Laporan Praktikum

Algoritme Pemrograman  
Eksekusi Kondisional dan Perulangan Dengan While

short line



Asisten :

| Russel Amanda Simangunsong | 04211075 |
| --- | --- |
| Muhammad Fadillah | 10201059 |

Disusun Oleh :

| **Clarinta Febrina** | **13231012** |
| --- | --- |
| Ade Ilham Pratama | 06231001 |
| Albarthina Novirna | 06231001 |
| Rezky Pratama | 06231078 |
| Ajwar Alwan | 13231003 |
|  |  |
|  |  |

12 Februari 2024

# Dasar Teori

1. **Eksekusi Kondisional**

Eksekusi kondisional adalah cara pemrograman yang memungkinkan komputer untuk mengambil tindakan berdasarkan kondisi tertentu dan meningkatkan tingkat kecerdasannya. Ini adalah elemen penting dalam membuat komputer tampak lebih cerdas, dan kuncinya terletak pada penggunaan pertanyaan if. Pertanyaan if adalah kata kunci yang menunjukkan niat kita untuk melakukan tindakan berdasarkan kondisi tertentu, dan hal ini membuat kode lebih mudah dibaca. Dalam konteks ini, pengambilan keputusan terjadi, yaitu kemampuan untuk mengantisipasi kondisi (melalui pertanyaan if) saat program dijalankan dan menentukan tindakan yang akan diambil berdasarkan kondisi tersebut. Di dalam Python, struktur pengambilan keputusan melibatkan statement if, else, dan elif. Kondisi if digunakan untuk menentukan apakah suatu kondisi bernilai TRUE atau FALSE. Oleh karena itu, saat menulis program, kita harus memutuskan tindakan yang akan diambil berdasarkan apakah hasilnya BENAR atau SALAH, dan sebaliknya. Untuk membuat program yang berguna, kita perlu memiliki kemampuan untuk memeriksa kondisi dan mengubah perilaku program sesuai dengan kondisi tersebut. Pernyataan kondisional memberi kita kemampuan ini. Bentuk paling sederhana adalah pernyataan :

Jika X > 0 :

Print (‘nilai X Positif’)

Ekspresi boolean setelah pernyataan if disebut kondisi. Akhiri pernyataan if dengan karakter titik dua (:) dan garis setelah pernyataan if di indentasi. Contoh programnya:

X = 5 if X

< 10 :

print(‘Lebih kecil’) if X > 20 :

Print (‘Lebih besar’)

Print (‘Selesai’)

**2. While**

Perulangan While merupakan metode pengulangan kode program di mana blok kode akan terus dieksekusi selama kondisi tertentu tetap terpenuhi. Perulangan While bersifat tidak pasti atau indefinite, yang berarti bahwa ia akan terus berjalan tanpa batasan waktu tertentu, dan perulangan hanya akan berhenti jika kondisi tertentu tidak lagi terpenuhi. Contohnya:  
i = int(input("Masukkan Bilangan : " ))

while(i <= 5):

print(f"Ini adalah angka {i}")

i = i + 1

Dalam perulangan while, ada beberapa tanda yang digunakan, yaitu :

1. <= : untuk masukan perulangan kurang dari sama dengan
2. >= : untuk masukan perulangan lebih dari sama dengan
3. = = : untuk masukan perulangan sama
4. != : untuk memfaktorial kan
5. - = : untuk perulangan yang dikurangkan 6) + = : untuk perulangan yang ditambahkan.

Adapun penggunaan end = “ ” yang digunakan untuk menampilkan perulangan mendatar.

Contohnya:

i = int(input("Masukkan Bilangan : " ))

while(i <= 5): print(i, end = “ ”)

i = i + 2

# *Source Code : Modul ‘Eksekusi Kondisional’*

| No.1 | Pingsuit.py |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49 | import random  import sys  # Tambahkan warna teks  class color:  PURPLE = '\033[95m'  CYAN = '\033[96m'  DARKCYAN = '\033[36m'  BLUE = '\033[94m'  GREEN = '\033[92m'  YELLOW = '\033[93m'  RED = '\033[91m'  BOLD = '\033[1m'  UNDERLINE = '\033[4m'  END = '\033[0m'  print(color.BOLD + ".:: Permainan Suit/Pingsut ::." + color.END)  print("1. " + color.BLUE + "Jempol (Gajah)" + color.END)  print("2. " + color.YELLOW + "Telunjuk (Manusia)" + color.END)  print("3. " + color.GREEN + "Kelingking (Semut)" + color.END)  pil = int(input(color.BOLD + "Pilihan anda ? " + color.END))  if(pil < 1 or pil > 3):  sys.exit(color.RED + "Masukkan pilihan yang benar!! Pilihan antara 1 - 3." + color.END)  # Pilihan komputer  kom = random.randint(1, 3)  if(kom == 1):  if(pil == 1):  print(color.CYAN + "Sama-sama Gajah! sesama gajah saling membantu..." + color.END)  if(pil == 2):  print(color.RED + "Diinjek gajah.. kamu kalah!" + color.END)  if(pil == 3):  print(color.GREEN + "Kamu gigit gajah, kamu menang!" + color.END)  elif(kom == 2):  if(pil == 1):  print(color.GREEN + "Kamu abis nginjek manusia, kamu menang!" + color.END)  if(pil == 2):  print(color.CYAN + "Sama-sama Manusia! Jangan berantem lah..." + color.END)  if(pil == 3):  print(color.RED + "Kamu dibunuh manusia, kamu kalah!" + color.END)  elif(kom == 3):  if(pil == 1):  print(color.RED + "Kamu abis dikerjain sama semut, kamu kalah!" + color.END)  if(pil == 2):  print(color.GREEN + "Kamu gak sengaja injek semut, kamu menang!" + color.END)  if(pil == 3):  print(color.CYAN + "Sesama semut saling membahu..!" + color.END) |

| No.2 | PenerjemahProtein.py |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | print("-----Program Penerjemah Protein-----")  k = input("Masukkan Kodon = ")  if (k == "AUG"):  print("Protein Kodon AUG Adalah Methionine")  elif (k == "UUU" or k == "UUC"):  print(f"Protein Kodon {k} adalah Phenylalanine")  elif (k == "UUA" or k == "UUG"):  print(f"Protein Kodon {k} adalah Leucine")  elif (k == "UCU" or k == "UUC" or "UCG" or k == "UCA"):  print(f"Protein Kodon {k} adalah Serine")  elif (k == "UAU" or k == "UAC"):  print(f"Protein Kodon {k} adalah Tyrosine")  elif (k == "UGC" or k == "UGU"):  print(f"Protein Kodon {k} adalah Cysteine")  elif (k == "UGG"):  print(f"Protein Kodon UGG adalah Tryptophan") |

| No.3 | Menghitung\_Akar-akar\_Persamaan\_Kuadrat.py |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | print('-----Menghitung Akar Persamaan Kuadrat-----')  a = float(input("Nilai A : "))  if a == 0 :  print('Bukan Persamaan Kuadrat')  print('Lanjutkan Nilai B untuk Menentukan Nilai X')  b = float(input("Nilai B : "))  c = float(input("Nilai C : "))  d = b \*\* 2 - 4 \* a \* c  if d<0:  print("Merupakan Akar Imajiner")  elif (d == 0):  x = -b / (2 \* a)  print("Akar Persamaan Kuadratnya adalah",str (x))  else :  x1 = (-b -d \*\* (float(1)/2))/(2\*a)  x2 = (-b -d \*\* (float(1)/2))/(2\*a)  print("Akar dari persamaan adalah",str(x1),str(x2)) |

# *Source Code : Modul ‘Perulangan dengan While’*

| No.1 | Mencetak\_Bilangan\_Ganjil.py |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | print("-----Program Mencetak Bilangan Ganjil-----")  batas = int(input("Masukkan batas angka : "))  x = 1  if(batas % 2 == 1):  while(x <= batas):  print(batas, end =" ")  batas -= 2  elif(batas % 2 == 0):  batas -= 1  while(x <= batas):  print(batas, end =" ")  batas -= 2 |

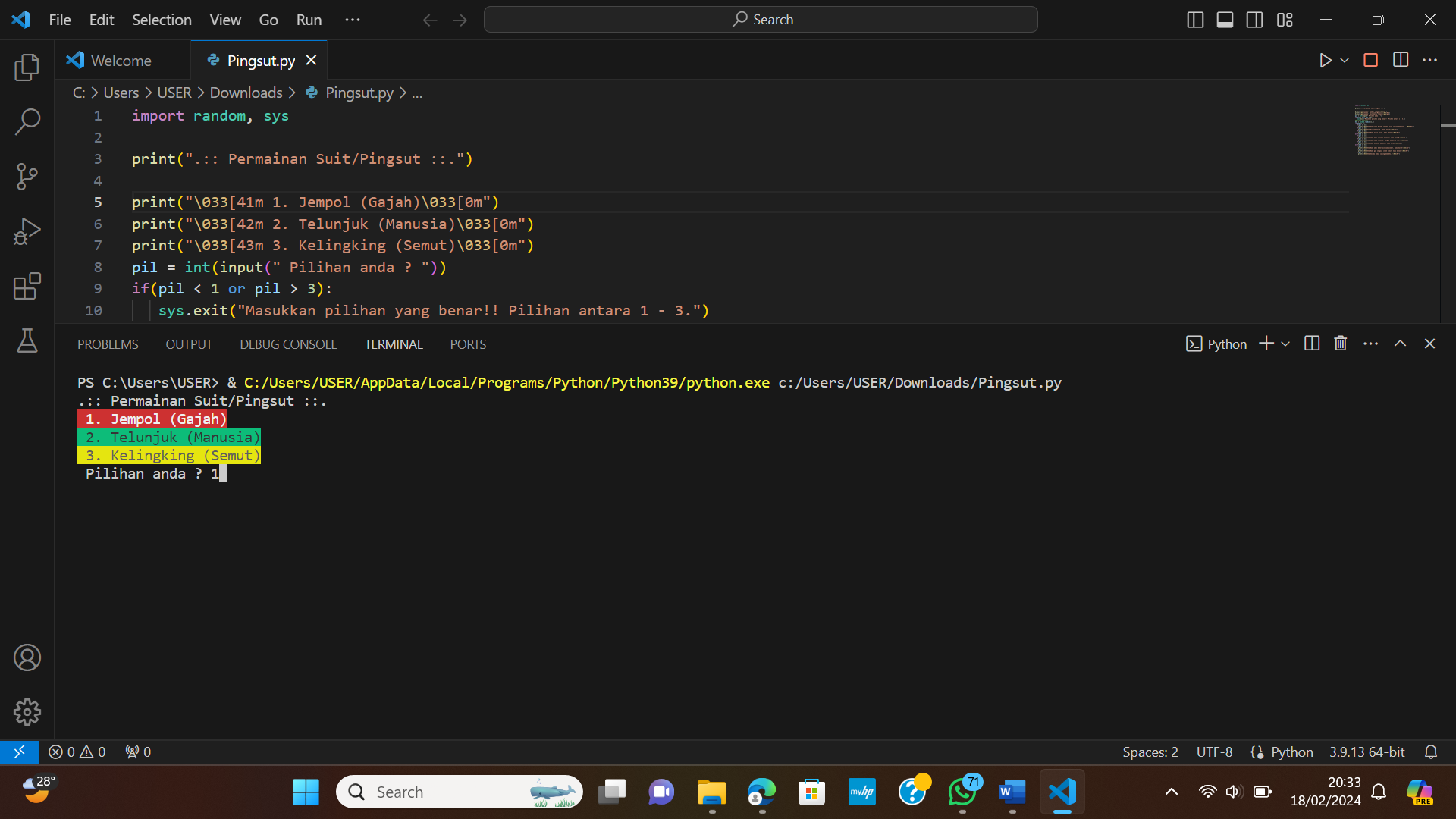
| No.2 | LuasdanKeliling.py |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | while True:  print("Menu:")  print("1. Persegi")  print("2. Persegi Panjang")  print("3. Segitiga")  print("4. Keluar")    choice = input("Masukkan pilihan (1/2/3/4): ")  if choice == '1':  sisi = float(input("Masukkan panjang sisi persegi: "))  luas = sisi \* sisi  keliling = 4 \* sisi  print("Luas persegi:", luas)  print("Keliling persegi:", keliling)  elif choice == '2':  panjang = float(input("Masukkan panjang persegi panjang: "))  lebar = float(input("Masukkan lebar persegi panjang: "))  luas = panjang \* lebar  keliling = 2 \* (panjang + lebar)  print("Luas persegi panjang:", luas)  print("Keliling persegi panjang:", keliling)  elif choice == '3':  alas = float(input("Masukkan alas segitiga: "))  tinggi = float(input("Masukkan tinggi segitiga: "))  sisi1 = float(input("Masukkan panjang sisi pertama segitiga: "))  sisi2 = float(input("Masukkan panjang sisi kedua segitiga: "))  luas = 0.5 \* alas \* tinggi  keliling = alas + sisi1 + sisi2  print("Luas segitiga:", luas)  print("Keliling segitiga:", keliling)  elif choice == '4':  print("Terima kasih telah menggunakan program ini.")  break  else:  print("Pilihan tidak valid. Silakan masukkan pilihan yang benar.") |

| No.3 | Mencari\_Nilai\_Faktorial.py |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | print("-----Program Mencari Nilai Faktorial-----")  n = int(input("Masukkan nilai n : "))  faktorial = 1  print(n,"! =", end=" ")  while n > 0:  faktorial \*= n  if n == 1:  print(n, end =" ")  else:  print(n, end =" x ")  n -= 1  print(f"= {faktorial}") |

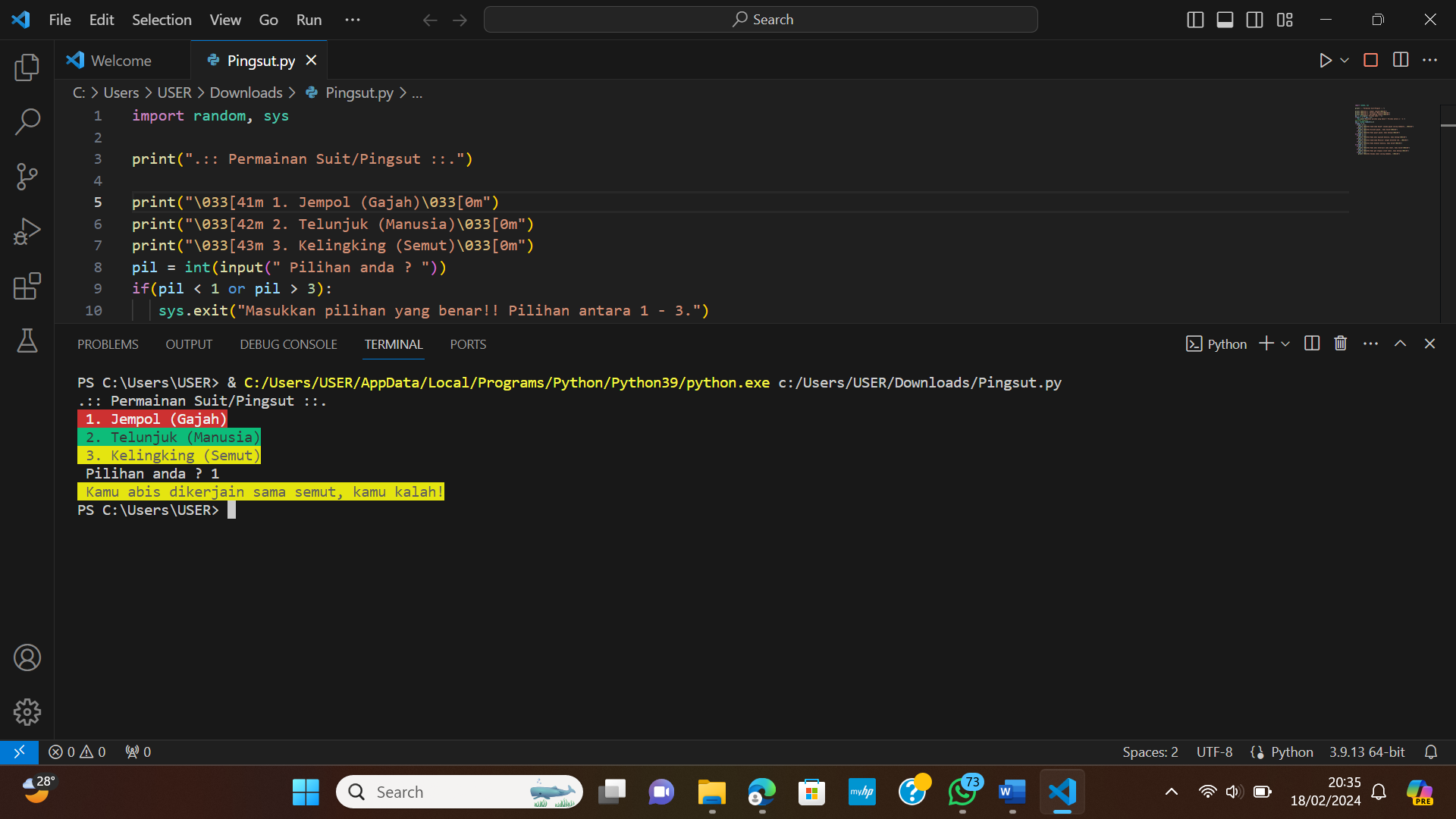
*Screenshot Modul ‘Eksekusi Kondisional’*

*Nomor 1 : Pingsut.py*

Pada Awal Tampilan Output, pengguna diminta untuk memasukkan pilihan suit.

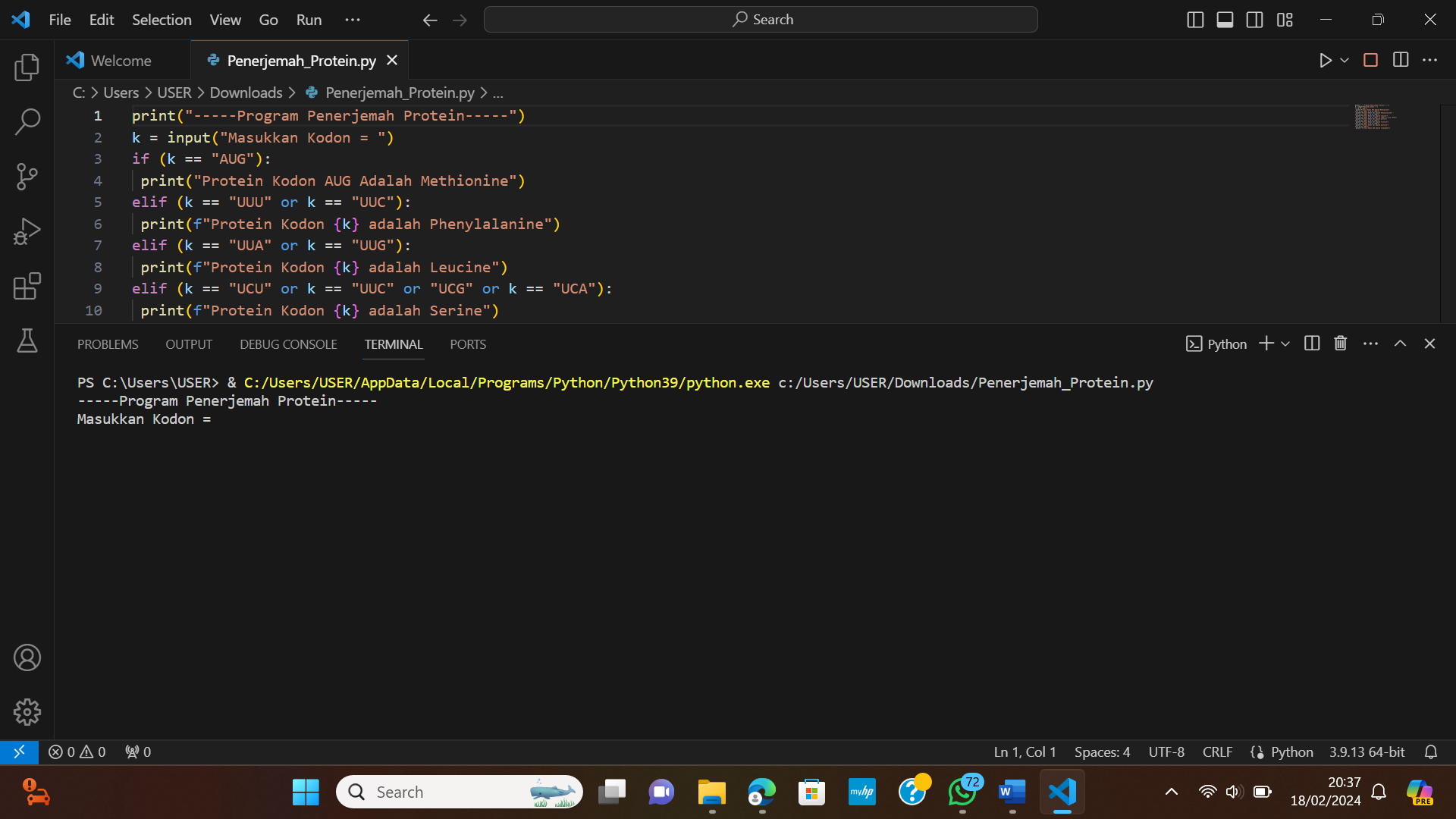


Setelah pengguna memasukkan pilihan suit, maka program akan menghasilkan output.

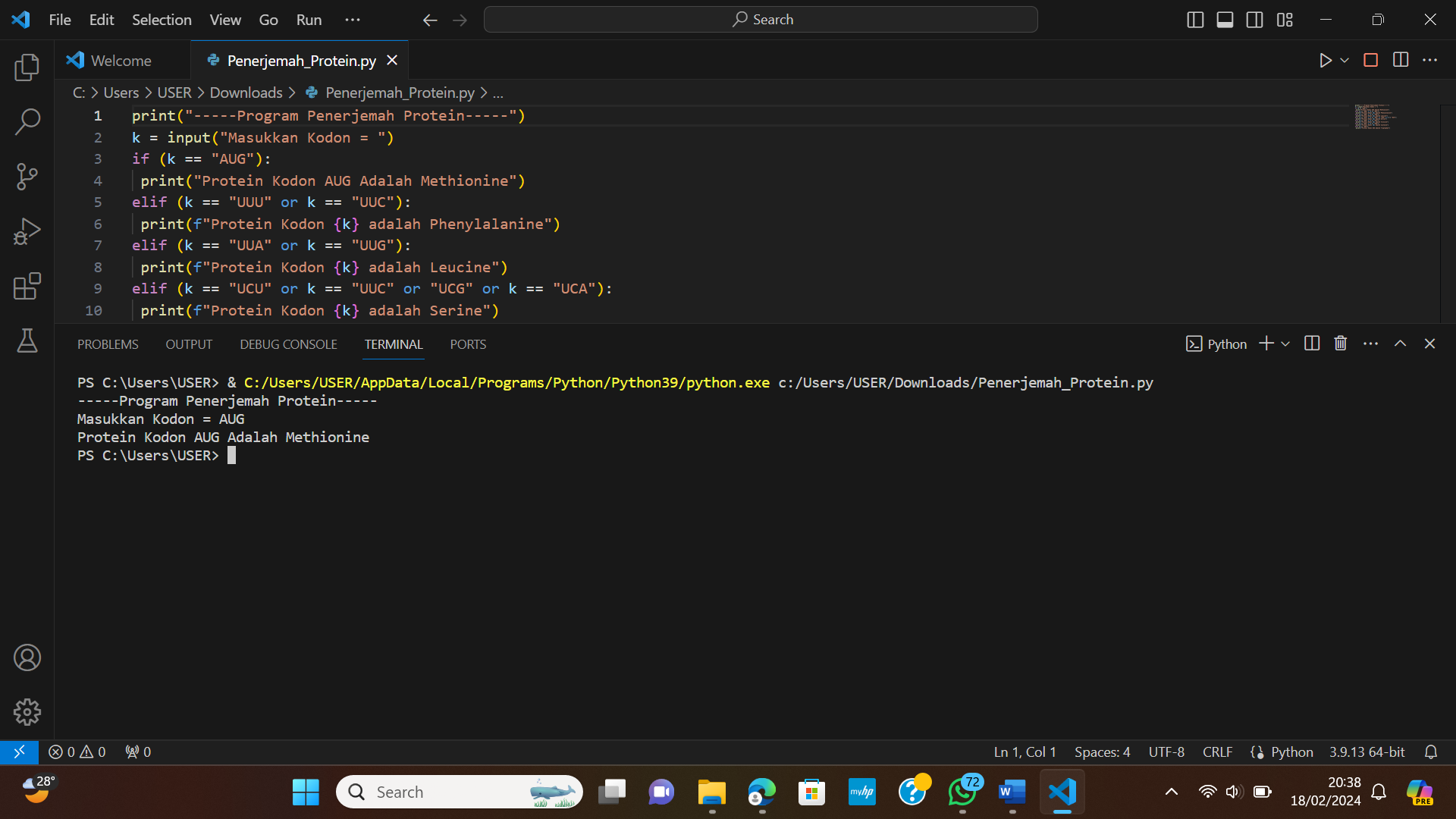


# *Nomor2: Penerjemah\_Protein.py*

Pada Awal Tampilan Output, pengguna diminta memasukkan nilai kodon.

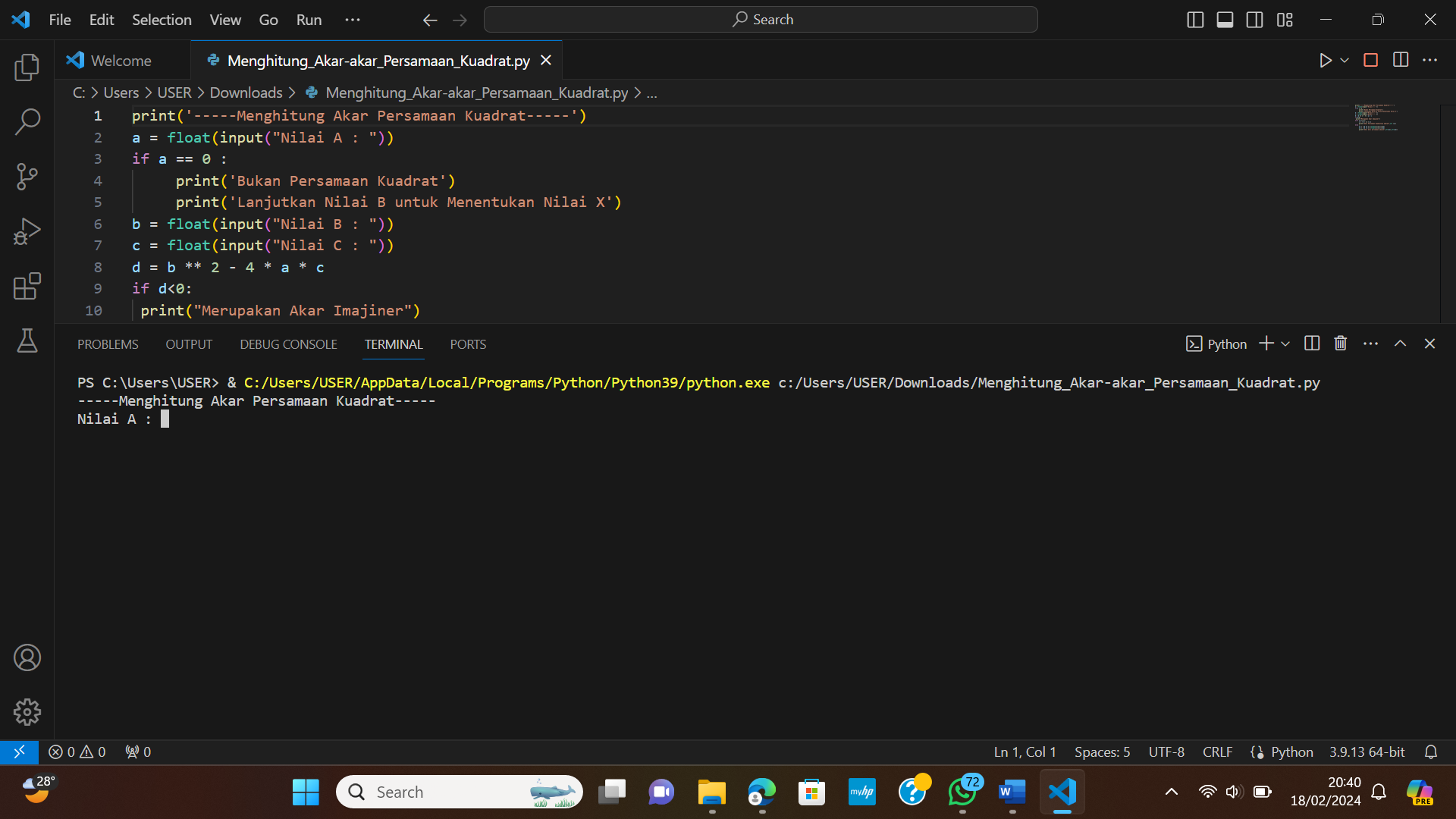


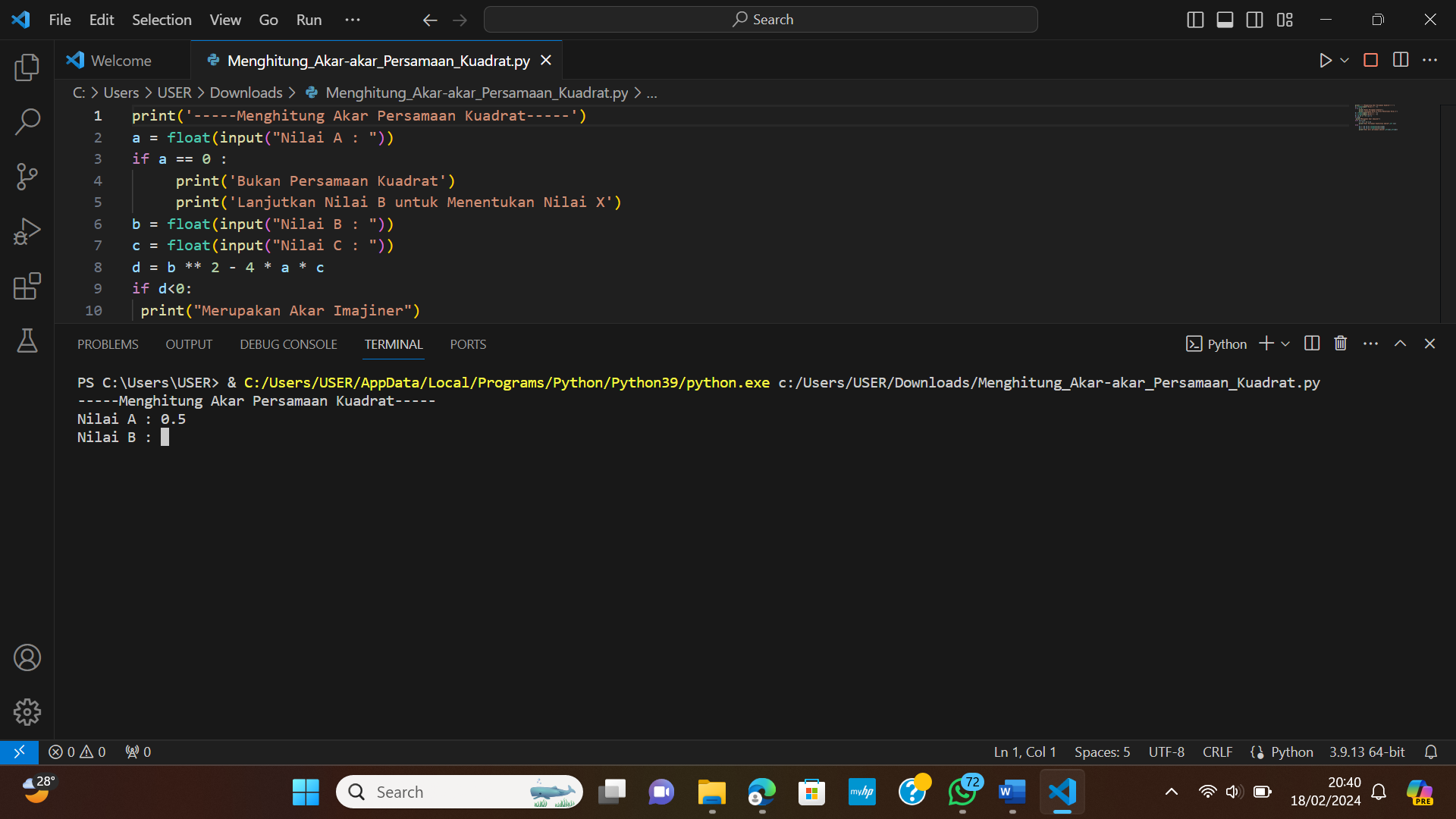
Setelah memasukkan nilai kodon, hasil program pengguna akan muncul di layar

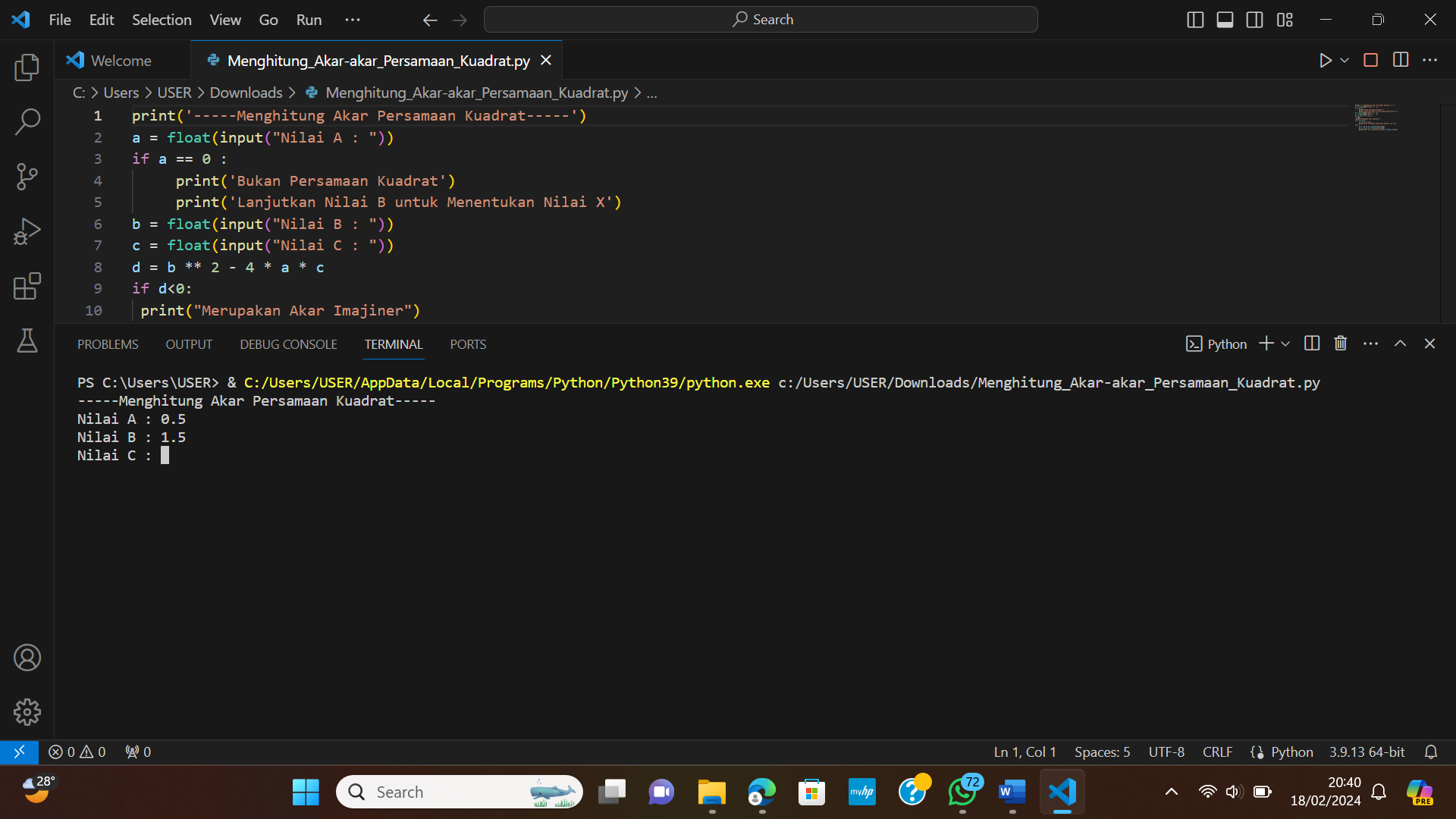


*Nomor3: Menghitung\_Akar\_Akar\_Persamaan\_Kuadrat.py*

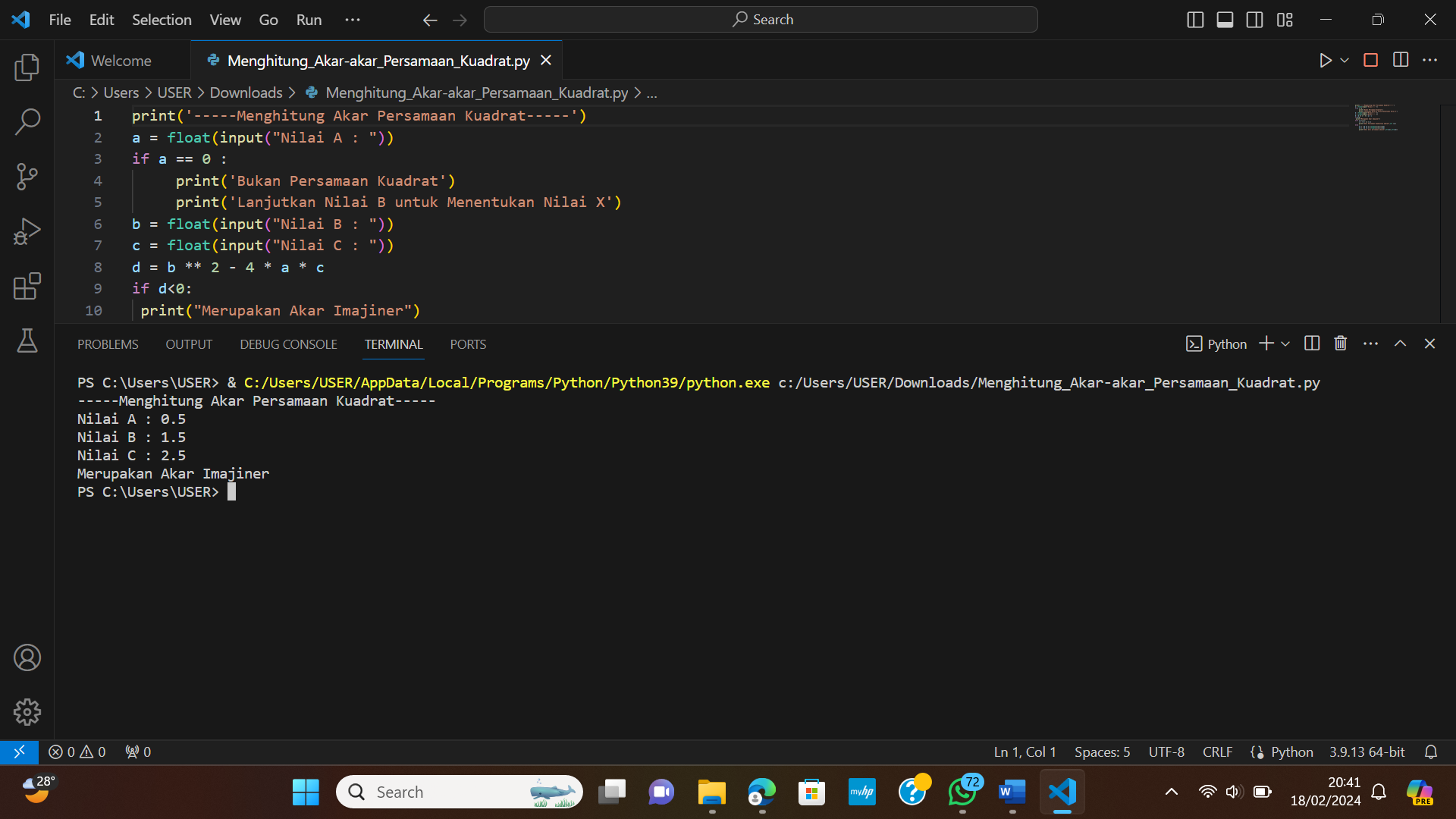
Pada Awal Tampilan Output, pengguna diminta memasukkan nilai A, B, dan C.







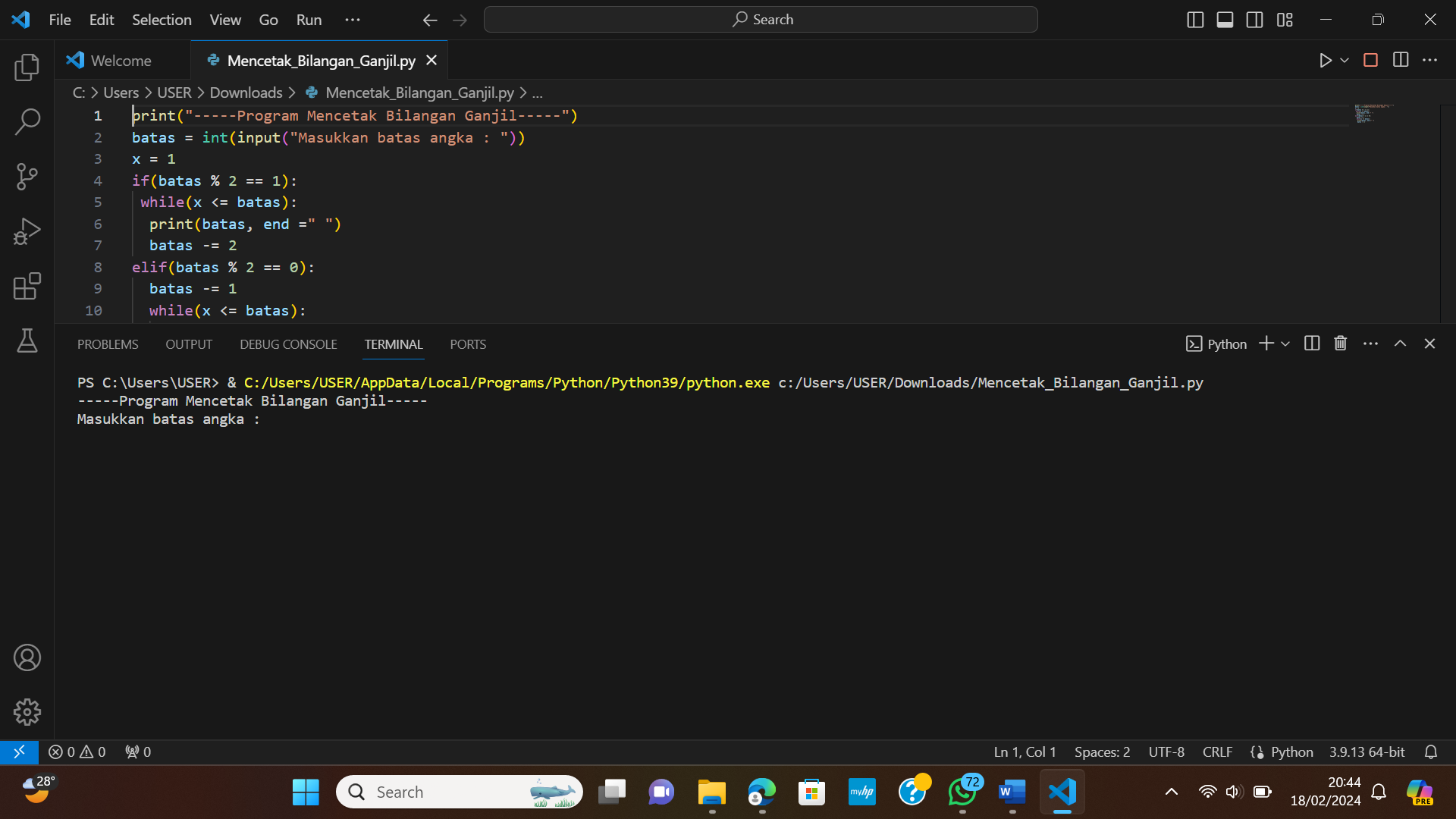
Setelah memasukkan nilai A,B,dan C, hasil program pengguna akan muncul di layar komputer



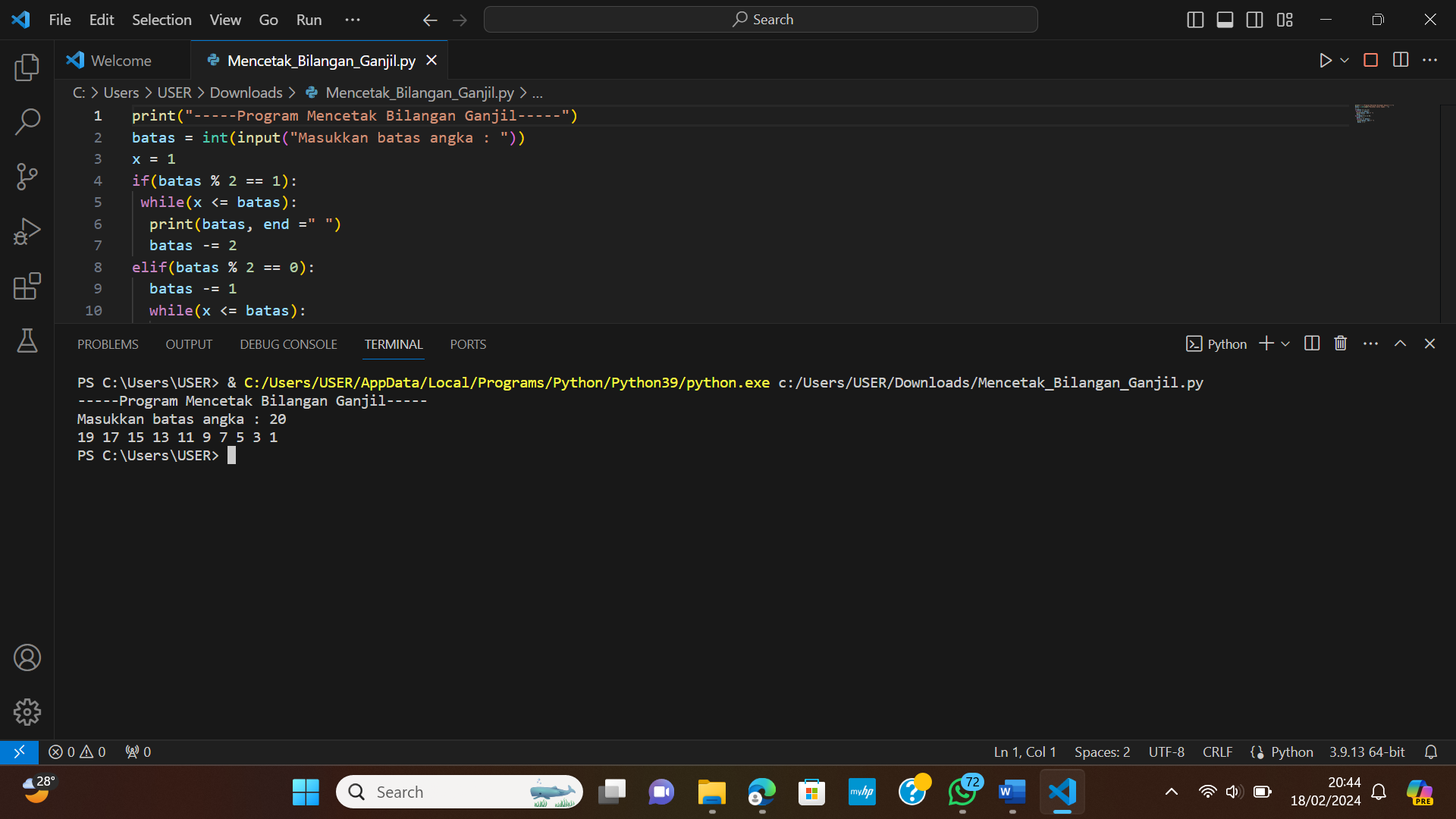
*Screenshoot modul ‘Perulangan dengan While’*

*Nomor1: Mencetak\_bilangan\_ganjil.py*

Pada Awal Tampilan Output, pengguna diminta untuk memasukkan batas angka.

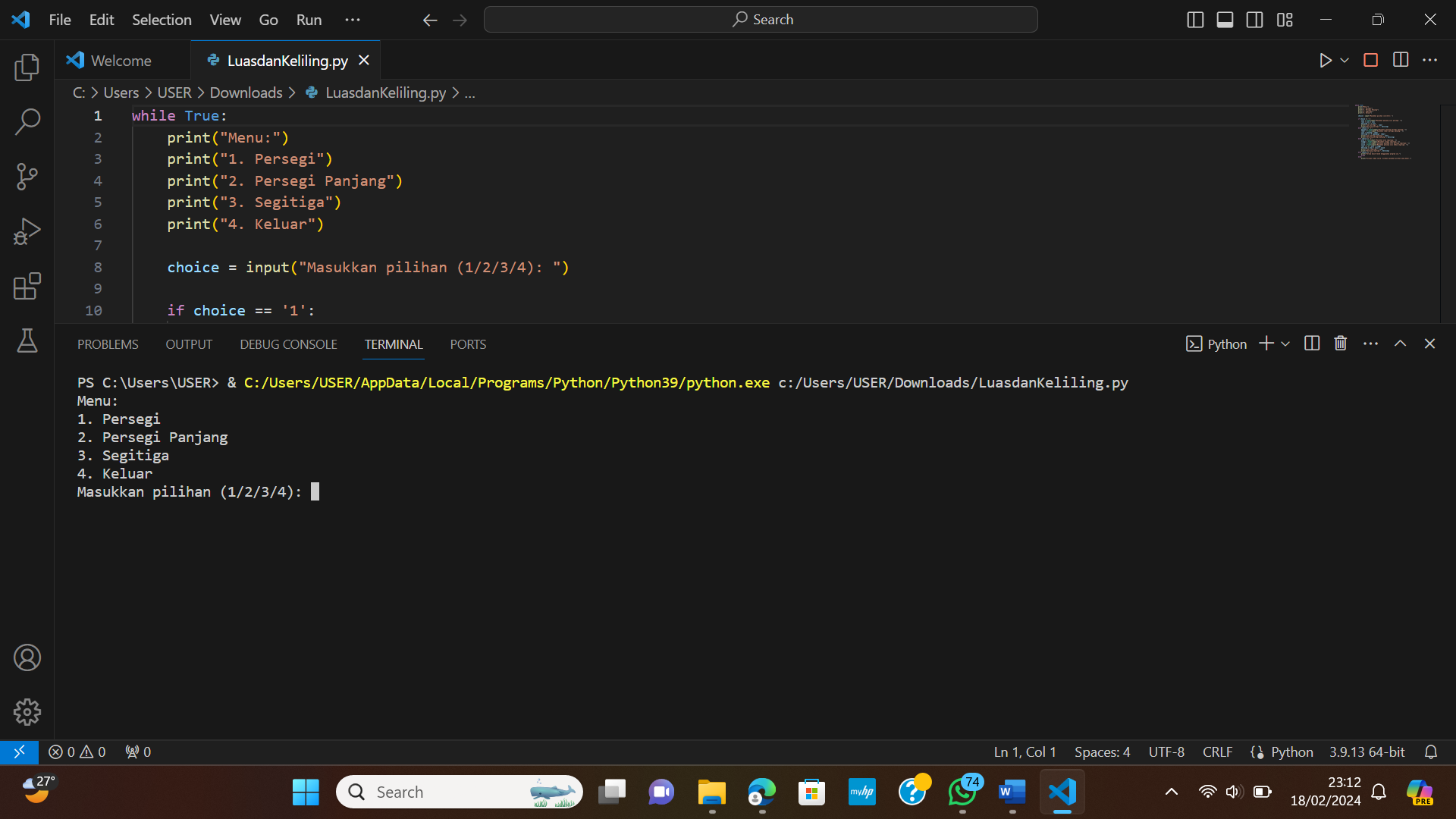


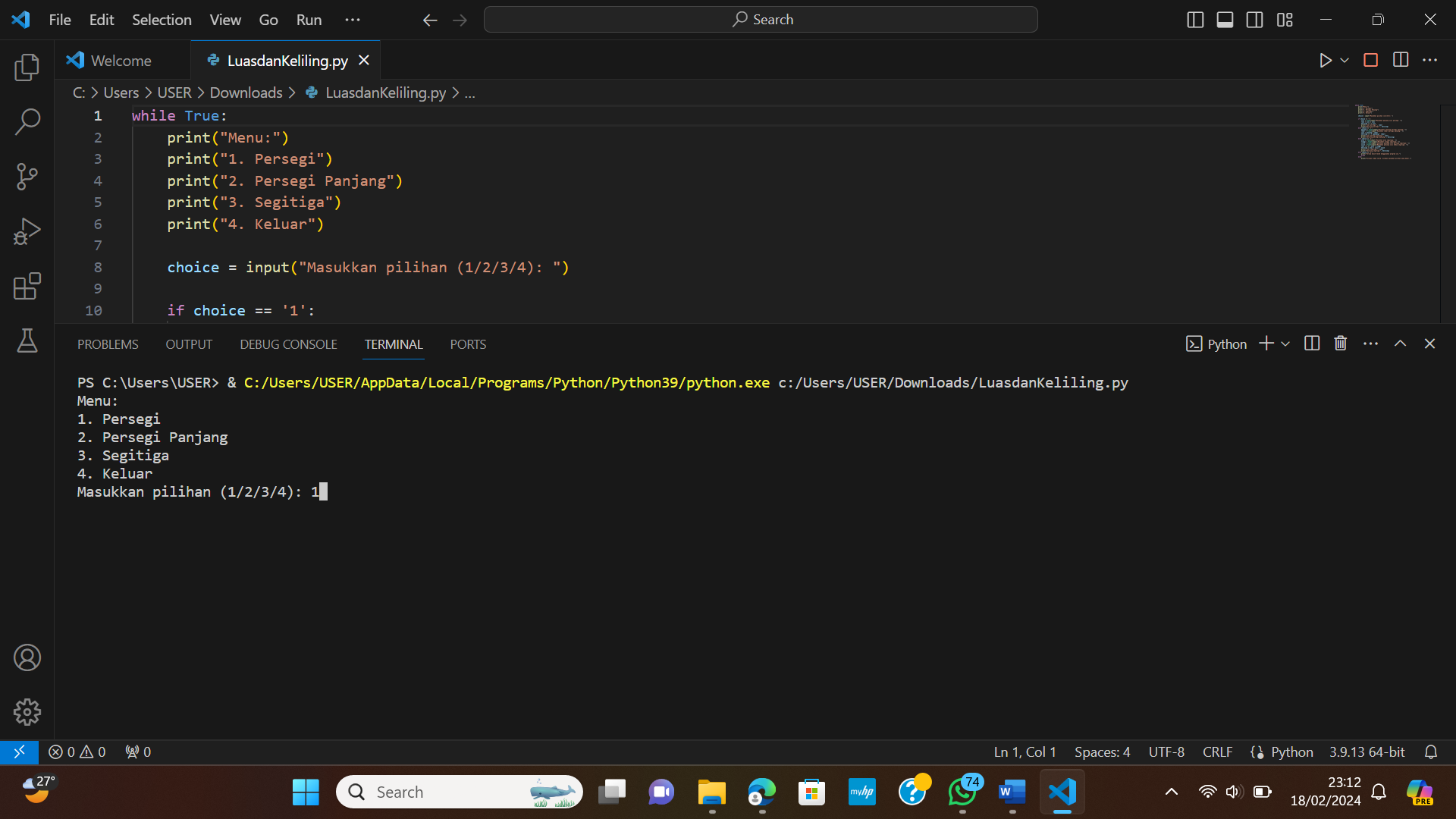
Setelah pengguna memasukkan batas angka, pengguna diminta memasukkan nilai A, B, dan C.



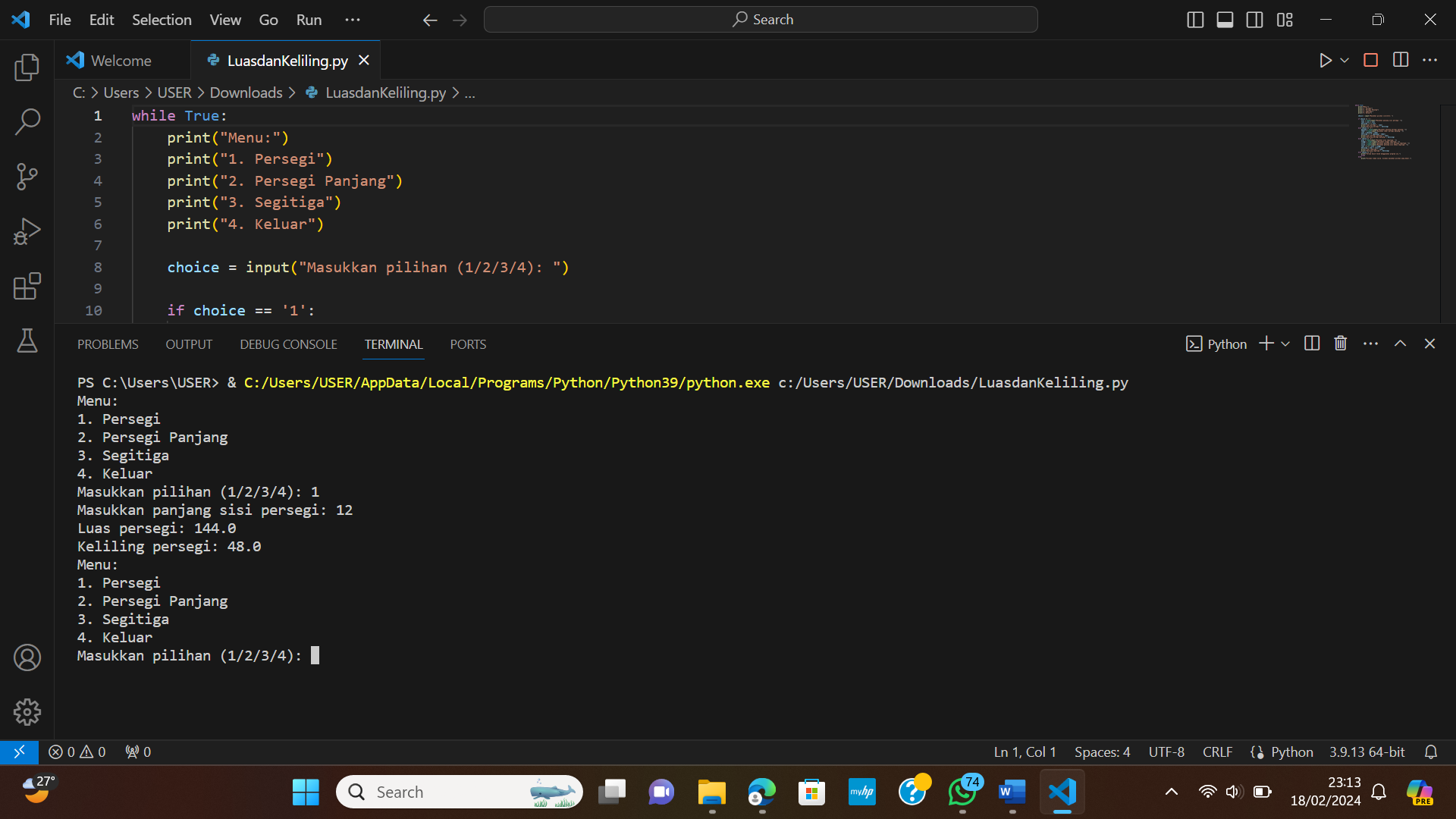
*Nomor2: Luas\_dan\_Keliling.py*

Pada Awal Tampilan Output, pengguna diminta untuk memilih menghitung luas dan keliling dari bangun datar, lalu pengguna memasukkan angka 1 untuk memilih menghitung luas dan keliling persegi.



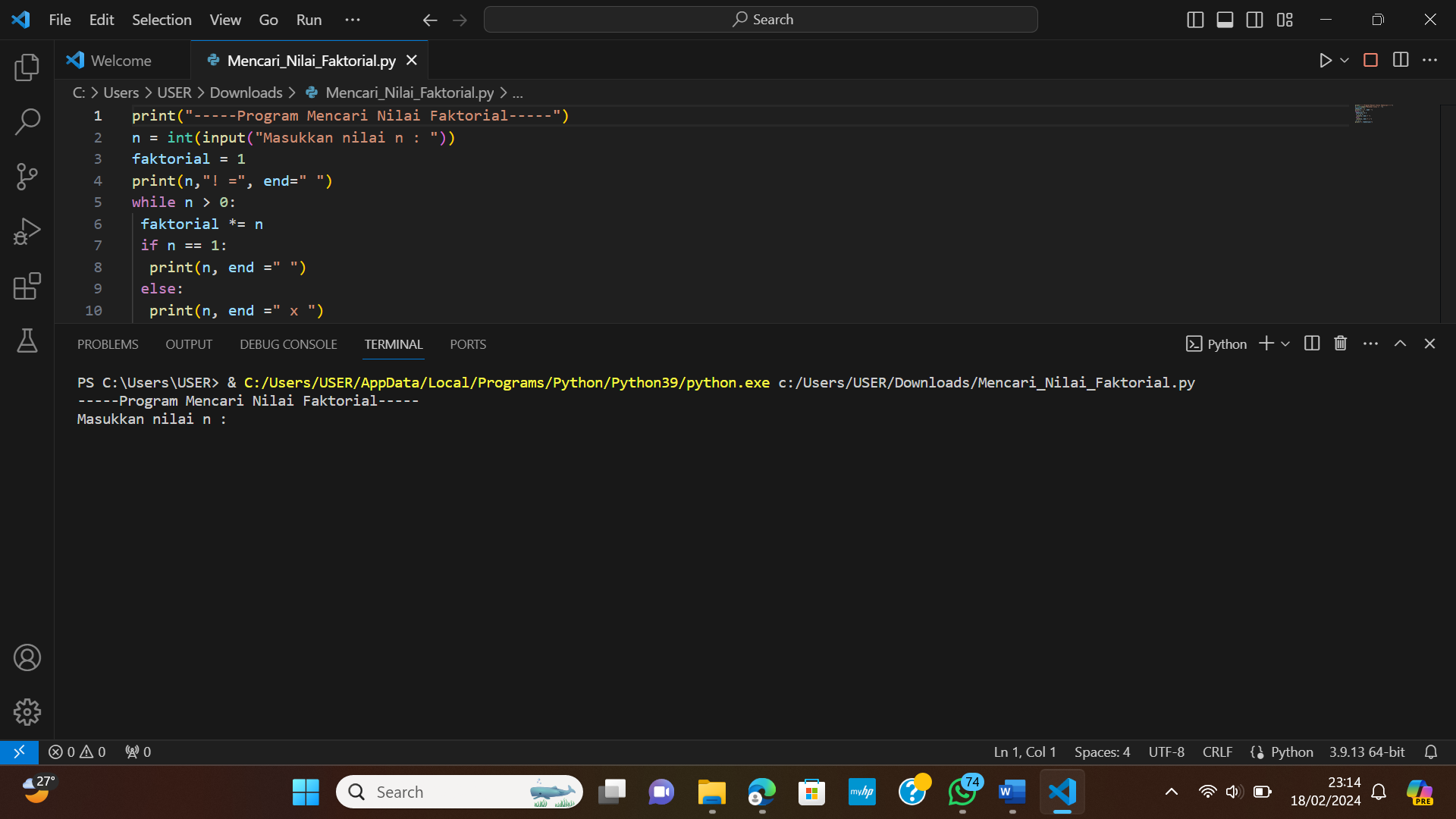


Setelah pengguna memasukkan nilai untuk panjang sisi, hasil program pengguna akan muncul di layar komputer.

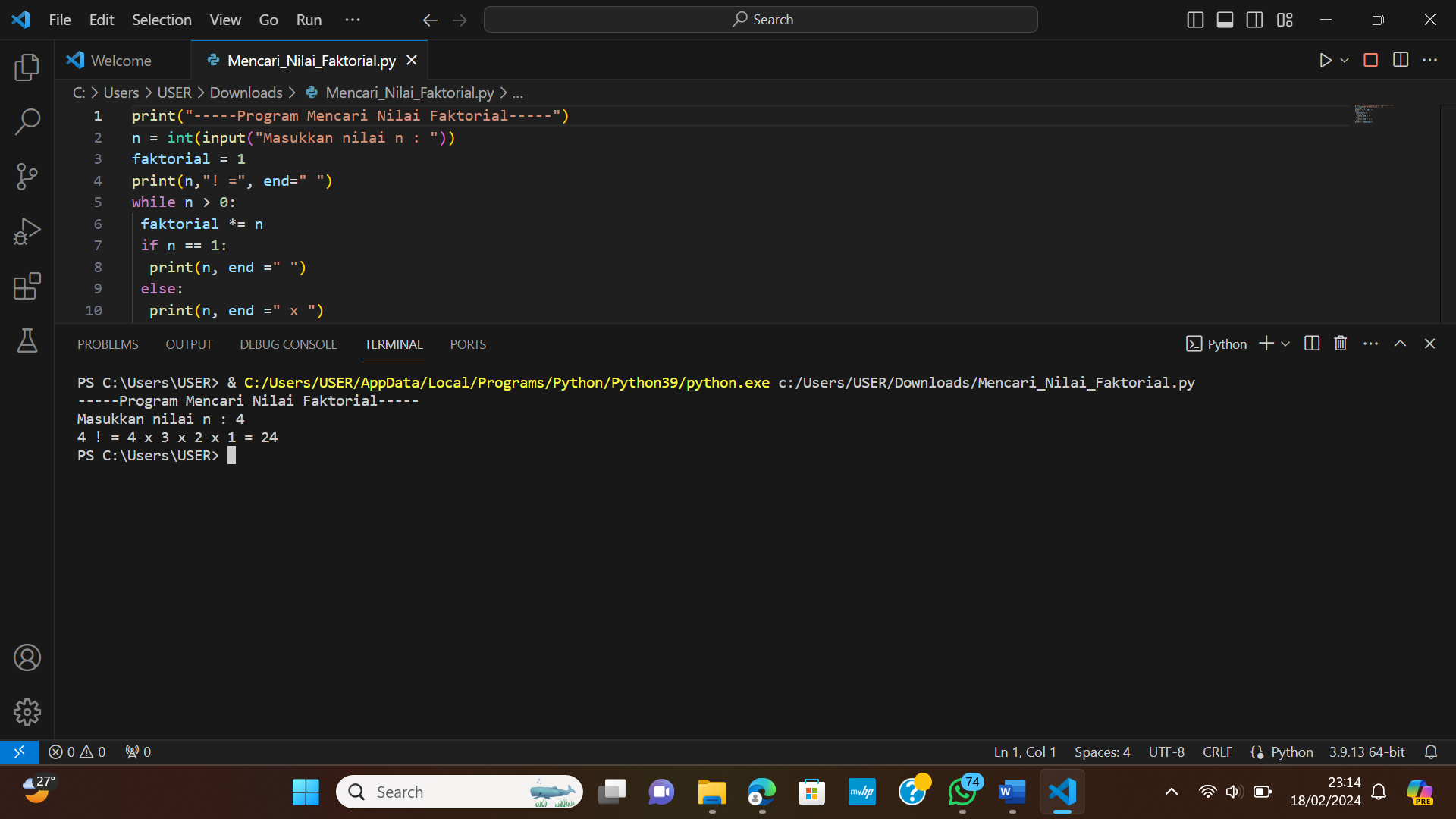


*Nomor3: Mencari\_Nilai\_Faktorial.py*

Pada Awal Tampilan Output, pengguna diminta memasukkan nilai n.



Setelah memasukkan nilai n, hasil program pengguna akan muncul di layar komputer.



# Pembahasan

## Pingsuit.py

| No. | Penjelasan |
| --- | --- |
| 1 | Program untuk memanggil modul random dan system |
| 3 | Print digunakan untuk mencetak nama program |
| 5–7 | Program untuk menjelaskan pilihan yang tersedia |
| 8 | Program untuk meng-input pilihan |
| 9 | Program if untuk pengkondisian pilihan dan print untuk meng-output |
| 10 | Programyangakanmenghentikanprogramapabilapilihanyangdi input pil < 1 or pil > 3 |
| 12 | Pengguna memasukkan pilihan 1-3 |
| 13-15 | Program if untuk kondisional pertama, pilihan dan print untuk meng-output |
| 16-17 | Program if untuk kondisional kedua, pilihan dan print untuk meng-output |
| 18-19 | Program if untuk kondisional ketiga, pilihan dan print untuk meng-output |
| 20 | Program elif untuk pengkondisian apabila (kom == 2) |
| 21-22 | Program if untuk kondisional keempat, pilihandanprint untuk meng-output |
| 23-24 | Program if untuk kondisional kelima, pilihan dan print untuk meng-output |
| 25-26 | Program if untuk kondisional keenam, pilihan dan print untuk meng-output |
| 27 | Program elif untuk pengkondisian apabila (kom == 3) |
| 28-29 | Program if untuk kondisional ketujuh, pilihan dan print untuk meng-output |
| 30-31 | Program if untuk kondisional kedelapan, pilihan dan print untuk meng-output |
| 32-33 | Program ifuntuk kondisional kesembilan,pilihan dan print untuk meng-output. |

## Nomor2.Penerjemah\_Protein.py

| No. | Penjelasan |
| --- | --- |
| 1 | Mencetak Judul Program |
| 2 | Input kodon berdasarkan jenisnya |
| 3-16 | Program if dan elif untuk jenis-jenis kodon, input dan print untuk meng-output |

## Nomor3.Menghitung\_Akar\_Akar\_Persamaan\_Kuadrat.py

| No. | Penjelasan |
| --- | --- |
| 1 | Mencetak Judul Program. |
| 2-3 | Input nilai A dan permisalan A = 0. |
| 4-7 | Input nilai B dan C sertamenentukan nilai X dan nilai bukan persamaan kuadrat . |
| 8-16 | Input rumus akar persamaan kuadrat. |
| 17 | Hasil dari akar persamaan kuadrat |

## 

## Nomor.1.Mencetak\_Bilangan\_Ganjil.py

| No. | Penjelasan |
| --- | --- |
| 1 | Mencetak Judul Program. |
| 3 | Variabel batas yang berisi masukan batas angka. |
| 4 | Variabel x yang berisi nilai 1. |
| 5 | if atau jika batas dibagi dengan 2 masih tersisa nilai 1. |
| 6 | While untuk x kurang dari sama dengan batas yang dimasukkan. |
| 7 | Cetak batas dengan end=” ” agar output sejajar atau tidak menurun. |
| 8 | Batas dikurangi dengan 2 |
| 9 | elif jika batas habis dibagi dengan 2 |
| 10 | Batas dikurangi dengan 1 |
| 11 | While atau ketika x kurang dari sama dengan nilai batas. |
| 12 | Cetak batas dengan end=” ” agar output sejajar. |
| 13 | Batas dikurangi dengan 2 |

## Nomor2.Luas\_dan\_Keliling.py

| No. | Penjelasan |
| --- | --- |
| 1 | Penjelasan baris 1 |
| 2 | Penjelasan baris 2 |
| 4 | Penjelasan baris 4 |
| 6 | Penjelasan baris 6 |
| 7-10 | Penjelasan baris 7-10 |

## Nomor3.Mencari\_Nilai\_Faktorial.py

| No. | Penjelasan |
| --- | --- |
| 1 | Mencetak Judul Program |
| 3 | Membuat variabel n yang berisi inputan nilai yang akan difaktorkan |
| 4 | Variabel faktorial dengan nilai 1 |
| 5 | Cetak nilai n! dengan end =” ” agar sejajar. |
| 6 | While atau ketika n lebih besar dari 0. |
| 7 | Variabel hasil dikali dengan nilai n |
| 8 | If atau jika n sama dengan 1 |
| 9 | Cetak n dengan end=” ” |
| 10 | Else. |
| 11 | Cetak n dengan end =” x ”. |
| 12 | Variabel n dikurangi 1. |
| 13 | Cetak hasil |

# Kesimpulan

1. Kesimpulan yang dapat ditarik dari praktikum ini adalah bahwa penggunaan Eksekusi Kondisional dalam pemrograman dapat memfasilitasi komputer untuk mengambil tindakan berdasarkan kondisi tertentu. Dalam eksekusi kondisional, terdapat tiga jenis sintaks yang digunakan, yaitu if, elif, dan else. Sintaks if adalah yang pertama kali mengevaluasi kondisi, dan jika hasilnya False, maka selanjutnya sintaks elif akan dievaluasi. Jika tidak ada satu pun dari kondisi if dan elif yang terpenuhi, maka baris kode pada sintaks else akan dieksekusi

2. Perulangan While adalah cara untuk mengulang eksekusi suatu blok kode program selama kondisi tertentu tetap terpenuhi. Perulangan While bersifat tidak pasti atau indefinite, yang berarti perulangan akan berlanjut tanpa batas waktu tertentu, dan hanya akan berhenti ketika kondisi tertentu tidak lagi terpenuhi.

3. Dalam Perulangan While, terdapat penggunaan fungsi end=" " yang berperan untuk menampilkan hasil dari perulangan secara horizontal. Selain itu, berbagai operasi juga bisa digunakan, seperti <=, >=, ==, dan sejenisnya.

# Saran

Kepada kakak asisten, pendekatan dan penjelasan yang diberikan dalam pengajaran sudah sangat baik. Saya ingin menyarankan untuk tetap mempertahankan gaya pengajaran seperti yang telah diterapkan sebelumnya. Terima kasih.

# Daftar Pustaka

*Belajar Pemrograman Python: Memahami Percabangan untuk Membuat Logika Program*. (2016, November 9). Petani Kode. Retrieved February 19, 2024, from https://www.petanikode.com/python-percabangan/

*Belajarpython – Situs Open Source Tutorial Pemrograman Python Bahasa Indonesia.* (n.d.). Belajarpython. Retrieved February 19, 2024, from https://belajarpython.com/tutorial/kondisi-if-else-python

*5 Aturan Penulisan Sintaks Python yang Harus dipatuhi*. (2017, July 31). Petani Kode. Retrieved February 19, 2024, from https://www.petanikode.com/python-sintaks/