**Contoh Soal:**

Sebuah pengelola tempat wisata ingin memprediksi keuntungan harian berdasarkan beberapa faktor:

* **Harga Tiket Masuk (h)**: Semakin tinggi harga tiket, potensi pendapatan per pengunjung meningkat, namun jumlah pengunjung mungkin menurun.
* **Jumlah Mainan Anak (m)**: Semakin banyak mainan anak yang tersedia, semakin menarik tempat wisata bagi keluarga, berpotensi meningkatkan jumlah pengunjung. Namun, penambahan mainan juga menambah biaya operasional.
* **Luas Wahana Air (w)**: Wahana air menjadi daya tarik tersendiri. Semakin luas wahana air, semakin banyak pengunjung yang tertarik, tetapi juga meningkatkan biaya perawatan.
* **Indeks Lokasi (l)**: Indeks ini merepresentasikan seberapa strategis lokasi tempat wisata (misalnya, dekat dengan pusat kota, akses mudah). Nilai indeks berkisar antara 0 hingga 1, di mana 1 adalah lokasi paling strategis. Lokasi yang strategis cenderung menarik lebih banyak pengunjung.
* **Ketersediaan Transportasi Umum (t)**: Variabel biner (1 jika transportasi umum mudah dijangkau, 0 jika sulit). Aksesibilitas transportasi umum dapat meningkatkan jumlah pengunjung.

**Persamaan Keuntungan (Contoh):**

Kita akan membuat persamaan keuntungan (K) sebagai fungsi dari variabel-variabel di atas. Ini hanyalah contoh, dan model yang sebenarnya mungkin lebih kompleks dan memerlukan data historis untuk penentuan koefisien yang akurat.

K(h,m,w,l,t)=(a⋅h+b⋅m​+c⋅ln(w+1)+d⋅l+e⋅t)⋅N(h,m,w,l,t)−C(m,w)

Di mana:

* (a, b, c, d, e) adalah koefisien yang menunjukkan pengaruh masing-masing faktor terhadap pendapatan per pengunjung.
* (N(h, m, w, l, t)) adalah fungsi yang memprediksi jumlah pengunjung harian, yang diasumsikan dipengaruhi oleh semua faktor. Contoh sederhana: N(h,m,w,l,t)=N0​−α⋅h2+β⋅m+γ⋅w+δ⋅l+ϵ⋅t di mana (N\_0) adalah jumlah pengunjung dasar, dan (\alpha, \beta, \gamma, \delta, \epsilon) adalah koefisien.
* (C(m, w)) adalah fungsi biaya operasional yang dipengaruhi oleh jumlah mainan dan luas wahana air. Contoh sederhana: C(m,w)=c1​⋅m+c2​⋅w di mana (c\_1) dan (c\_2) adalah biaya per unit mainan dan per unit luas wahana air.

**Tujuan:**

Kita ingin mencari kombinasi nilai (h, m, w, l, t) yang memaksimalkan keuntungan (K).

**Penggunaan Turunan:**

Untuk menemukan nilai optimal, kita dapat menggunakan konsep turunan parsial. Kita akan mencari titik di mana turunan parsial fungsi keuntungan (K) terhadap setiap variabel sama dengan nol:

∂h/∂K​=0,∂m/∂K​=0,∂w/∂K​=0,∂l/∂K​=0,∂t/∂K​=0

Menyelesaikan sistem persamaan turunan parsial ini secara analitis bisa sangat kompleks tergantung pada bentuk fungsi (N) dan (C). Oleh karena itu, kita akan menggunakan pendekatan numerik dengan Python.

**Kode Python untuk Menyelesaikan Kasus:**

Python

import numpy as np

import matplotlib.pyplot as plt

from scipy.optimize import minimize

# Definisi fungsi keuntungan (contoh)

def jumlah\_pengunjung(h, m, w, l, t, n0=1000, alpha=2, beta=5, gamma=3, delta=200, epsilon=50):

return n0 - alpha \* h\*\*2 + beta \* m + gamma \* w + delta \* l + epsilon \* t

def biaya\_operasional(m, w, c1=10, c2=5):

return c1 \* m + c2 \* w

def keuntungan(params):

h, m, w, l, t = params

if h < 0 or m < 0 or w < 0 or l < 0 or l > 1 or t not in [0, 1]:

return np.inf # Mengembalikan nilai tak hingga untuk parameter yang tidak valid

pendapatan\_per\_pengunjung = 1.5 \* h + 2 \* np.sqrt(m) + 0.8 \* np.log(w + 1) + 10 \* l + 20 \* t

jumlah = jumlah\_pengunjung(h, m, w, l, t)

biaya = biaya\_operasional(m, w)

return -(pendapatan\_per\_pengunjung \* jumlah - biaya) # Negatif karena minimize mencari nilai minimum

# Nilai awal untuk variabel

initial\_guess = [50, 10, 20, 0.7, 1]

# Batasan untuk variabel (opsional)

bounds = ((10, 100), (0, 50), (0, 100), (0, 1), (0, 1))

# Melakukan optimasi

result = minimize(keuntungan, initial\_guess, bounds=bounds, method='L-BFGS-B')

# Mendapatkan hasil optimal

if result.success:

h\_optimal, m\_optimal, w\_optimal, l\_optimal, t\_optimal = result.x

keuntungan\_optimal = -result.fun

jumlah\_pengunjung\_optimal = jumlah\_pengunjung(h\_optimal, m\_optimal, w\_optimal, l\_optimal, t\_optimal)

biaya\_optimal = biaya\_operasional(m\_optimal, w\_optimal)

print("Hasil Optimal:")

print(f"Harga Tiket Optimal: ${h\_optimal:.2f}")

print(f"Jumlah Mainan Anak Optimal: {int(m\_optimal)}")

print(f"Luas Wahana Air Optimal: {w\_optimal:.2f} unit")

print(f"Indeks Lokasi: {l\_optimal:.2f}")

print(f"Ketersediaan Transportasi Umum: {int(t\_optimal)}")

print(f"Jumlah Pengunjung Optimal: {jumlah\_pengunjung\_optimal:.2f}")

print(f"Biaya Operasional Optimal: ${biaya\_optimal:.2f}")

print(f"Keuntungan Harian Optimal: ${keuntungan\_optimal:.2f}")

else:

print("Optimasi gagal:", result.message)

# Visualisasi (Contoh pengaruh satu variabel terhadap keuntungan dengan variabel lain tetap)

h\_range = np.linspace(10, 100, 100)

keuntungan\_h = [-keuntungan([h, initial\_guess[1], initial\_guess[2], initial\_guess[3], initial\_guess[4]]) for h in h\_range]

plt.figure(figsize=(10, 6))

plt.plot(h\_range, keuntungan\_h)

plt.xlabel("Harga Tiket Masuk ($)")

plt.ylabel("Keuntungan Harian ($)")

plt.title("Pengaruh Harga Tiket terhadap Keuntungan (Variabel Lain Tetap)")

plt.grid(True)

plt.show()

# Visualisasi pengaruh jumlah mainan anak terhadap keuntungan

m\_range = np.linspace(0, 50, 51)

keuntungan\_m = [-keuntungan([initial\_guess[0], m, initial\_guess[2], initial\_guess[3], initial\_guess[4]]) for m in m\_range]

plt.figure(figsize=(10, 6))

plt.plot(m\_range, keuntungan\_m)

plt.xlabel("Jumlah Mainan Anak")

plt.ylabel("Keuntungan Harian ($)")

plt.title("Pengaruh Jumlah Mainan Anak terhadap Keuntungan (Variabel Lain Tetap)")

plt.grid(True)

plt.show()

# ... (Anda dapat menambahkan visualisasi untuk variabel lainnya)

**Penjelasan Kode:**

1. **Definisi Fungsi:**
   * jumlah\_pengunjung(): Memprediksi jumlah pengunjung berdasarkan input.
   * biaya\_operasional(): Menghitung biaya operasional.
   * keuntungan(): Menghitung keuntungan harian (negatif karena minimize mencari nilai minimum).
2. **Nilai Awal dan Batasan:**
   * initial\_guess: Tebakan awal untuk nilai setiap variabel optimasi.
   * bounds: Batasan nilai yang diperbolehkan untuk setiap variabel.
3. **Optimasi:**
   * minimize() dari scipy.optimize digunakan untuk mencari nilai variabel yang meminimalkan fungsi negatif keuntungan (yaitu, memaksimalkan keuntungan). Metode L-BFGS-B adalah algoritma optimasi yang umum digunakan untuk masalah dengan batasan.
4. **Hasil Optimal:**
   * Setelah optimasi berhasil, kode mencetak nilai optimal untuk setiap variabel, jumlah pengunjung, biaya operasional, dan keuntungan harian yang diprediksi.
5. **Visualisasi:**
   * Kode membuat contoh grafik yang menunjukkan bagaimana perubahan satu variabel (harga tiket dan jumlah mainan anak) mempengaruhi keuntungan, dengan asumsi variabel lain tetap pada nilai awal. Anda dapat menambahkan grafik serupa untuk variabel lainnya.

**Penting untuk Diperhatikan:**

* **Model Persamaan:** Persamaan keuntungan dan jumlah pengunjung di atas hanyalah contoh sederhana. Dalam kasus nyata, Anda perlu membangun model yang lebih akurat berdasarkan data historis, riset pasar, dan pemahaman yang lebih mendalam tentang faktor-faktor yang mempengaruhi bisnis tempat wisata Anda.
* **Koefisien:** Nilai koefisien dalam persamaan ((a, b, c, d, e, \alpha, \beta, \gamma, \delta, \epsilon, c\_1, c\_2)) sangat krusial dan harus ditentukan berdasarkan analisis data atau perkiraan yang masuk akal.
* **Optimasi:** Hasil optimasi akan sangat bergantung pada bentuk fungsi keuntungan dan batasan yang diberikan.
* **Visualisasi:** Grafik yang dihasilkan hanya menunjukkan pengaruh satu variabel pada satu waktu. Untuk pemahaman yang lebih komprehensif, Anda mungkin memerlukan visualisasi yang lebih kompleks atau analisis sensitivitas.

Dengan contoh ini, Anda dapat mulai membangun model prediksi keuntungan yang lebih sesuai dengan karakteristik unik tempat wisata Anda. Jangan ragu untuk menyesuaikan persamaan, koefisien, dan batasan sesuai dengan data dan pemahaman Anda.