|  |  |
| --- | --- |
| Nama Mahasiswa/NIM | 1.Adhitia Agung Febrian (301220025) |
| Judul Tugas | Buatlah pemodelan dan simulasi menggunakan python (gunakan google colab). Studi kasus yang  digunakan bebas tentang RNG, dan setiap mahasiswa berbeda studikasusnya: |
| Tahun | 2024 |

|  |  |
| --- | --- |
| **PEMODELAN DAN SIMULASI** | |
|  | **Teori Pendukung** |
| **A. Deskripsi Studi Kasus**  Simulasi ini bertujuan untuk menganalisis hasil turnamen catur di mana sejumlah pemain dengan rating berbeda bersaing satu sama lain. Setiap pertandingan menentukan pemenang berdasarkan rating dan actor acak. Dengan menggunakan Random Number Generator (RNG), kita dapat mensimulasikan berbagai hasil pertandingan untuk memahami dinamika turnamen catur.  **B. Persamaan/Rumus-rumus yang digunakan**  Probabilitas menang:  *P*(Menang)  Di mana *P*(Menang) adalah probabilitas kemenangan seorang pemain berdasarkan rating mereka.  **C. Pemodelan Menggunakan Python**     |  | | --- | | *import random*  *import matplotlib.pyplot as plt*  *# Kelas untuk pemain*  *class Player:*  *def \_\_init\_\_(self, name, rating):*  *self.name = name*  *self.rating = rating*  *self.points = 0*  *# Fungsi untuk menentukan pemenang*  *def match(player1, player2):*  *outcome = random.random()*  *total\_rating = player1.rating + player2.rating*  *prob\_player1 = player1.rating / total\_rating*    *if outcome < prob\_player1:*  *player1.points += 1*  *return player1.name*  *else:*  *player2.points += 1*  *return player2.name*  *# Simulasi turnamen*  *def simulate\_tournament(players):*  *round\_winners = []*    *while len(players) > 1:*  *random.shuffle(players)*  *round\_winner = []*    *for i in range(0, len(players), 2):*  *if i + 1 < len(players):*  *winner = match(players[i], players[i + 1])*  *round\_winner.append(players[i] if winner == players[i].name else players[i + 1])*    *players = round\_winner*  *round\_winners.append(players)*    *return players[0]*  *# Menjalankan simulasi*  *def main():*  *num\_players = 16*  *players = [Player(f'Player {i + 1}', random.randint(1000, 2400)) for i in range(num\_players)]*  *champion = simulate\_tournament(players)*  *print(f"Champion: {champion.name} with {champion.points} points")*  *plt.bar([p.name for p in players], [p.points for p in players])*  *plt.xlabel('Players')*  *plt.ylabel('Points')*  *plt.title('Tournament Results')*  *plt.xticks(rotation=45)*  *plt.show()*  *# Menjalankan program*  *main()* |   **D. Penjelasan Koding**   1. Player Class: Kelas ini menyimpan informasi mengenai nama, rating, dan poin setiap pemain. 2. match Function: Fungsi ini menentukan pemenang antara dua pemain berdasarkan rating dan RNG. Pemenang mendapat satu poin. 3. simulate\_tournament Function: Fungsi ini mengatur turnamen dengan mengacak urutan pemain dan melakukan pertandingan hingga satu pemain tersisa. 4. main Function: Membuat daftar pemain, menjalankan turnamen, dan menampilkan hasil serta grafik.   **E. Tampilan Grafik Pemodelan dan Simulasi**    **F. Penjelasan Grafik**  Grafik batang menunjukkan total poin yang diperoleh setiap pemain setelah turnamen. Pemain dengan lebih banyak poin menunjukkan kinerja yang lebih baik selama turnamen. Grafik ini membantu dalam menganalisis bagaimana rating dan faktor acak memengaruhi hasil turnamen. | |
|  | **Alat Dan Bahan** |
| * Hardware (Laptop) * Pemrograman python * Google Collab * Algoritma RNG | |
|  | **Tutorial** |
| Tutorial Simulasi Turnamen Catur  1. Persiapan Lingkungan   * Akses Google Colab:   + Buka browser Anda dan kunjungi [Google Colab](https://colab.research.google.com/).   2. Buat Notebook Baru   * Klik pada "File" > "New Notebook" untuk membuat notebook baru.   3. Salin dan tempel Kode simulasi  import random  import matplotlib.pyplot as plt  # Kelas untuk pemain  class Player:  def \_\_init\_\_(self, name, rating):  self.name = name  self.rating = rating  self.points = 0  # Fungsi untuk menentukan pemenang  def match(player1, player2):  # Menggunakan RNG untuk menentukan pemenang dengan mempertimbangkan rating  outcome = random.random()  total\_rating = player1.rating + player2.rating    # Probabilitas menang berdasarkan rating  prob\_player1 = player1.rating / total\_rating    if outcome < prob\_player1:  player1.points += 1  return player1.name  else:  player2.points += 1  return player2.name  # Simulasi turnamen  def simulate\_tournament(players):  round\_winners = []    while len(players) > 1:  random.shuffle(players) # Acak urutan pemain  round\_winner = []    # Pertandingan antar pasangan pemain  for i in range(0, len(players), 2):  if i + 1 < len(players):  winner = match(players[i], players[i + 1])  round\_winner.append(players[i] if winner == players[i].name else players[i + 1])    players = round\_winner # Pemain yang menang maju ke putaran berikutnya  round\_winners.append(players)    return players[0] # Pemain terakhir yang menang  # Menjalankan simulasi  def main():  # Membuat daftar pemain dengan rating acak  num\_players = 16  players = [Player(f'Player {i + 1}', random.randint(1000, 2400)) for i in range(num\_players)]  # Menjalankan turnamen  champion = simulate\_tournament(players)  print(f"Champion: {champion.name} with {champion.points} points")  # Visualisasi hasil  plt.bar([p.name for p in players], [p.points for p in players])  plt.xlabel('Players')  plt.ylabel('Points')  plt.title('Tournament Results')  plt.xticks(rotation=45)  plt.show()  # Menjalankan program  main()  **4.Menjalankan Kode**  Klik tombol "Run" (atau tekan Shift + Enter) untuk menjalankan sel yang berisi kode. Dan hasilnya akan muncul. | |
|  | **Link Video Tutorial** |
|  | |
|  | **Referensi:** |
|  | |