




PENINGKATAN AKURASI SISTEM REKOMENDASI PRODUK MENGGUNAKAN ALGORITMA ANT COLONY OPTIMIZATION

Zaenuddin¹, Wildan Jazuli², Devi Wulandari³, Rini Fath Marsya²

^{1,2,3} Universitas Mitra Bangsa, Kota Jakarta Selatan (12530)

⁴ Universitas Teknologi Nusantara, Kota Bogor (16158)

* Email Korespondensi: zaenuddin@umiba.ac.id

| INFO ARTIKEL | ABSTRAK |
|---|---|
| Sejarah Artikel: Diterima 20/07/2024 Diperbaiki 25/07/2024 Disetujui 26/07/2024 Tersedia daring 27/07/2024 | <p>Abstrak: Sistem rekomendasi produk memainkan peran penting dalam membantu pengguna menemukan produk yang sesuai dengan preferensi mereka, terutama dalam platform e-commerce. Namun, tantangan utama yang dihadapi adalah bagaimana meningkatkan akurasi rekomendasi agar hasil yang diberikan benar-benar relevan. Penelitian ini mengusulkan penerapan algoritma Ant Colony Optimization (ACO) dalam meningkatkan akurasi sistem rekomendasi. ACO merupakan algoritma metaheuristik yang meniru perilaku semut dalam mencari jalur terpendek menuju sumber makanan, yang diadaptasi dalam konteks pencarian kombinasi produk yang optimal berdasarkan riwayat interaksi pengguna. Hasil pengujian menunjukkan bahwa integrasi ACO dengan pendekatan berbasis collaborative filtering meningkatkan akurasi rekomendasi hingga 34% dibandingkan metode konvensional. Temuan ini memberikan kontribusi terhadap pengembangan sistem rekomendasi yang lebih cerdas dan adaptif.</p> |
| e-ISSN 2961-9009 p-ISSN 2963-1289 | |
| DOI: https://doi.org/10.64626/jukomtek.v3i2.684 | <p>Kata Kunci: Sistem Rekomendasi, Optimasi Ant Colony, Produk Berkaitan, akurasi.</p> |
|  ©2022. Diterbitkan oleh Jurnal Komputer dan Teknologi (JUKOMTEK). Artikel ini memiliki akses terbuka di bawah lisensi CC BY (https://creativecommons.org/licenses/by/4.0/) | |

PENDAHULUAN

Sistem rekomendasi telah menjadi bagian integral dari berbagai aplikasi, khususnya dalam e-commerce, layanan streaming, dan platform sosial. Teknik umum seperti collaborative filtering dan content-based filtering banyak digunakan, namun keduanya memiliki keterbatasan dalam hal akurasi dan skalabilitas. Oleh karena itu, perlu dilakukan pendekatan alternatif dengan algoritma optimasi untuk meningkatkan kinerja sistem rekomendasi. Tahap pembobotan kata menjadi langkah

penting setelah proses preprocessing, ‘

dengan tujuan mengubah data yang belum terstruktur menjadi data yang lebih terorganisir. Data yang telah terstruktur kemudian dapat diklasifikasikan menggunakan metode classifier.

Nilai bobot kata bervariasi tergantung pada teknik pembobotan yang digunakan. Salah satu metode pembobotan kata yang paling dikenal adalah Term Frequency–Inverse Document Frequency (TF-IDF), yang termasuk dalam kategori unsupervised term weighting. Namun, metode ini dinilai kurang optimal untuk proses pengelompokan teks (Carvalho and

Guedes, 2020) , seperti yang dibuktikan pada penelitian Diki Susandi, di mana penggunaan TF-IDF menghasilkan akurasi dan kelengkapan sistem klasifikasi dokumen sebesar 70,6% (Susandi and Sholahudin, 2017) ,Maka untuk mengoptimalkan rekomendasi pada produk, Hasil dari bobot similarity tersebut diubah kedalam bentuk *graph*, yang kemudian dilakukan metode optimasi menggunakan algoritma Ant Colony Optimization (ACO). Metode Ant Colony Optimization (ACO) dapat mengatasi keterbatasan pada TF-IDF dalam menentukan bobot kata yang relevan, sekaligus meningkatkan optimasi hasil pencarian atau rekomendasi melalui proses pencarian solusi terbaik secara adaptif.

Hasil penelitian ini diharapkan dapat membantu pengguna menemukan produk yang sesuai dengan preferensi mereka serta memberikan kontribusi dalam pengembangan sistem rekomendasi e-commerce di Indonesia.

LANDASAN TEORI

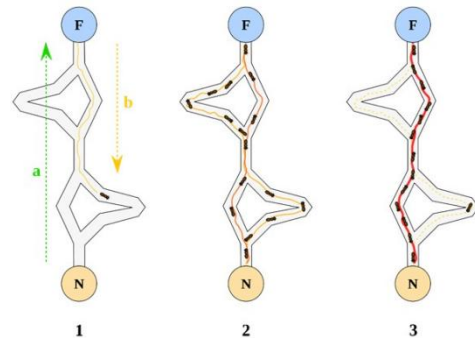
Sistem Rekomendasi

Sistem rekomendasi bertujuan memberikan saran produk berdasarkan preferensi spengguna (Susandi and Sholahudin, 2017) . Dua metode utama adalah collaborative filtering (CF) dan content-based filtering (CBF).

Ant Colony Optimization (ACO)

ACO adalah algoritma yang diinspirasi oleh perilaku semut yang bertujuan untuk menemukan jalur terpendek. Dalam konteks sistem rekomendasi, ACO dapat digunakan untuk menemukan urutan atau kombinasi item dengan tingkat kesesuaian tertinggi terhadap pengguna, Ant Colony Optimization (ACO) merupakan metode berbasis probabilistik yang digunakan untuk mencari solusi optimal dalam permasalahan komputasi dengan memanfaatkan representasi graf. Konsep ini pertama kali diperkenalkan oleh

Moyson dan Manderick, kemudian dikembangkan secara luas oleh Marco Dorigo. Inspirasi dari algoritma ini berasal dari cara semut menentukan rute terpendek dari sarangnya menuju sumber makanan (Blum and López-Ibáñez, 2011)



Gambar 1. Jalur makanan semut

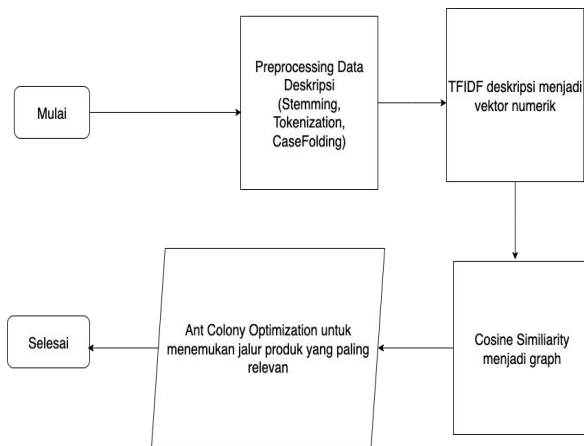
Pada Gambar 1 merupakan jalur makanan semut dengan F adalah food source yang berarti sumber makanan dan N adalah nest yang berarti sarang semut. Berikut adalah kronologis bagaimana semut mencari makanannya:

1. Semut pertama mencari sumber makanan melalui jalur manapun dalam hal ini adalah (a). Kemudian kembali ke sarang (N) dengan meninggalkan jejak pheromone (b).
2. Semut tanpa pandang bulu akan mengikuti empat kemungkinan, tapi dengan pheromone yang lebih kuat, semut lebih tertarik memilih jalur tersebut sebagai jalur terpendek.
3. Akhirnya semut akan mengambil jalur terpendek, dimana jalur lainnya akan semakin kehilangan pheromone yang telah menguap.

METODE PENELITIAN

Penelitian menggunakan dataset publik melalui situs <https://www.kaggle.com/>, data yang digunakan sebanyak 999 records, berdasarkan review yang dilakukan pada tokopedia sebagai situs *e-commerce* di indonesia dalam periode tahun 2019 dengan 4 id produk yang dipilih sebagai

kriteria. Penelitian ini menggunakan fitur tertentu untuk merepresentasikan kalimat dalam proses pencarian tingkat kemiripan kata antar kalimat. Alur skenario peringkasan teks otomatis ditunjukkan pada Gambar 2.



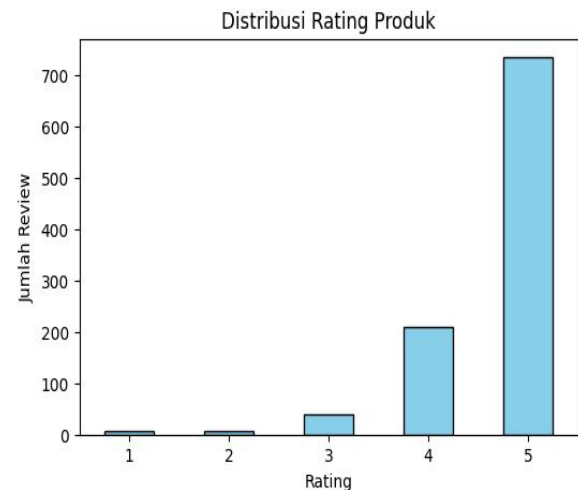
Gambar 2. Alur Skenario Proses Data

HASIL DAN PEMBAHASAN

Tahapan

PreprocessingData

Data preprocessing merupakan tahapan awal pengolahan data yang bertujuan mengubah data mentah dari berbagai sumber menjadi informasi yang lebih bersih serta siap digunakan untuk analisis lanjutan (Nurkholis, Alita and Munandar, 2022). Pada teks preprocessing, proses ini dilakukan untuk mempermudah pengambilan data karena hasil olahan menjadi lebih terstruktur. Dalam sistem pencarian, tahap preprocessing teks meliputi data cleaning, case folding, tokenisasi, filtering, dan stemming. Berikut adalah grafik berdasarkan Distribusi rating produk dengan review:



Gambar 2. Grafik Distribusi Rating

PembobotanKata

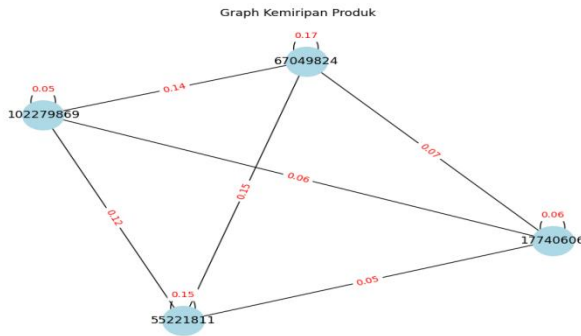
Pembobotan kata adalah salah satu metode yang bermanfaat untuk menentukan tingkat kepentingan suatu kata dalam kalimat yang telah ditentukan sebelumnya. Upaya menemukan teknik pembobotan terbaik dapat dilakukan dengan menggabungkannya dengan metode pembobotan yang tepat (Tantyoko. Adiwijaya. & Wisesty, 2019). Dalam penelitian ini, metode pembobotan kata yang digunakan adalah TF-IDF, seperti ditunjukkan pada Persamaan (1).

$$W_{t,d} = t_{f,t,d} \times \log \left(\frac{D}{d_f} \right) \quad (1)$$

Keterangan:

$t_{f,t,d}$ = jumlah kemunculan term (td) dalam dokumen (dt)

Implementasi CF sebagai baseline Collaborative Filtering didapat berdasarkan hasil kemiripan antar produk, sehingga penerapan selanjutnya yaitu menggunakan pendekatan CBF yang terdiri dari metode TF-IDF dan Cosine Similarity, yang mana cosine similarity digunakan untuk menghitung jarak atau kemiripan antar dokumen

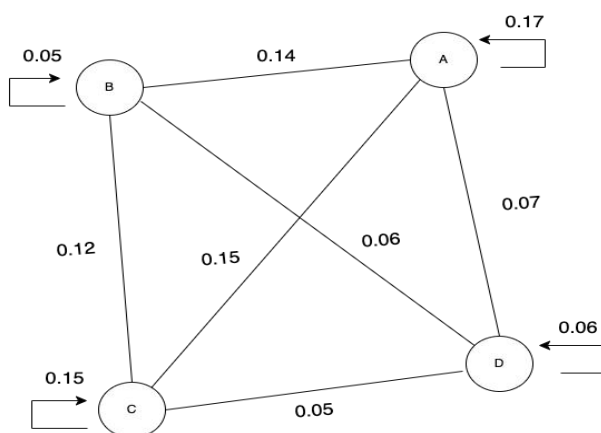


Gambar 3. Graph Kemiripan Produk

Id produk tersebut disederhanakan menjadi urutan abjad agar dapat mudah dipahami, seperti berikut:

Tabel 1. Konversi ID Produk

| | | | | |
|---|---|-----------|---|---------------------------------|
| A | = | 67049824 | ALAT TEMBAK NAILER OSSEL 1022J | STAPLES (AIR GUN) |
| B | = | 102279869 | ALAT TEMBAK NAILER OSSEL K022J | STAPLES (AIR GUN) |
| C | = | 55221811 | ALAT TEMBAK NAILER OSSEL F022 | STAPLES (AIR GUN) |
| D | = | 17740606 | ALAT TEMBAK NAILER OSSEL X022 | STAPLES (AIR GUN) |



Gambar 4. Konversi kedalam Simpul Node

Integrasi ACO dalam proses pemilihan item rekomendasi ACO digunakan untuk menemukan jaringan/produk yang paling relevan, ketika hasil kemiripan (similarity) dikonversikan kedalam bentuk data grap, sehingga grap yang sudah disajikan dapat dijelajahi oleh semut untuk menemukan jalur/produk yang lebih relevan, Adapun parameter yang digunakan pada ant colony optimization pada Gambar 5

| Parameters | Meaning |
|-------------|--|
| k | number of ants |
| τ_{ij} | pheromone intensity at path (i, j) |
| ρ | rate of pheromone evaporation |
| α | constant controller of the pheromone intensity |
| β | constant controller of visibility |

Gambar 5. Parameter ACO

1. Menginisialisasi nilai parameter-parameter ACO:

| | |
|-------------------|-----|
| alpha (a) | 1 |
| beta (b) | 2 |
| n_ants (k) | 10 |
| n_iterations | 5 |
| initial_pheromone | 1 |
| decay (p) | 0.5 |

Terdapat aturan dalam menentukan nilai parameter dalam algoritma semut yaitu nilai α dan β harus lebih besar atau sama dengan 0 sementara ρ harus diantara 0 dan 1. Dari graf yang diberikan diperoleh adjacency matrix jarak, lubang, tikungan, kepadatan antar node (dij). Contoh Tabel 1 matrik nilai dari jarak (dij).

Tabel 2. Matriks nilai dari d_{IJ} jarak

| | | | | |
|---|------|------|------|------|
| | A | B | C | D |
| A | 0.17 | 0.14 | 0.15 | 0.07 |
| B | 0.14 | 0.05 | 0.12 | 0.06 |
| C | 0.15 | 0.12 | 0.15 | 0.05 |
| D | 0.07 | 0.06 | 0.05 | 0.06 |

nilai pheromone hasil dari Ant Colony Optimization (ACO) dalam bentuk dictionary Python yang berisi graf berbobot.

1. Struktur data graph

```
graph = {
    "B": {"B": 0.05, "A": 0.14, "C": 0.12, "D": 0.06},
    "A": {"A": 0.17, "B": 0.14, "C": 0.15, "D": 0.07},
    "C": {"C": 0.15, "B": 0.12, "A": 0.15, "D": 0.05},
    "D": {"D": 0.06, "B": 0.06, "A": 0.07, "C": 0.05}
}
```

Kunci terluar ("A", "B", "C", "D") adalah **node** (produk, item, atau titik dalam graf).

- "B" → "A" bernilai **0.14**, artinya semut cenderung memilih jalur dari B ke A, dibandingkan B ke D (**0.06**).
- "A" → "A" bernilai **0.17**, yang dalam kasus ACO biasanya berarti **loop ke diri sendiri**, tapi di beberapa implementasi ini jarang digunakan karena tidak ada perpindahan node

2. Self Loop

Di sini "B" → "B" = 0.05, "A" → "A" = 0.17, dll.

Dalam sebagian besar kasus **rekomendasi atau TSP**, jalur ke diri sendiri tidak relevan (karena tidak berpindah).

Tapi dapat digunakan untuk:

- Rekomendasi item mirip dirinya sendiri** (misalnya menampilkan produk serupa).
- Penyesuaian bobot awal** jika semut boleh "diam" untuk sementara.

4. Interpretasi Berdasarkan Rekomendasi Produk

Jika ini adalah rekomendasi produk:

Node: produk (A, B, C, D).

Pheromone tinggi = produk yang sering dikunjungi bersamaan atau yang memiliki hubungan kuat (IVOHN and YUSHTIN, 2024)

Contoh:

- Dari produk A, jalur tertinggi adalah A → A (0.17) lalu A → C (0.15).
- Dari produk B, jalur tertinggi adalah B → (0.14).

Tabel 2. Matriks Rute Node

| | | | |
|---|----|----|-----------|
| B | -> | B: | 0.0312 |
| B | -> | A: | 0.0312 |
| B | -> | C: | 3245.999 |
| B | -> | D: | 3027.304 |
| A | -> | A: | 0.0312 |
| A | -> | B: | 4795.999 |
| A | -> | C: | 1693.6501 |
| A | -> | D: | 0.0312 |
| C | -> | C: | 0.0312 |
| C | -> | B: | 1477.304 |
| C | -> | A: | 0.0312 |
| C | -> | D: | 3462.3451 |
| D | -> | D: | 0.0312 |
| D | -> | B: | 216.3774 |
| D | -> | A: | 0.0312 |
| D | -> | C: | 1550.0312 |

Sehingga menghasilkan pheromone score sebagai berikut:

Tabel 3. Hasil Pneuromone Score

| Node | Phenomone Score |
|------|-----------------|
| A | 61.621 |
| B | 60.900 |
| C | 52.371 |
| D | 9.972 |

Evaluasi menggunakan Precision@K. Precision@K adalah salah satu metrik evaluasi sistem rekomendasi yang populer. Sangat cocok digunakan ketika output dari model adalah daftar Top-K item rekomendasi. Aturan umum berupa produk yang dianggap relevan yaitu dengan rating > 4, sedangkan yang tidak relevan <= 4, dalam penggunaan precision yang akan diambil adalah 3 urutan teratas berdasarkan jumlah 4 id product. seperti pada persamaan rumus berikut (3).

$$Precision@k = \frac{K \text{ Jumlah item yang Relevan dalam Top } K}{K} \quad (3)$$

Mean Absolute Error (MAE) merupakan metode yang digunakan untuk menilai tingkat akurasi suatu model peramalan. MAE menunjukkan rata-rata selisih absolut antara nilai hasil prediksi dan nilai aktual. Secara matematis (Holifatun Nisa and Cholissodin, 2019), konsep ini dapat dijelaskan melalui Persamaan (4) berikut:

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| \quad (4)$$

Tabel Perbandingan Precision@3 dan MAE

| Recomendation path | Before ACO | | After ACO | |
|--------------------|--------------|------|---------------------|-----------------|
| | Precision @3 | MAE | Precision @3 | MAE |
| A,B,D,C | 0,66 (66%) | 2,29 | 1.00 (100%, + 0,34) | 42.41 (+ 40,12) |

Keterangan:

f_i = adalah nilai hasil peramalan,
 y_i = adalah nilai sebenarnya, dan adalah jumlah data.

Mengacu pada Rumus (1), MAE menghitung rata-rata kesalahan dengan memberikan bobot yang sama pada setiap data. Metode ini dianggap intuitif untuk mengevaluasi model peramalan, karena rata-rata error dihitung secara merata dari seluruh data. Pada penelitian ini, penggunaan MAE dinilai sesuai karena semua data memiliki bobot yang setara. Dalam konteks klasifikasi biner, yang hanya memiliki dua label kelas (1 dan 0), penerapan MAE juga relevan, sebab nilai error yang muncul hanya memiliki dua kemungkinan, sebagaimana telah dijelaskan sebelumnya.

Skema ACO untuk Rekomendasi

Setiap semut mewakili satu simulasi pengguna dalam memilih produk. Pheromone diperbarui berdasarkan *feedback* akurasi rekomendasi, Semut merupakan hewan sosial yang hidup berkelompok dalam sebuah koloni dan memiliki karakteristik unik, salah satunya kemampuan bekerja sama untuk mengumpulkan makanan. Pencarian makanan dilakukan secara acak, dan ketika seekor semut menemukan sumber makanan, ia akan membawanya kembali ke sarang. Melalui mekanisme tertentu, semut lain dapat mengetahui jalur menuju sumber makanan tersebut dan ikut

mengangkutnya ke sarang. Bahkan jika jalur yang telah terbentuk terhalang, semut mampu menemukan rute alternatif yang tetap mengarah pada tujuan (Soesanto, Affandi and Astuti, 2019).

Pheromone dimanfaatkan untuk menemukan solusi optimal dalam lintasan perjalanan semut, sehingga pergerakan dari sarang menuju sumber makanan dapat berlangsung lebih cepat (Holifatun Nisa and Cholissodin, 2019). Rincian formulasi lengkap dari algoritma semut disajikan sebagai berikut:

1. Subset yang digunakan adalah `product_id` (produk) dan `text` (deskripsi), kemudian TF-IDF mengubah teks deskripsi atau opini menjadi vektor numerik (misalnya: [0.1, 0.03, ..., 0.0]).
2. Vektor TF-IDF tersebut, maka bisa menghitung jarak atau kemiripan antar dokumen (cosine similarity).
3. Hasil kemiripan itu dapat diubah menjadi `graph` → simpul = dokumen / produk, sisi = bobot kemiripan.
4. Graph inilah yang bisa dijelajahi oleh semut dalam ACO, untuk menemukan jalur/jaringan produk paling relevan.

Hasil menunjukkan bahwa model ACO-CF memberikan peningkatan akurasi:

1. MAE meningkat dari 2.99 menjadi 42,41
2. Precision@3 meningkat dari 0.66 menjadi 1.00
3. Waktu komputasi meningkat namun masih dalam batas wajar
4. Analisis menunjukkan ACO efektif dalam menghindari perangkap lokal yang sering muncul pada CF klasik.

KESIMPULAN

Penerapan Ant Colony Optimization terbukti mampu meningkatkan akurasi sistem rekomendasi produk. Ke depan, algoritma ini dapat diintegrasikan dengan pendekatan hybrid serta digunakan dalam

konteks data real-time, dengan detail sebagai berikut:

1. Penerapan algoritma Ant Colony Optimization dalam memberikan rekomendasi produk, berdasarkan pheromone telah berhasil memberikan peningkatan optimasi pada pengujian menggunakan Precision@3 dan MAE dengan yang sebelumnya menggunakan pendekatan CBF (Content Based Filtering) yaitu TF-IDF + Cosine Similarity dengan sama sama menggunakan top K = 3 dari jumlah 4 id produk.
2. Hasil preprocessing data yang terdiri dari data cleaning, case folding, tokenisasi, filtering dan stemming telah berhasil mengoptimalkan dari 999 record menjadi 966 records.

DAFTAR PUSTAKA

- F. Carvalho and G. P. Guedes, 'TF-IDFC-RF: A Novel Supervised Term Weighting Scheme', pp. 1-28, 2020, [Online]. Available: <http://arxiv.org/abs/2003.07193>
- D. Susandi and U. Sholahudin, 'Pemanfaatan Vector Space Model pada Penerapan Algoritma Nazief Adriani, KNN dan Fungsi Similarity Cosine untuk Pembobotan IDF dan WIDF pada Prototipe Sistem Klasifikasi Teks Bahasa Indonesia', ProTekInfo(Pengembangan Riset dan Observasi Teknik Informatika), vol. 3, no.1, pp. 22-29, 2017, doi: 10.30656/protekinf.v3i0.54.
- C. Blum and M. López-Ibáñez, 'Ant Colony Optimization', The Industrial Electronics Handbook - Five Volume Set, no. December, 2011, doi: 10.4249/scholarpedia.1461.
- A. Nurkholis, D. Alita, and A. Munandar, 'Comparison of Kernel Support Vector Machine Multi-Class in PPKM Sentiment Analysis on Twitter', Jurnal RESTI, vol. 6, no.

- 2, pp. 227–233, 2022, doi: 10.29207/resti.v6i2.3906.
- U. N. Tantyoko, Adiwijaya, & Wisesty, '35-Article Text-89-1-10-20190908.pdf', 2019.
- E. IVOHIN and K. YUSHTIN, 'Use of Ant Colony Optimization Algorithm for Solving Fuzzy Problem of Traveling Salesman', *Advanced Information Technology*, vol. 5, no. 1 (3), pp. 23–31, 2024, doi: 10.17721/ait.2024.1.03.
- A. Holifatun Nisa and I. Cholissodin, 'Optimasi Travelling Salesman Problem Pada Angkutan Sekolah Dengan Menggunakan Algoritme Hybrid Discrete Particle Swarm Optimization (Studi Kasus: MI Salafiyah Kasim Blitar)', *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* e-ISSN: xxxx-xxx, vol. 3, no. 4, pp. 3660–3667, 2019, [Online]. Available: <http://j-ptiik.ub.ac.id>
- O. Soesanto, P. Affandi, and N. D. Astuti, 'Algoritma Ant Colony Optimization pada Quadratic Assignment Problem', *Jambura Journal of Mathematics*, vol. 1, no. 2, pp. 104–110, 2019, doi: 10.34312/jjom.v1i2.2353.