

**SIMULASI ASSIGNMENT PROBLEM UNTUK DISTRIBUSI TUGAS OPTIMAL  
MENGGUNAKAN ALGORITMA HUNGARIAN DAN PYTHON****Khoiriayati Azmi**

Program Studi Matematika, Fakultas Matematika Dan Ilmu Pengetahuan Alam, Universitas Negeri Medan  
e-mail: khoiriayati\_a.4222230003@mhs.unimed.ac.id\*

**Dinda Kartika**

Program Studi Matematika, Fakultas Matematika Dan Ilmu Pengetahuan Alam, Universitas Negeri Medan  
e-mail: dindakartika@unimed.ac.id

**Abstrak**

Penelitian ini membahas ketidakseimbangan dalam pembagian tugas pengiriman dokumen di lingkungan pemerintahan dapat menimbulkan keterlambatan dan ketidakefisienan kerja. Penelitian ini bertujuan untuk mengoptimalkan pembagian tugas pengiriman dokumen secara adil dan efisien menggunakan algoritma Hungarian. Metode penelitian yang digunakan adalah pendekatan kuantitatif melalui simulasi *assignment problem*, di mana setiap kombinasi staf dan tugas dimodelkan dalam bentuk matriks beban kerja. Data dianalisis secara manual dan dengan implementasi *Python* untuk membandingkan hasil penugasan. Hasil penelitian menunjukkan bahwa pembagian tugas optimal menghasilkan total beban minimum sebesar 105 dokumen, dengan distribusi beban kerja yang merata untuk setiap staf. Implementasi algoritma Hungarian melalui *Python* menghasilkan hasil yang identik dengan perhitungan manual, namun dengan efisiensi waktu yang lebih baik. Temuan ini menunjukkan bahwa algoritma Hungarian efektif dan akurat dalam mengatasi permasalahan pembagian beban kerja di instansi pemerintahan, serta berpotensi untuk diotomatisasi dalam sistem informasi internal.

**Kata Kunci:** *assignment problem*, algoritma Hungarian, *Python*.

**Abstract**

*Imbalance in the task distribution of document delivery within government institutions can lead to delays and inefficiencies. This study aims to optimize the assignment of delivery tasks fairly and efficiently using the Hungarian algorithm. The research method employed a quantitative approach through simulation of the assignment problem, where each combination of staff and task was modeled in a workload cost matrix. Data were analyzed both manually and through Python implementation to compare the assignment results. The study found that the optimal task distribution resulted in a total minimum workload of 105 documents, with balanced assignments across all staff. The Hungarian algorithm implemented in Python produced identical results to the manual calculation, but with improved time efficiency. These findings demonstrate that the Hungarian algorithm is effective and accurate in addressing workload distribution problems in government institutions and has the potential to be automated within internal information systems.*

**Keywords:** *assignment problem*, Hungarian algorithm, *Python*.

**PENDAHULUAN**

Efisiensi dalam pengelolaan dan pengiriman dokumen di lingkungan pemerintahan merupakan faktor penting yang menentukan kelancaran birokrasi dan pelayanan publik. Dokumen menjadi bagian tidak terpisahkan dari proses administrasi, baik dalam bentuk fisik maupun digital, dan harus dikelola secara tepat, cepat, serta sistematis (Ayudia et al., 2022).

Namun, masih banyak instansi pemerintahan yang menjalankan proses pengiriman dokumen secara manual tanpa sistem penjadwalan dan pembagian kerja yang terstruktur. Hal ini menimbulkan permasalahan seperti ketidakseimbangan beban kerja antar staf pengantar, keterlambatan pengiriman dokumen penting, dan tidak optimalnya distribusi tugas (Zamili et al., 2023).

Kondisi tersebut menunjukkan bahwa pengelolaan dokumen tidak hanya bergantung pada sistem pengarsipan yang baik, tetapi juga pada mekanisme

pembagian kerja yang adil dan efisien. Dalam praktiknya, pembagian tugas pengiriman dokumen masih dilakukan secara subjektif atau berdasarkan kebiasaan, tanpa mempertimbangkan beban kerja aktual yang diterima masing-masing staf (Luthfiyyah & Takarini, 2024). Ketimpangan ini dapat menghambat efektivitas organisasi dan menurunkan produktivitas pegawai. Permasalahan utama yang dikaji adalah bagaimana membagi tugas pengiriman dokumen agar beban kerja antar staf seimbang dan efisien.

Untuk mengatasi permasalahan tersebut, dibutuhkan sistem pembagian tugas yang adil dan efisien. Salah satu pendekatan yang dapat digunakan adalah *Hungarian Algorithm*, sebuah metode optimasi yang mampu memecahkan persoalan penugasan (*assignment problem*) dengan meminimalkan total beban kerja (Hia, 2019). Dengan bantuan perangkat lunak *Python*, algoritma ini dapat diimplementasikan secara praktis untuk menghasilkan pembagian tugas yang terukur dan optimal. Tujuan dari penelitian ini adalah untuk menerapkan *Hungarian Algorithm* dalam membagi tugas pengiriman dokumen secara adil dan efisien di lingkungan pemerintahan.

Secara teoritis, masalah pembagian beban kerja dapat dimodelkan sebagai *assignment problem*, di mana sejumlah pekerja dialokasikan ke sejumlah tugas berdasarkan nilai beban kerja tertentu. *Hungarian Algorithm*, yang diperkenalkan oleh Harold Kuhn berdasarkan teori *Egerváry*, merupakan metode yang sering digunakan dalam berbagai konteks seperti *logistik* dan *manajemen produksi* untuk menyelesaikan masalah ini secara optimal dan efisien (Mardiani et al., 2020).

## KAJIAN TEORI

Beberapa konsep teori yang menjadi landasan utama dalam penelitian ini meliputi :

### **Model Linear Programming**

Dalam model *Linear Programming*, ada dua macam fungsi: fungsi tujuan (*objective function*) dan fungsi batasan (*constraint function*). Fungsi tujuan menggambarkan tujuan atau sasaran dalam permasalahan *linear programming* yang berkaitan dengan pengaturan yang optimal untuk memperoleh biaya minimal (Pratama & Kurniawan, 2020).

Fungsi tujuan:

$$\begin{aligned} \text{Min } Z = & c_1x_1 + c_2x_2 + \cdots + c_jx_j + \cdots \\ & + c_nx_n \end{aligned} \quad (1)$$

Fungsi batasan:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1j}x_j + a_{1n}x_n \leq h_1 \quad (2)$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2j}x_j + a_{2n}x_n \leq h_2 \quad (3)$$

⋮

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ij}x_j + a_{in}x_n \leq h_i \quad (4)$$

⋮

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mj}x_j + a_{mn}x_n \leq h_m \quad (5)$$

$$x_j \geq 0, j = 1, 2, \dots, n \quad (6)$$

### **Beban Kerja**

Pembagian beban kerja merupakan suatu proses strategis dalam manajemen organisasi yang bertujuan mengalokasikan tugas secara adil kepada setiap individu berdasarkan kapasitas dan waktu kerja yang tersedia (Widowati et al., 2024). Tujuan utamanya adalah untuk menciptakan efisiensi, pemerataan beban, serta memastikan bahwa sumber daya manusia dimanfaatkan secara optimal. Dalam konteks ini, pembagian kerja harus mempertimbangkan berbagai kendala operasional, seperti batas maksimum jam kerja, kebutuhan tenaga kerja minimum, dan jumlah tugas yang harus diselesaikan dalam periode tertentu (Suryaningrat et al., 2021).

Beban kerja yang tidak seimbang dapat menyebabkan kelelahan kerja, penurunan motivasi, dan ketidakefisienan organisasi. Oleh karena itu, diperlukan pendekatan yang dapat menjamin distribusi kerja yang tidak hanya merata, tetapi juga efisien dan sesuai dengan batasan-batasan sistem (Anggraini et al., 2024). Dalam banyak kasus, pembagian beban kerja dapat dimodelkan ke dalam bentuk persoalan penugasan, di mana sejumlah tenaga kerja harus dialokasikan untuk sejumlah tugas tertentu. Representasi ini memungkinkan penggunaan pendekatan kuantitatif untuk menghasilkan pembagian tugas yang optimal dan adil (Farhan Aly et al., 2022).

### **Assignment Problem**

Masalah penugasan adalah jenis pemrograman linear khusus di mana sumber dialokasikan kepada kegiatan atas dasar satu-satu (*one-to-one basis*). Oleh karena itu, setiap sumber atau penugasan ditugaskan secara khusus kepada satu kegiatan atau tujuan (Christwandy, 2024). Masalah penugasan adalah masalah mengatur objek untuk menyelesaikan tugas

dengan tujuan mengurangi beban kerja. Masalah penugasan merupakan kasus khusus dari model transportasi, dimana sejumlah sumber akan ditugaskan ketujuan sehingga total biaya minimal (Rahman & Wahyudin, 2021).

Dalam suatu bentuk beban  $c_{ij}$  sangat berkaitan terhadap staf  $i$  ( $i = 1, 2, \dots, m$ ) yang melaksanakan tugas  $j$  ( $j = 1, 2, \dots, n$ ), sehingga tujuannya pun untuk menentukan seperti apa tugas yang harus dikerjakan sehingga meminimalkan biaya (Sindar & Zendrato, 2019). Sehingga, bentuk umum model matematika penugasan :

Minimum

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (7)$$

Kendala

$$\sum_{j=1}^n x_{ij} = 1 ; j = 1, 2, \dots, n \quad (8)$$

$$\sum_{i=1}^m x_{ij} = 1 ; i = 1, 2, \dots, m \quad (9)$$

### Metode Hungarian

Metode Hungarian adalah salah satu pendekatan untuk mengatur alokasi optimal dari berbagai jenis sumber daya yang produktif untuk berbagai jenis pekerjaan. Ini juga digunakan untuk mengatur alokasi pekerjaan terhadap pusat-pusat kerja berdasarkan tingkat efisiensi masing-masing pekerjaan (Subaderi et al., 2024). Algoritma ini menghasilkan solusi terbaik, yang disebut sebagai solusi minimum. Syarat untuk menerapkan metode ini harus memenuhi beberapa hal, antara lain (Mardiani et al., 2020)

- a. Jumlah baris dan kolom harus sama.
- b. Setiap sumber hanya melakukan satu tugas.
- c. Jika jumlah sumber tidak seimbang, ditambahkan variabel *dummy*.
- d. Ada solusi optimal.

Tujuan dari metode Hungarian ini adalah untuk menempatkan sumber daya dan kegiatan dalam jumlah yang sama besar untuk meminimalkan beban. Adapun langkah-langkah metode Hungarian antara lain (Simatupang, 2021) :

- a) Menyusun ke dalam bentuk tabel penugasan berupa matriks.

- b) Pilih elemen di setiap baris yang memiliki nilai terkecil, kemudian kurangi nilai awal setiap baris.
- c) Jika kolom tidak memiliki elemen nol, pilih elemen di kolom yang memiliki nilai terkecil, dan kemudian kurangi nilai awal setiap kolom.
- d) Menutup semua elemen bernilai nol secara vertical maupun horizontal.
  - 1) Apabila jumlah garis = jumlah kolom, maka dikatakan optimal
  - 2) Sebaliknya, jika garis < banyaknya baris atau kolom, solusi belum optimal dan lakukan pengecekan ulang
- e) Jika tabel telah optimal, cari baris atau kolom yang hanya mengandung satu angka nol, dan pilih nilai itu.

### Python

*Python* merupakan salah satu bahasa pemrograman yang paling umum digunakan sebagai alat bantu dalam pengolahan data. Bahasa pemograman *python* memiliki banyak library yang dapat di akses. Penggunaan bahasa pemograman *python* salah satunya yaitu untuk menyelesaikan permasalahan optimasi terkhusus *linear programming*. Penggunaan *Python* dalam penelitian ini tidak hanya mempercepat proses komputasi dan validasi hasil, tetapi juga memungkinkan *flexibilitas* dalam mengolah data nyata atau simulasi. Pemrograman berbasis *Python* juga memungkinkan integrasi visualisasi dan analisis hasil untuk pengambilan keputusan yang lebih objektif (Zahara et al., 2024).

## METODE

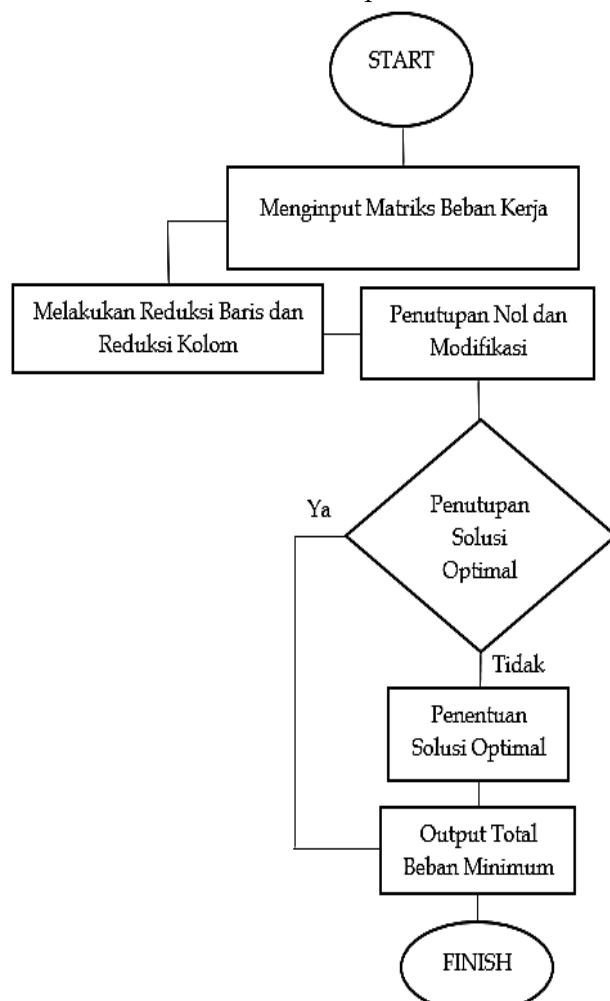
Dalam penelitian ini, metode kuantitatif dan simulasi digunakan untuk memodelkan masalah pembagian beban kerja dalam pengiriman dokumen. Tujuan dari penelitian ini adalah menemukan pembagian tugas yang optimal dan adil bagi staf pengantar dokumen, sehingga beban kerja dapat didistribusikan secara efisien.

Objek dalam penelitian ini adalah proses pembagian tugas pengiriman dokumen kepada staf di lingkungan instansi pemerintahan. Setiap kombinasi staf dan tugas memiliki nilai beban kerja tertentu yang disusun dalam bentuk matriks biaya. Nilai-nilai tersebut mewakili estimasi beban kerja berdasarkan kompleksitas tugas pengiriman. Data dikumpulkan secara simulatif melalui penyusunan

matriks beban kerja, berdasarkan observasi lapangan dengan staf pengantar dokumen terkait tingkat kesulitan, jarak tempuh, dan *urgensi* setiap tugas. Nilai skor dalam matriks mewakili estimasi beban kerja relatif antar tugas, yang dinyatakan dalam satuan dokumen setara kerja. Matriks ini menjadi dasar dalam proses analisis dan optimasi pembagian tugas.

Data yang telah disusun kemudian dianalisis menggunakan algoritma Hungarian yang diimplementasikan dengan bantuan bahasa pemrograman *Python*, yaitu dengan menggunakan fungsi *linear\_sum\_assignment* dari pustaka *scipy.optimize*. Proses analisis meliputi pemrosesan matriks biaya untuk menemukan kombinasi penugasan yang meminimalkan total beban kerja secara keseluruhan. Hasil dari proses ini berupa skema penugasan yang optimal, yang dapat dibandingkan dengan penugasan manual untuk menilai efisiensinya.

Berikut *Flowchart* dari metode penelitian:



Gambar 1. *Flowchart* Metode Hungarian

## HASIL DAN PEMBAHASAN

### PENYELESAIAN MANUAL DENGAN ALGORITMA HUNGARIAN

Berdasarkan permasalahan penugasan pada staf pemerintahan, setiap staf memiliki beban kerja yang berbeda-beda. Hal ini menyebabkan distribusi beban dalam proses pengantaran ke berbagai tujuan menjadi tidak merata antar staf. Berikut disajikan data beban yang dibawa oleh masing-masing staf pada saat proses pengantaran ke tujuan.

Tabel 1. Data Beban Masing-Masing Staf

Pekerja	Unit					
	Eko-nomi	Ta-pem	Ke-sra	Um-um	Orga-nisasi	Prok opim
Staf 1	19	20	26	23	21	14
Staf 2	21	18	20	26	24	22
Staf 3	16	17	25	19	20	21
Staf 4	25	21	22	19	22	20
Staf 5	25	22	21	26	20	24
Staf 6	15	22	23	21	27	19

Pihak pemerintahan berupaya menugaskan keenam staf tersebut ke masing-masing unit tujuan secara optimal, dengan mempertimbangkan total beban minimum. Tujuannya adalah untuk meningkatkan kesetaraan dalam distribusi beban kerja antar staf.

#### Langkah 1. Reduksi Baris

Proses ini dilakukan dengan cara mencari nilai terkecil di masing-masing baris, kemudian mengurangkan seluruh elemen dalam baris dengan nilai minimum. Hasil dari proses ini ditunjukkan pada Tabel 2.

Tabel 2. Langkah Penyesuaian 1

Pekerja	Unit					
	Eko-nomi	Ta-pem	Ke-sra	Um-um	Orga-nisasi	Prok opim
Staf 1	5	6	12	9	7	0
Staf 2	3	0	2	8	6	4
Staf 3	0	1	9	3	4	5
Staf 4	6	2	3	0	3	1
Staf 5	5	2	1	6	0	4
Staf 6	0	7	8	6	12	4

#### Langkah 2. Reduksi Kolom

Proses ini dilakukan dengan mencari nilai terkecil di masing-masing kolom, kemudian digunakan untuk mengurangi seluruh elemen kolom. Tujuan tahap ini adalah untuk menghasilkan elemen-elemen nol

tambahan dalam matriks agar penugasan dilakukan secara optimal. Hasilnya ditampilkan pada Tabel 3.

Tabel 3. Langkah Penyesuaian 2

Pekerja	Unit					
	Eko-nomi	Ta-pem	Ke-sra	Um-um	Orga-nisasi	Prok opim
Staf 1	5	6	11	9	7	0
Staf 2	3	0	1	8	6	4
Staf 3	0	1	8	3	4	5
Staf 4	6	2	2	0	3	1
Staf 5	5	2	0	6	0	4
Staf 6	0	7	7	6	12	4

#### Langkah 3. Penutupan Nol

Jika jumlah garis yang diperoleh sudah sama dengan jumlah baris atau kolom, maka solusi sudah optimal. Namun, jika jumlah garis yang diperoleh tidak sama, maka perlu dilakukan modifikasi.

Tabel 4. Langkah Penyesuaian 3

Pekerja	Unit					
	Eko-nomi	Ta-pem	Ke-sra	Um-um	Orga-nisasi	Prok opim
Staf 1	5	6	11	9	7	0
Staf 2	3	0	1	8	6	4
Staf 3	0	1	8	3	4	5
Staf 4	6	2	2	0	3	1
Staf 5	5	2	0	6	0	4
Staf 6	0	7	7	6	12	4

Dari tabel 4, jika jumlah garis < jumlah baris atau kolom, penarikan garis dilakukan lagi.

#### Langkah 4. Modifikasi

Proses ini mencari nilai terkecil dari elemen yang tidak terlewati oleh garis. Setelah itu, nilai ditambahkan pada elemen yang berada di persilangan garis dan dikurangkan dari elemen yang tidak melewati garis.

Tabel 5. Langkah Penyesuaian 4

Pekerja	Unit					
	Eko-nomi	Ta-pem	Ke-sra	Um-um	Orga-nisasi	Prok opim
Staf 1	5	5	10	8	7	0
Staf 2	4	0	1	8	6	5
Staf 3	0	0	7	2	3	5
Staf 4	7	2	2	0	3	2
Staf 5	6	2	0	6	0	5
Staf 6	0	6	6	5	11	4

Dari tabel 5, jika jumlah garis < jumlah baris atau kolom, penarikan garis dilakukan lagi.

#### Langkah 5. Modifikasi

Proses ini mencari nilai terkecil dari elemen yang tidak terlewati oleh garis. Nilai kemudian dikurangkan dari elemen yang tidak dilewati garis dan ditambahkan pada elemen yang berada di persilangan garis.

Tabel 6. Langkah Penyesuaian 5

Pekerja	Unit					
	Eko-nomi	Ta-pem	Ke-sra	Um-um	Orga-nisasi	Prok opim
Staf 1	6	6	10	8	7	0
Staf 2	4	0	0	7	5	4
Staf 3	0	0	6	1	2	4
Staf 4	8	3	2	0	3	2
Staf 5	7	3	0	6	0	5
Staf 6	0	6	5	4	10	4

Dari tabel 6 terlihat bahwa Jumlah garis = jumlah baris atau kolom sehingga solusi dapat dikatakan optimal dan proses dapat dihentikan.

Tabel 7. Hasil Penugasan Optimal

Pekerja	Unit					
	Eko-nomi	Ta-pem	Ke-sra	Um-um	Orga-nisasi	Prok opim
Staf 1	6	6	10	8	7	0
Staf 2	4	0	0	7	5	4
Staf 3	0	0	6	1	2	4
Staf 4	8	3	2	0	3	2
Staf 5	7	3	0	6	0	5
Staf 6	0	6	5	4	10	4

Staf 1 ke unit prokopim	= 14 dokumen
Staf 2 ke unit kesra	= 20 dokumen
Staf 3 ke unit tapem	= 17 dokumen
Staf 4 ke unit umum	= 19 dokumen
Staf 5 ke unit organisasi	= 20 dokumen
Staf 6 ke unit perekonomian	<u>= 15 dokumen</u>
	= 105 dokumen

#### IMPLEMENTASI PYTHON DALAM PROSES OPTIMASI

Untuk mendukung penyelesaian masalah penugasan secara efisien, algoritma Hungarian juga diimplementasikan menggunakan bahasa pemrograman agar proses komputasi dilakukan lebih cepat dan akurat.

### Langkah 1. Menyusun Matriks Biaya

Matriks berisi estimasi beban kerja masing-masing staf ke setiap unit tujuan, yang dinyatakan dalam bentuk *array* 2 dimensi:

```
[ ]: cost_matrix = np.array([
    [19, 20, 26, 23, 21, 14],
    [21, 18, 20, 26, 24, 22],
    [16, 17, 25, 19, 20, 21],
    [25, 21, 22, 19, 22, 20],
    [25, 22, 21, 26, 20, 24],
    [15, 22, 23, 21, 27, 19],
])
```

Gambar 2. Kode Python untuk Matriks Data

### Langkah 2. Reduksi Baris

Setiap baris dikurangi dengan nilai terkecil dalam baris tersebut untuk menghasilkan nol minimal di tiap baris:

```
[ ]: def reduksi_baris(matrix):
    return matrix - np.min(matrix, axis=1, keepdims=True)

reduced_baris = reduksi_baris(cost_matrix)
```

Gambar 3. Kode Python Reduksi Baris

### Langkah 3. Reduksi Kolom

Setelah reduksi baris, tiap kolom juga direduksi dengan mengurangi nilai terkecilnya:

```
[ ]: def reduksi_kolom(matrix):
    return matrix - np.min(matrix, axis=0, keepdims=True)

reduced_kolom = reduksi_kolom(reduced_baris)
```

Gambar 4. Kode Python Reduksi Kolom

**Langkah 4. Penutupan Nol dan Modifikasi Matrik**  
 Modifikasi dilakukan Jika jumlah garis > daripada jumlah baris atau kolom. Elemen yang memotong garis ditambahkan, dan elemen yang tidak tertutup oleh garis dikurangi dengan nilai terkecil:

```
[ ]: modified_matrix = modifikasi_matrix(reduced_kolom, baris_tertutup, kolom_tertutup)
```

Jika belum otimal, modifikasi ini diulang lagi dengan penutupan baru:

```
[ ]: modified_matrix_2 = modifikasi_matrix(modified_matrix, baris_tertutup_5, kolom_tertutup_5)
```

Gambar 5. Kode Python Penutupan dan Modifikasi

**Langkah 5. Menentukan penugasan optimal**  
 Langkah terakhir menggunakan fungsi *Linear\_sum\_assignment* dari pustaka *scipy.optimize* untuk kombinasi tugas yang meminimalkan total beban kerja:

```
[ ]: from scipy.optimize import linear_sum_assignment

row_ind, col_ind = linear_sum_assignment(cost_matrix)
total_beban = cost_matrix[row_ind, col_ind].sum()
```

Gambar 6. Kode Python Penugasan Optimal

### Ouput :

```
Langkah 5: Hasil Penugasan Optimal (Staf ke Unit Tujuan)
Staf 1 → Prokopim (Beban: 14)
Staf 2 → Kesra (Beban: 20)
Staf 3 → Tapem (Beban: 17)
Staf 4 → Umum (Beban: 19)
Staf 5 → Organisasi (Beban: 20)
Staf 6 → Perekonomian (Beban: 15)

Total Biaya Minimum Penugasan: 105
```

Gambar 7. Total Biaya Minimum

Hasil penugasan menggunakan *Python* menghasilkan total beban minimum sebesar 105 dokumen, sama seperti perhitungan manual. Ini menunjukkan bahwa implementasi algoritma Hungarian secara komputasi *valid* dan akurat.

Hasil ini menunjukkan konsistensi metode manual dan komputasi. Lebih jauh, pendekatan berbasis *Python* menawarkan potensi signifikan dalam otomatisasi pembagian tugas di lingkungan pemerintahan. Dengan integrasi sistem ini ke dalam aplikasi manajemen internal, instansi dapat mengurangi ketergantungan pada penilaian subjektif serta mempercepat proses pengambilan keputusan penugasan. Temuan ini menunjukkan bahwa pemanfaatan metode optimasi dalam praktik.

## PENUTUP

## SIMPULAN

Studi ini menunjukkan bahwa algoritma Hungarian dapat digunakan untuk mengoptimalkan pembagian tugas pengiriman dokumen di pemerintahan. Melalui simulasi *assignment problem*, metode ini terbukti mampu menghasilkan distribusi beban kerja yang adil dan efisien dengan total beban minimum sebesar 105 dokumen.

Implementasi secara manual maupun menggunakan *Python* memberikan hasil yang konsisten, namun penggunaan *Python* menawarkan efisiensi waktu dan potensi untuk pengembangan sistem otomatis. Temuan ini tidak hanya menjawab tujuan penelitian, tetapi juga membuka kemungkinan penerapan algoritma optimasi dalam skala

lebih luas pada pengelolaan tugas di instansi pemerintahan, sehingga proses administrasi dapat berjalan lebih objektif, efisien, dan profesional.

## DAFTAR PUSTAKA

- Anggraini, D. S., Syaripuddin, S., & A. Q. Q. (2024). Optimasi Penjadwalan Menggunakan Pemrograman Linier Integer pada Masalah Penjadwalan Perawat UPT Dinas Kesehatan Puskesmas Jonggon Jaya. *Jurnal Ilmiah Matematika*, 3(1), 54–60.
- Ayudia, R., Haryanto, H., & Lailani, E. O. (2022). Sistem Kearsipan Untuk Menunjang Efisiensi Kerja Di Bagian Tata Usaha Direktorat I Pada Inspektorat Jenderal Kemendikbudristek Ri. *Kompleksitas: Jurnal Ilmiah Manajemen, Organisasi Dan Bisnis*, 11(1), 70–79. <https://doi.org/10.56486/kompleksitas.vol11.no1.219>
- Christwandy, R. . (2024). Penerapan Metode Hungarian dalam Optimasi Penugasan Karyawan Tubeles Pre Assy PT.GT. 6(July), 412–422.
- Farhan Aly, M., Nu'man, A. H., & Bachtiar, I. (2022). Pengukuran Beban Kerja untuk Mengoptimalkan Pembagian Kerja pada Proses Produksi Custom Sparepart. *Bandung Conference Series: Industrial Engineering Science*, 2(1), 50–56. <https://doi.org/10.29313/bcgies.v2i1.1536>
- Hia, O. (2019). Implementasi Metode Hungarian Dalam Penugasan Karyawan (Studi Kasus: PT. Jefrindo Consultant). *Jurnal Riset Komputer (JURIKOM)*, 6(1), 85–92. <http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom>
- Luthfiyyah, K., & Takarini, N. (2024). Implementasi Aplikasi E-Surat Di Dinas Perindustrian Dan Tenaga Kerja Kota Surabaya. *INNOVATIVE:Journal Of Social Science Research*, 4, 5608–5617.
- Mardiani, Sari, Novita, Fanani, & Afandhi. (2020). Penerapan Metode Hungarian dalam Optimasi Penugasan Karyawan CV. Paksi Teladan. *Bulletin of Applied Industrial Engineering Theory*, 1(1), 1–3.
- Pratama, D. T., & Kurniawan, H. S. (2020). Optimasi Masalah Penugasan Menggunakan Metode Hungarian untuk Meminimalkan Waktu Produksi. *Bulletin of Applied Industrial Engineering Theory*, 1(1), 16–20. <http://jim.unindra.ac.id/index.php/baiet/article/view/277>
- Rahman, L. N., & Wahyudin, W. (2021). Optimalisasi Penugasan Karyawan Jasa Ekspedisi Menggunakan Metode Hungarian (Studi Kasus CV. Anteraja Cabang Mekarmukti). *Jurnal Serambi Engineering*, 6(3), 2120–2127. <https://doi.org/10.32672/jse.v6i3.3245>
- Simatupang, J. (2021). Teknik Penugasan Karyawan Vhida Ponsel dalam Penjualan Kartu Paket Internet Dengan Menggunakan Metode Hungarian. *Bulletin of Information Technology (BIT)*, 2(1), 1–12. <https://journal.fkpt.org/index.php/BIT/article/view/47>
- Sindar, A., & Zendrato, R. N. (2019). Optimasi Penugasan Pegawai Menggunakan Metode Hungarian. *Journal of Innovation Information Technology and Application (JINITA)*, 1(01), 16–24. <https://doi.org/10.35970/jinita.v1i01.93>
- Subaderi, S., Wahyu, C. S., & Suwondo, A. J. (2024). Penentuan Waktu Optimal Penugasan Karyawan Yang Tidak Berimbang Dengan Metode Hungarian (Studi kasus: Usaha Cuci Motor "Mas BEJO"). *Jurnal PASTI (Penelitian Dan Aplikasi Sistem Dan Teknik Industri)*, 17(3), 389. <https://doi.org/10.22441/pasti.2023.v17i3.010>
- Suryaningrat, I. B., Kuswardhani, N., & Hastuti, N. R. (2021). OPTIMALISASI BEBAN KERJA PADA INDUSTRI MAKANAN MENGGUNAKAN METODE WORKLOAD ANALYSIS (Studi Kasus pada UD. MR-Jember). *Jurnal Ilmiah Rekayasa Pertanian Dan Biosistem*, 9(2), 118–129. <https://doi.org/10.29303/jrbp.v9i2.219>
- Widowati, U., Ilham, A., Kom, S., & Kom, M. (2024). Optimasi penjadwalan kerja dengan kendala operasional. *Jurnal Komputer Dan Teknologi Informasi*, 1(2), 1–5. <https://doi.org/10.26714/jkti.v3i1.13957>
- Zahara, L., Rahmany, M., & Moulana, R. (2024). Penggunaan Phyton dalam Optimasi Bahan Baku pada Perusahaan X. *Jurnal Ilmiah Mahasiswa Pertanian*, 9, 428–433.
- Zamili, R., Koto, S., & Sipahutar, H. (2023). Pengaruh Pembagian Kerja Dan Beban Kerja Terhadap Kinerja Pada Balai Pemasyarakatan (Bapas) Kelas II Sibolga. *Jurnal Manajemen Dan Bisnis Ekonomi*, 1(1), 42–58. <https://doi.org/10.54066/jmbe-itb.v1i1.58>
- Masitoh, S. (2018). Blended Learning Berwawasan Literasi Digital Suatu Upaya Meningkatkan Kualitas Pembelajaran dan Membangun Generasi Emas 2045. *Proceedings of the ICECRS*, 1(3), 13–34. <https://doi.org/10.21070/picecrs.v1i3.1377>