with Item-Based Collaborative Filtering Method," *JACIS J. Autom. Comput. Inf. Syst.*, vol. 1, no. 2, pp. 55–63, 2021, [Online]. Available: <https://manganenakyog.my.id/>,
- [13] T. H. Cormen, *Introduction to Algorithms*. 2009.
- [14] S. Rahmawati, M. R. Adhani, N. Husna, R. Diyana, and Y. Astuti, "Implementasi Metode Prototype dalam Sistem Informasi Transaksi Pembelian Tiket Wisata pada Desa Wisata Brayut," pp. 4983–4992, 2024.
- [15] M. A. Sirait and A. Ichwani, "Sistem Informasi Wisata Berbasis Website Menggunakan Metode Prototype," *METHOMIKA J. Manaj. Inform. dan Komputerisasi Akunt.*, vol. 7, no. 1, pp. 34–40, 2023, doi: 10.46880/jmika.vol7no1.pp34-40.
- [16] A. A. K. Samsul Hadi, "Rancang Bangun Sistem Informasi Pemesanan Paket Tour and Travel Berbasis Web (Studi Kasus Gangsar Wisata Tour and Travel)," *J. PILAR Teknol. J. Ilm. Ilmu Tek.*, vol. 8, no. 2, pp. 28–38, 2023, doi: 10.33319/piltek.v8i2.148.
- [17] P. H. Sutanto, N. Lidwan, W. Ridwan, M. A. Lahat, and M. M. Al Atas, "Rancang Bangun Sistem Informasi Pemesanan Paket Wisata Berbasis Web (Studi Kasus pada Jams Tour)," *Jutisi J. Ilm. Tek. Inform. dan Sist. Inf.*, vol. 10, no. 1, p. 75, 2021, doi: 10.35889/jutisi.v10i1.582.
- [18] A. Nugroho, K. Prihandani, and R. Mayasari, "Rancang Bangun Sistem Pembelian E-Ticket Berbasis Website Dengan Konsep Server-Side Rendering Menggunakan Framework Next Js Pada Wisata Telaga Kusuma Jumantono," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 4, pp. 5771–5777, 2024, doi: 10.36040/jati.v8i4.9960.
- [19] R. Parlika, T. A. Nisaa', S. M. Ningrum, and B. A. Haque, "Studi Literatur Kekurangan Dan Kelebihan Pengujian Black Box," *Teknomatika*, vol. 10, no. 02, pp. 131–140, 2020.
- [20] R. Aditya, V. H. Pranatawijaya, and P. B. A. A. Putra, "Rancang Bangun Aplikasi Monitoring Kegiatan Menggunakan Metode Prototype," *J. Inf. Technol. Comput. Sci.*, vol. 1, no. 1, pp. 47–57, 2021.

