**LAPORAN PRAKTIKUM**

**PEMROGRAMAN I**

**MODUL 6**

****

**Array**

**Oleh:**

**Achmad Reihan Alfaiz NIM. 2410817210019**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS LAMBUNG MANGKURAT**

**DESEMBER 2024**

# LEMBAR PENGESAHAN

**LAPORAN PRAKTIKUM PEMROGRAMAN I**

**MODUL 6**

Laporan Praktikum Pemrograman I Modul 6 : Array ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman I. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Achmad Reihan Alfaiz

NIM : 2410817210019

|  |  |
| --- | --- |
| Menyetujui,  Asisten Praktikum  Muhammad Ryan Rizky Rahmadi  NIM. 2210817310001 | Mengetahui,  Dosen Penanggung Jawab Praktikum  Helda Yunita, S.Kom., M.Kom.  NIP. 199106192024062001 |

# DAFTAR ISI

[LEMBAR PENGESAHAN 2](#_Toc185086908)

[DAFTAR ISI 3](#_Toc185086909)

[DAFTAR GAMBAR 5](#_Toc185086910)

[DAFTAR TABEL 6](#_Toc185086911)

[SOAL 1 7](#_Toc185086912)

[A. Source Code 8](#_Toc185086913)

[**1.** **Source Code C** 8](#_Toc185086914)

[**2.** **Source Code Python** 8](#_Toc185086915)

[B. Output Program 9](#_Toc185086916)

[**1.** **Output Program C** 9](#_Toc185086917)

[**2.** **Output Program Python** 9](#_Toc185086918)

[C. Pembahasan 9](#_Toc185086919)

[**1.** **Pembahasan Code/Program C** 9](#_Toc185086920)

[**2.** **Pembahasan Code/Program Python** 10](#_Toc185086921)

[SOAL 2 12](#_Toc185086922)

[A. Source Code 12](#_Toc185086923)

[**1.** **Source Code C** 12](#_Toc185086924)

[**2.** **Source Code Python** 13](#_Toc185086925)

[B. Output Program 13](#_Toc185086926)

[**1.** **Output Program C** 13](#_Toc185086927)

[**2.** **Output Program Python** 13](#_Toc185086928)

[C. Pembahasan 13](#_Toc185086929)

[**1.** **Pembahasan Code/Program C** 13](#_Toc185086930)

[**2.** **Pembahasan Code/Program Python** 14](#_Toc185086931)

[SOAL 3 16](#_Toc185086932)

[A. Source Code 16](#_Toc185086933)

[**1.** **Source Code C** 16](#_Toc185086934)

[**2.** **Source Code Python** 17](#_Toc185086935)

[B. Output Program 17](#_Toc185086936)

[**1.** **Output Program C** 17](#_Toc185086937)

[**2.** **Output Program Python** 18](#_Toc185086938)

[C. Pembahasan 18](#_Toc185086939)

[**1.** **Pembahasan Code/Program C** 18](#_Toc185086940)

[**2.** **Pembahasan Code/Program Python** 19](#_Toc185086941)

[SOAL 4 21](#_Toc185086942)

[A. Source Code 22](#_Toc185086943)

[**1.** **Source Code C** 22](#_Toc185086944)

[**2.** **Source Code Python** 23](#_Toc185086945)

[B. Output Program 23](#_Toc185086946)

[**1.** **Output Program C** 23](#_Toc185086947)

[**2.** **Output Program Python** 24](#_Toc185086948)

[C. Pembahasan 24](#_Toc185086949)

[**1.** **Pembahasan Code/Program C** 24](#_Toc185086950)

[**2.** **Pembahasan Code/Program Python** 25](#_Toc185086951)

[SOAL 5 27](#_Toc185086952)

[A. Source Code 27](#_Toc185086953)

[**1.** **Source Code C** 27](#_Toc185086954)

[**2.** **Source Code Python** 29](#_Toc185086955)

[B. Output Program 29](#_Toc185086956)

[**1.** **Output Program C** 29](#_Toc185086957)

[**2.** **Output Program Python** 30](#_Toc185086958)

[C. Pembahasan 30](#_Toc185086959)

[**1.** **Pembahasan Code/Program C** 30](#_Toc185086960)

[**2.** **Pembahasan Code/Program Python** 31](#_Toc185086961)

# DAFTAR GAMBAR

[Gambar 1. Screenshot Hasil Jawaban Soal 1 pada Bahasa Pemrograman C 9](#_Toc185086898)

[Gambar 2. Screenshot Hasil Jawaban Soal 1 pada Bahasa Pemrograman Python 9](#_Toc185086899)

[Gambar 3. Screenshot Hasil Jawaban Soal 2 pada Bahasa Pemrograman C 13](#_Toc185086900)

[Gambar 4. Screenshot Hasil Jawaban Soal 2 pada Bahasa Pemrograman Python 13](#_Toc185086901)

[Gambar 5. Screenshot Hasil Jawaban Soal 3 pada Bahasa Pemrograman C 17](#_Toc185086902)

[Gambar 6. Screenshot Hasil Jawaban Soal 3 pada Bahasa Pemrograman Python 18](#_Toc185086903)

[Gambar 7. Screenshot Hasil Jawaban Soal 4 pada Bahasa Pemrograman C 23](#_Toc185086904)

[Gambar 8. Screenshot Hasil Jawaban Soal 4 pada Bahasa Pemrograman Python 24](#_Toc185086905)

[Gambar 9. Screenshot Hasil Jawaban Soal 5 pada Bahasa Pemrograman C 29](#_Toc185086906)

[Gambar 10. Screenshot Hasil Jawaban Soal 5 pada Bahasa Pemrograman Python 30](#_Toc185086907)

# DAFTAR TABEL

[Tabel 1. Soal Nomor 1 7](#_Toc185086883)

[Tabel 2. Source Code Soal 1 pada Bahasa Pemrograman C 8](#_Toc185086884)

[Tabel 3. Source Code Soal 1 pada Bahasa Pemrograman Python 8](#_Toc185086885)

[Tabel 4. Soal Nomor 2 12](#_Toc185086886)

[Tabel 5. Source Code Soal 2 pada Bahasa Pemrograman C 13](#_Toc185086887)

[Tabel 6. Source Code Soal 2 pada Bahasa Pemrograman Python 13](#_Toc185086888)

[Tabel 7. Soal Nomor 3 16](#_Toc185086889)

[Tabel 8. Source Code Soal 3 pada Bahasa Pemrograman C 17](#_Toc185086890)

[Tabel 9. Source Code Soal 3 pada Bahasa Pemrograman Python 17](#_Toc185086891)

[Tabel 10. Soal Nomor 4 21](#_Toc185086892)

[Tabel 11. Source Code Soal 4 pada Bahasa Pemrograman C 22](#_Toc185086893)

[Tabel 12. Source Code Soal 4 pada Bahasa Pemrograman Python 23](#_Toc185086894)

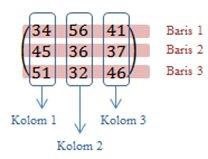
[Tabel 13. Soal Nomor 5 27](#_Toc185086895)

[Tabel 14. Source Code Soal 5 pada Bahasa Pemrograman C 28](#_Toc185086896)

[Tabel 15. Source Code Soal 5 pada Bahasa Pemrograman Python 29](#_Toc185086897)

# SOAL 1

Matriks adalah kumpulan bilangan yang disusun secara baris dan kolom yang kemudian diisi dengan angka-angka pada matriks tersebut. Misalnya sebuah matriks memiliki 3 baris dan 3 kolom dengan isi yang ada di dalam matriks tersebut adalah {(34), (56), (41), (45), (36), (37), (51), (32), (46)} maka akan terbentuk matriks sebagai berikut:



Untuk mendapatkan kekuatan *One For All* Midoriya Izuku harus membuat sebuah matriks sesuai dengan baris dan kolom yang ditetapkan beserta isi yang ada didalamnya. Buatlah sebuah program untuk membantu Midoriya Izuku membuat matriks.

Format input:

Input pada baris pertama berupa jumlah baris dan kolom.

Input pada baris kedua berupa angka yang ada di dalam matriks tersebut. Jumlah input pada baris kedua = (baris x kolom) yang dipisahkan dengan spasi. Format output: Output merupakan sebuah matriks .

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 3  1 2 3 4 5 6 | 1 2 3  4 5 6 |
| 3 3  34 56 41 45 36 37 51 32 46 | 34 56 41  45 36 37  51 32 46 |
| 4 5  1 1 1 1 2 3 5 6 4 5 8 7 9 6 5 4 1 2 5 6 | 1 1 1 1 2  3 5 6 4 5  8 7 9 6 5  4 1 2 5 6 |

Tabel 1. Soal Nomor 1

## Source Code

### **Source Code C**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | #include <stdio.h>  int main()  {       int row, column;       scanf("%d %d", &row, &column);       int matrix[row][column];       for (int i = 0; i < row; i++)       {            for (int j = 0; j < column; j++)            {                 scanf("%d", &matrix[i][j]);                 printf("%d ", matrix[i][j]);            }            printf("\n");       }       return 0;  } |

Tabel 2. Source Code Soal 1 pada Bahasa Pemrograman C

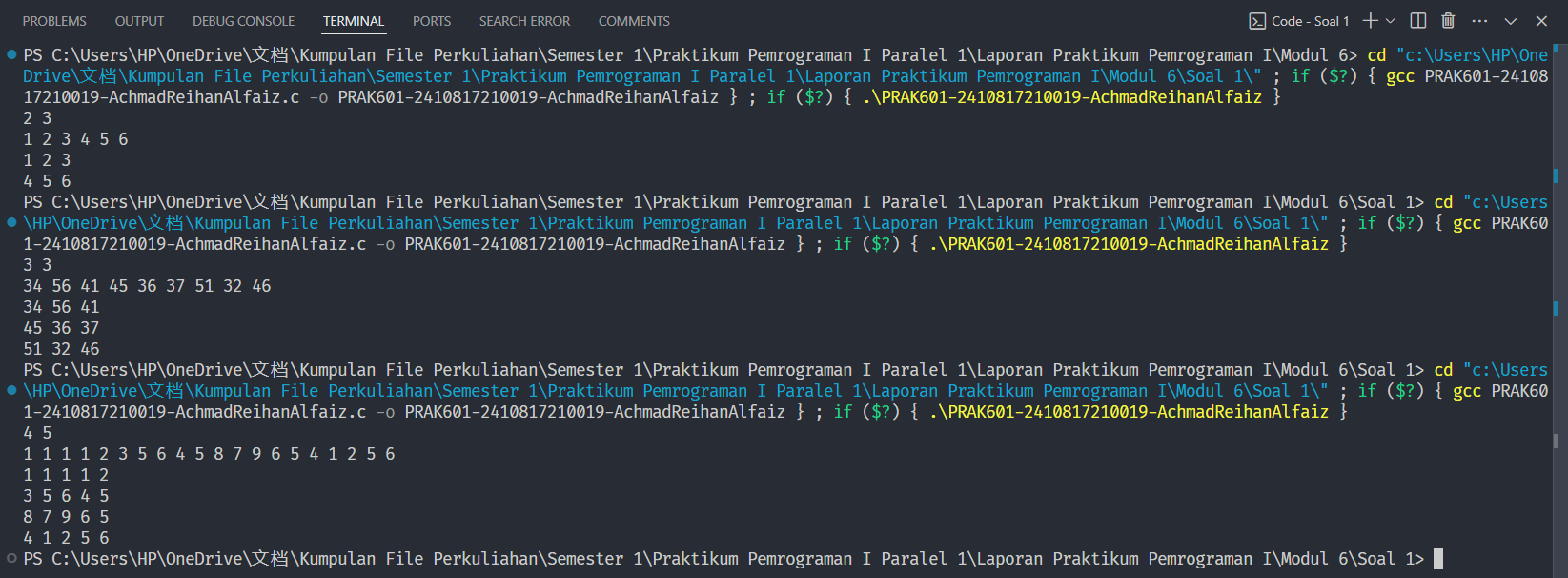
### **Source Code Python**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | baris, kolom = map(int, input().split())  matrix = []  values = list(map(int, input().split()))  for i in range(baris):      row = values[i \* kolom : (i + 1) \* kolom]      matrix.append(row)  for row in matrix:      print(" ".join(map(str, row))) |

Tabel 3. Source Code Soal 1 pada Bahasa Pemrograman Python

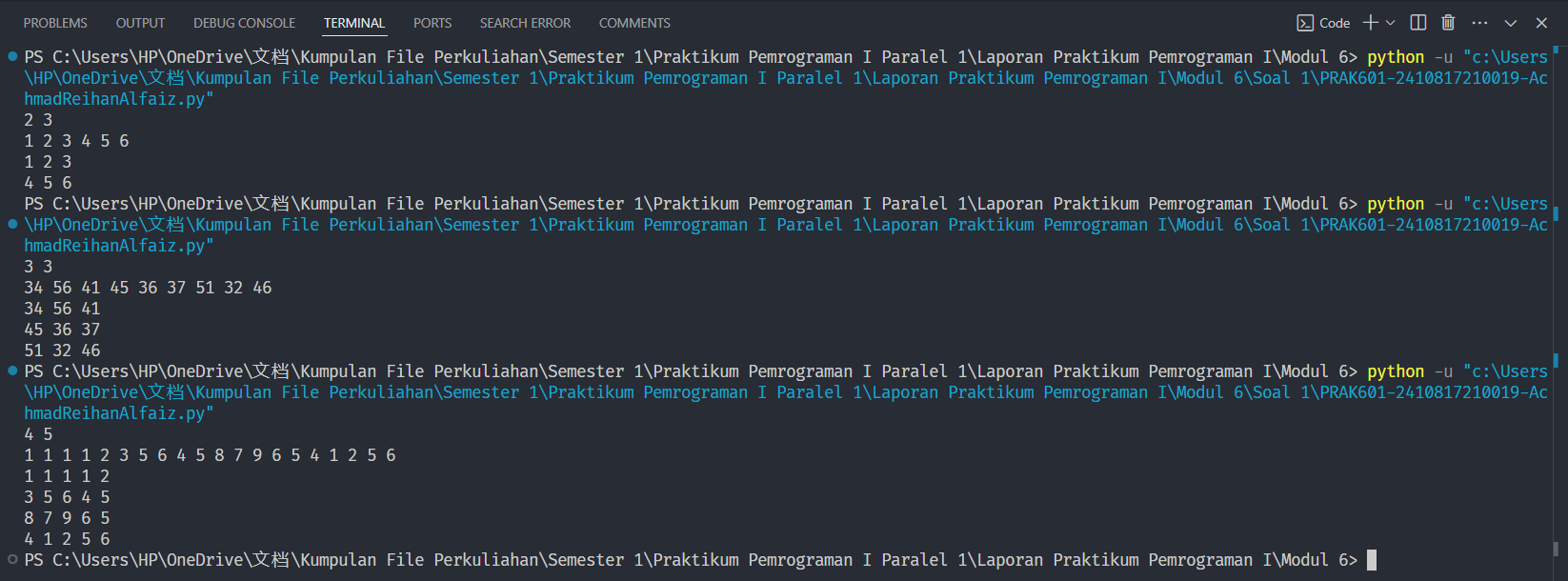
## Output Program

### **Output Program C**



Gambar 1. Screenshot Hasil Jawaban Soal 1 pada Bahasa Pemrograman C

### **Output Program Python**



Gambar 2. Screenshot Hasil Jawaban Soal 1 pada Bahasa Pemrograman Python

## Pembahasan

### **Pembahasan Code/Program C**

**Line 1:** Baris pertama ini menyertakan library standar input-output di C (<stdio.h>) yang menyediakan fungsi untuk melakukan operasi input dan output, seperti printf untuk mencetak teks ke layar dan fgets untuk membaca input dari pengguna.

**Line 3:** Fungsi main() adalah titik masuk utama dari program. Fungsi main() merupakan standar dalam bahasa C yang menandakan bahwa program akan dimulai dari fungsi ini saat dieksekusi.

**Line 5–6:** row: Menyimpan jumlah baris pada matriks. column: Menyimpan jumlah kolom pada matriks. scanf("%d %d", &row, &column); Program membaca dua bilangan bulat dari pengguna: Jumlah baris (row). Jumlah kolom (column).

**Line 8:** Mendeklarasikan sebuah matriks 2D berukuran row × column. Matriks ini dinamis karena ukurannya ditentukan oleh input pengguna. Contoh: Jika row = 3 dan column = 2, matriks akan memiliki 3 baris dan 2 kolom.

**Line 10–18:** Perulangan Baris (for i): Perulangan luar berjalan dari i = 0 hingga i < row (mengakses setiap baris). Perulangan Kolom (for j): Perulangan dalam berjalan dari j = 0 hingga j < column (mengakses setiap elemen dalam baris). Membaca Elemen Matriks: scanf("%d", &matrix[i][j]); Program membaca elemen matriks dari input pengguna untuk posisi [i][j]. Menampilkan Elemen Matriks: printf("%d ", matrix[i][j]); Elemen matriks pada posisi [i][j] langsung dicetak ke layar, dipisahkan oleh spasi. Pindah Baris: printf("\n"); Setelah semua elemen dalam satu baris dicetak, program berpindah ke baris berikutnya.

**Line 20:** Baris terakhir return 0; digunakan untuk menunjukkan bahwa program telah selesai dijalankan tanpa kesalahan. Di C, nilai 0 biasanya menandakan eksekusi sukses, sementara nilai non-zero dapat menunjukkan bahwa ada masalah yang terjadi.

### **Pembahasan Code/Program Python**

**Line 1–2:** input(): Membaca satu baris input dari pengguna, yang berisi jumlah baris dan kolom matriks, dipisahkan oleh spasi. map(int, ...): Mengonversi setiap elemen dari input string menjadi bilangan bulat. Variabel baris dan kolom: baris: Menyimpan jumlah baris pada matriks. kolom: Menyimpan jumlah kolom pada matriks. Matriks diinisialisasi sebagai daftar kosong (list), yang akan diisi pada langkah berikutnya.

**Line 4:** input(): Membaca satu baris input dari pengguna, yang berisi elemen-elemen matriks dipisahkan oleh spasi. map(int, ...): Mengonversi setiap elemen string menjadi bilangan bulat. list(...): Mengubah hasil map menjadi daftar Python. Variabel values: Menyimpan semua elemen matriks sebagai daftar 1D.

**Line 6–8:** Perulangan for i: Iterasi dari i = 0 hingga i = baris - 1, untuk membentuk setiap baris matriks. Mengambil Elemen untuk Baris ke-i: row = values[i \* kolom : (i + 1) \* kolom] Menggunakan slicing pada daftar values untuk mengambil elemen-elemen dari kolom ke-i \* kolom hingga kolom ke-(i + 1) \* kolom - 1. Elemen-elemen ini membentuk satu baris matriks. Menambahkan Baris ke Matriks: matrix.append(row) Menambahkan baris yang baru saja dibentuk ke matriks. Contoh: Jika values = [1, 2, 3, 4, 5, 6], baris = 3, kolom = 2: Iterasi 1 (i = 0): row = values[0:2] = [1, 2] matrix = [[1, 2]] Iterasi 2 (i = 1): row = values[2:4] = [3, 4] matrix = [[1, 2], [3, 4]] Iterasi 3 (i = 2): row = values[4:6] = [5, 6] matrix = [[1, 2], [3, 4], [5, 6]]

**Line 10–11:** Perulangan for row: Iterasi melalui setiap baris dalam matriks. Mengubah Elemen ke String: map(str, row) Mengonversi setiap elemen dalam baris menjadi string, karena fungsi join membutuhkan input berupa string. Menggabungkan Elemen dengan Spasi: " ".join(map(str, row)) Menggabungkan elemen-elemen dalam baris menjadi satu string, dipisahkan oleh spasi. Mencetak Baris: Setiap baris matriks dicetak dalam format 2D.

# SOAL 2

Zetsu putih merupakan karakter yang mampu membelah diri dengan kelipatan sesuai dengan label ruangan yang ada.

Misalnya terdapat 3 ruangan dengan rincian sebagai berikut:

Pada ruangan berlabel 1 dimasukkan 4 zetsu putih, pada ruangan berlabel 2 dimasukkan 7 zetsu putih, pada ruangan berlabel 3 dimasukkan 9 zetsu putih. Setelah beberapa saat, zetsu putih yang ada pada ruangan berlabel 1 menjadi 4×1 zetsu putih, zetsu putih yang ada pada ruangan berlabel 2 menjadi 7×2 zetsu putih, zetsu putih yang ada pada ruangan berlabel 3 menjadi 9×3 zetsu putih. Buatlah sebuah program untuk menghitung jumlah zetsu putih pada setiap ruangan.

Format input:

Baris pertama merupakan jumlah ruangan.

Baris kedua merupakan banyaknya zetsu putih yang dimasukkan pada setiap ruangan. Format output:

Output merupakan jumlah zetsu putih setelah membelah diri pada setiap ruangan.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  4 7 9 | 4 14 27 |
| 5  1 2 3 4 5 | 1 4 9 16 25 |
| 10  5 6 45 78 21 3 6 8 45 1 | 5 12 135 312 105 18 42 64 405 10 |

Tabel 4. Soal Nomor 2

## Source Code

### **Source Code C**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | #include <stdio.h>  int main()  {       int valueCount;       scanf("%d", &valueCount);       int result[valueCount];       for (int i = 0; i < valueCount; i++)            scanf("%d", &result[i]), result[i] \*= (i + 1);       for (int i = 0; i < valueCount; i++)            printf("%d ", result[i]);       return 0;  } |

Tabel 5. Source Code Soal 2 pada Bahasa Pemrograman C

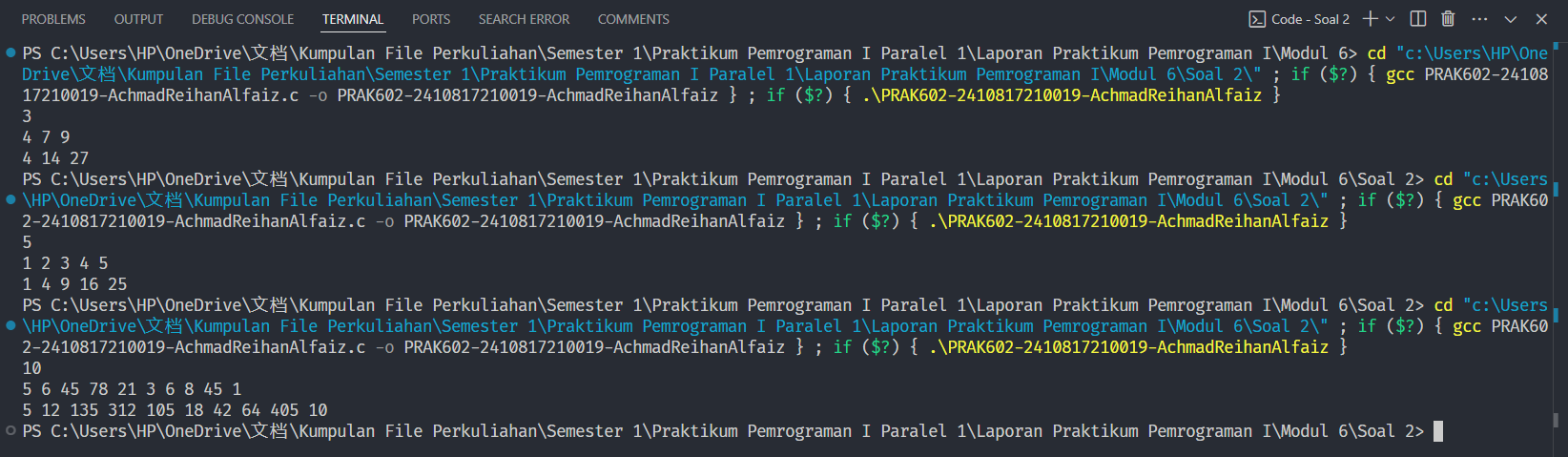
### **Source Code Python**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | value\_count = int(input())  values = list(map(int, input().split()))  result = []  for i in range(value\_count):      result.append(values[i] \* (i + 1))  print(" ".join(map(str, result))) |

Tabel 6. Source Code Soal 2 pada Bahasa Pemrograman Python

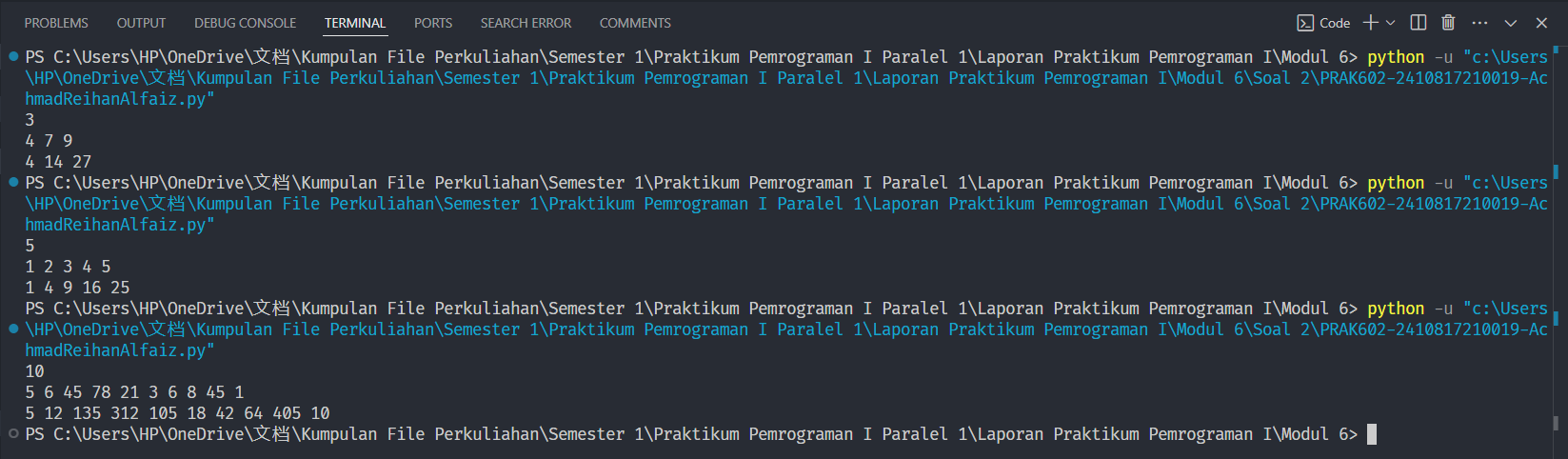
## Output Program

### **Output Program C**



Gambar 3. Screenshot Hasil Jawaban Soal 2 pada Bahasa Pemrograman C

### **Output Program Python**



Gambar 4. Screenshot Hasil Jawaban Soal 2 pada Bahasa Pemrograman Python

## Pembahasan

### **Pembahasan Code/Program C**

**Line 1:** Baris pertama ini menyertakan library standar input-output di C (<stdio.h>) yang menyediakan fungsi untuk melakukan operasi input dan output, seperti printf untuk mencetak teks ke layar dan fgets untuk membaca input dari pengguna.

**Line 3:** Fungsi main() adalah titik masuk utama dari program. Fungsi main() merupakan standar dalam bahasa C yang menandakan bahwa program akan dimulai dari fungsi ini saat dieksekusi.

**Line 5–6:** valueCount: Menyimpan jumlah bilangan yang akan dimasukkan oleh pengguna. scanf("%d", &valueCount); scanf("%d", &valueCount): Membaca sebuah bilangan bulat dari pengguna dan menyimpannya ke dalam variabel valueCount.

**Line 8:** result: Array dinamis yang panjangnya ditentukan oleh nilai valueCount. Array ini digunakan untuk menyimpan bilangan yang telah dimodifikasi.

**Line 10–11:** Perulangan for: Perulangan berjalan dari i = 0 hingga i < valueCount. Membaca Nilai: scanf("%d", &result[i]) Membaca bilangan ke-i dari pengguna dan menyimpannya dalam array result di indeks i. Mengalikan dengan Indeks: result[i] \*= (i + 1); Bilangan pada indeks i dikalikan dengan i + 1 (indeks dimulai dari 1, bukan 0). Contoh: Jika valueCount = 5 dan input adalah 1 2 3 4 5: Iterasi 1 (i = 0): result[0] = 1 \* (0 + 1) = 1 Iterasi 2 (i = 1): result[1] = 2 \* (1 + 1) = 4 Iterasi 3 (i = 2): result[2] = 3 \* (2 + 1) = 9 Iterasi 4 (i = 3): result[3] = 4 \* (3 + 1) = 16 Iterasi 5 (i = 4): result[4] = 5 \* (4 + 1) = 25.

**Line 13–14:** Perulangan for: Perulangan berjalan dari i = 0 hingga i < valueCount. Mencetak Nilai: printf("%d ", result[i]); Setiap elemen dalam array result dicetak ke layar, dipisahkan oleh spasi.

**Line 16:** Baris terakhir return 0; digunakan untuk menunjukkan bahwa program telah selesai dijalankan tanpa kesalahan. Di C, nilai 0 biasanya menandakan eksekusi sukses, sementara nilai non-zero dapat menunjukkan bahwa ada masalah yang terjadi.

### **Pembahasan Code/Program Python**

**Line 1:** input(): Membaca input dari pengguna dalam bentuk string. int(): Mengonversi input string menjadi bilangan bulat. Variabel value\_count: Menyimpan jumlah bilangan yang akan diproses.

**Line 2:** input(): Membaca satu baris input dari pengguna, yang berisi bilangan-bilangan dipisahkan oleh spasi. split(): Memisahkan string input berdasarkan spasi menjadi beberapa elemen. map(int, ...): Mengonversi setiap elemen string menjadi bilangan bulat. list(): Mengubah hasil map menjadi daftar Python. Variabel values: Menyimpan semua bilangan dalam bentuk daftar.

**Line 3–6:** result = []: Inisialisasi daftar kosong untuk menyimpan hasil perkalian. Perulangan for i: Iterasi dari i = 0 hingga i = value\_count - 1, untuk mengakses setiap elemen dalam daftar values. Mengakses Elemen ke-i: values[i] Mengambil elemen ke-i dari daftar values. Mengalikan dengan Indeks: values[i] \* (i + 1) Elemen ke-i dikalikan dengan i + 1 (indeks dimulai dari 1, bukan 0). Menambahkan ke result: result.append(...) Hasil perkalian ditambahkan ke daftar result. Contoh: Jika values = [1, 2, 3, 4, 5]: Iterasi 1 (i = 0): result = [1 \* (0 + 1)] = [1] Iterasi 2 (i = 1): result = [1, 2 \* (1 + 1)] = [1, 4] Iterasi 3 (i = 2): result = [1, 4, 3 \* (2 + 1)] = [1, 4, 9] Iterasi 4 (i = 3): result = [1, 4, 9, 4 \* (3 + 1)] = [1, 4, 9, 16] Iterasi 5 (i = 4): result = [1, 4, 9, 16, 5 \* (4 + 1)] = [1, 4, 9, 16, 25]

**Line 8:** map(str, result): Mengonversi setiap elemen dalam daftar result menjadi string. " ".join(...): Menggabungkan elemen-elemen string dalam daftar menjadi satu string, dipisahkan oleh spasi. print(): Mencetak string hasil gabungan ke layar.

# SOAL 3

Ruli adalah seorang guru tik dan matematika di sebuah smp rumah bangsa. Ruli ingin membuat perkalian bilangan antara baris 1 dan baris 2.

Misal di beri angka n1=2 dan n2=2 maka inputan baris selanjutnya 2 kali untuk baris pertama dan 2 kali juga untuk baris kedua. Lalu angka untuk baris pertama 2 dan 3 selanjutnya angka untuk baris kedua 4 dan 5. Jadi perkaliannya itu baris1 kolom1 dikali baris2 kolom1 selanjutnya baris1 kolom2 dikali baris2 kolom2. Berikut rumusnya :

2

3

4

5

Jadi hasilnya itu 8 15

Format input :

bilangan n1 dan n2; jika n1 dan n2 tidak sama maka menampilkan jumlah tidak sama. jika n1 dan n2 sama maka inputan angka sebanyak 2\*n1 atau 2\*n2.

Format Output :

hasil output adalah baris pertama di kali baris ke dua, jadi liat contoh input output untuk lebih jelas

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 3  1 2 3  4 5 6 | 4 10 18 |
| 2 3 | Jumlah tidak sama |
| 5 5  1 2 3 4 5  5 4 3 2 1 | 5 8 9 8 5 |

Tabel 7. Soal Nomor 3

## Source Code

### **Source Code C**

|  |  |
| --- | --- |
| 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 | #include <stdio.h>  int main()  {       int n1, n2;       scanf("%d %d", &n1, &n2);       if (n1 != n2)       {            printf("Jumlah tidak sama\n");            return 0;       }       int row1[n1], row2[n2];       for (int i = 0; i < n1; i++)            scanf("%d", &row1[i]);       for (int i = 0; i < n2; i++)            scanf("%d", &row2[i]);       for (int i = 0; i < n1; i++)            printf("%d ", row1[i] \* row2[i]);       return 0;  } |

Tabel 8. Source Code Soal 3 pada Bahasa Pemrograman C

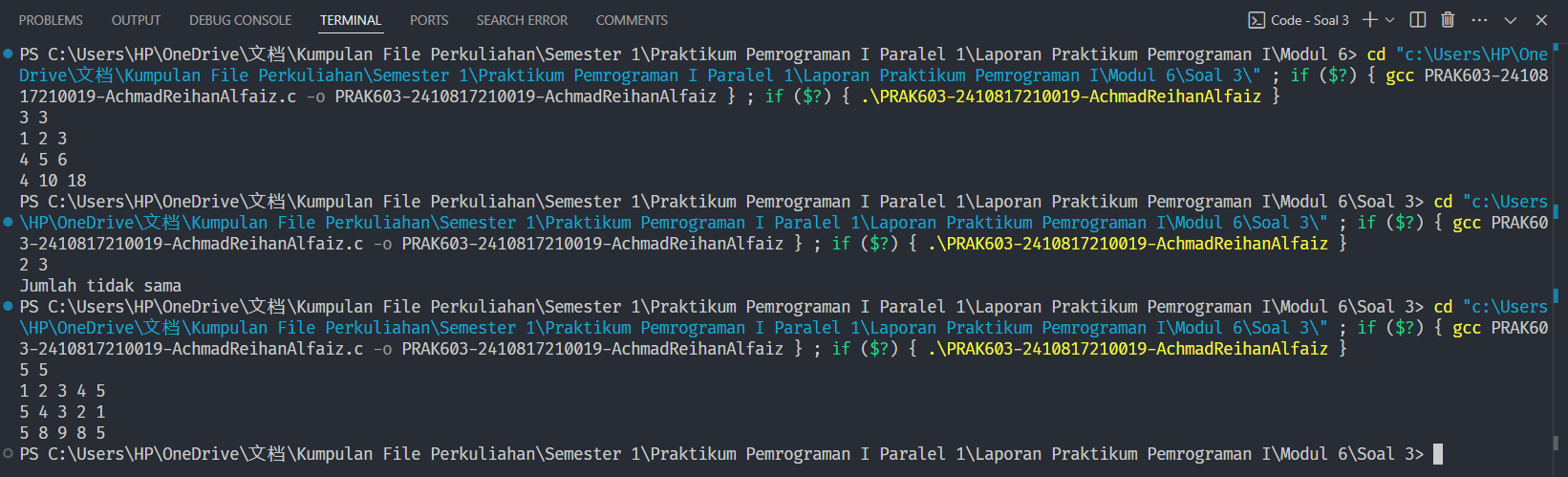
### **Source Code Python**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | n1, n2 = map(int, input().split())  if n1 != n2:      print("Jumlah tidak sama")  else:      row1 = list(map(int, input().split()))      row2 = list(map(int, input().split()))      print(" ".join(str(row1[i] \* row2[i]) for i in range(n1))) |

Tabel 9. Source Code Soal 3 pada Bahasa Pemrograman Python

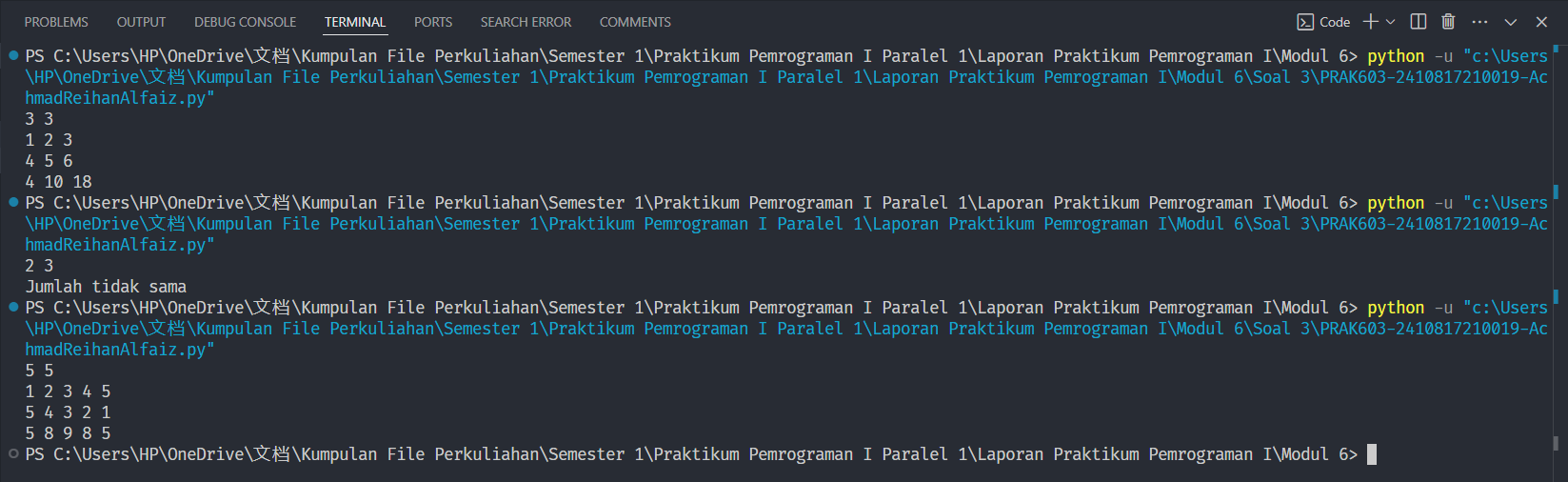
## Output Program

### **Output Program C**



Gambar 5. Screenshot Hasil Jawaban Soal 3 pada Bahasa Pemrograman C

### **Output Program Python**



Gambar 6. Screenshot Hasil Jawaban Soal 3 pada Bahasa Pemrograman Python

## Pembahasan

### **Pembahasan Code/Program C**

**Line 1:** Baris pertama ini menyertakan library standar input-output di C (<stdio.h>) yang menyediakan fungsi untuk melakukan operasi input dan output, seperti printf untuk mencetak teks ke layar dan fgets untuk membaca input dari pengguna.

**Line 3:** Fungsi main() adalah titik masuk utama dari program. Fungsi main() merupakan standar dalam bahasa C yang menandakan bahwa program akan dimulai dari fungsi ini saat dieksekusi.

**Line 5–6:** n1: Menyimpan jumlah elemen array pertama. n2: Menyimpan jumlah elemen array kedua. scanf: Membaca dua bilangan bulat dari pengguna dan menyimpannya di variabel n1 dan n2.

**Line 8–12:** Kondisi if (n1 != n2): Memeriksa apakah jumlah elemen array pertama (n1) tidak sama dengan jumlah elemen array kedua (n2). Pesan Kesalahan: Jika jumlah elemen tidak sama, program mencetak: Jumlah tidak sama Kemudian menghentikan eksekusi dengan return 0. Contoh: Jika pengguna memasukkan: 3 4 Output akan menjadi: Jumlah tidak sama.

**Line 14:** row1: Array dinamis untuk menyimpan elemen-elemen array pertama dengan panjang n1. row2: Array dinamis untuk menyimpan elemen-elemen array kedua dengan panjang n2.

**Line 16–20:** Perulangan for: Iterasi pertama membaca elemen-elemen array pertama (row1). Iterasi kedua membaca elemen-elemen array kedua (row2). Contoh Input: Jika pengguna memasukkan:

1 2 3

4 5 6

Maka: row1 = [1, 2, 3] row2 = [4, 5, 6].

**Line 22–23:** Perulangan for: Perulangan berjalan dari i = 0 hingga i < n1. Mengakses Elemen yang Bersesuaian: row1[i] \* row2[i] Elemen ke-i dari row1 dikalikan dengan elemen ke-i dari row2. Mencetak Hasil: Hasil perkalian langsung dicetak dengan spasi antar elemen. Contoh: Jika row1 = [1, 2, 3] dan row2 = [4, 5, 6]: Iterasi 1 (i = 0): 1 \* 4 = 4 Iterasi 2 (i = 1): 2 \* 5 = 10 Iterasi 3 (i = 2): 3 \* 6 = 18 Output: 4 10 18

**Line 25:** Baris terakhir return 0; digunakan untuk menunjukkan bahwa program telah selesai dijalankan tanpa kesalahan. Di C, nilai 0 biasanya menandakan eksekusi sukses, sementara nilai non-zero dapat menunjukkan bahwa ada masalah yang terjadi.

### **Pembahasan Code/Program Python**

**Line 1:** input(): Membaca satu baris input dari pengguna, berupa dua bilangan bulat yang dipisahkan oleh spasi. map(int, ...): Mengonversi kedua bilangan dalam input string menjadi bilangan bulat. Variabel n1 dan n2: n1: Menyimpan jumlah elemen array pertama. n2: Menyimpan jumlah elemen array kedua.

**Line 3–4:** Kondisi if n1 != n2: Mengecek apakah jumlah elemen pada array pertama (n1) berbeda dari jumlah elemen pada array kedua (n2). Pesan Kesalahan: Jika jumlah elemen tidak sama, program mencetak: Jumlah tidak sama Dan berhenti karena tidak ada logika lebih lanjut setelah if. Contoh: Jika pengguna memasukkan: 3 4 Outputnya: Jumlah tidak sama.

**Line 6–7:** row1 = list(map(int, input().split())) row2 = list(map(int, input().split())). input(): Membaca satu baris input dari pengguna untuk setiap array, dengan elemen-elemen array dipisahkan oleh spasi. map(int, ...): Mengonversi setiap elemen input string menjadi bilangan bulat. list(...): Mengubah hasil map menjadi daftar Python. Variabel row1 dan row2: row1: Menyimpan elemen-elemen array pertama. row2: Menyimpan elemen-elemen array kedua. Contoh Input: Jika pengguna memasukkan:

1 2 3

4 5 6

Maka: row1 = [1, 2, 3]. row2 = [4, 5, 6].

**Line 8:** range(n1): Menghasilkan indeks dari 0 hingga n1 - 1, untuk mengakses elemen-elemen dari kedua array. Perulangan for i in range(n1): Mengiterasi melalui setiap indeks dari array, untuk mengakses elemen bersesuaian pada row1 dan row2. Perkalian Elemen: row1[i] \* row2[i] Mengalikan elemen pada indeks i dari row1 dengan elemen pada indeks i dari row2. Mengonversi ke String: str(row1[i] \* row2[i]) Hasil perkalian dikonversi menjadi string agar dapat digabungkan dengan join. Menggabungkan Hasil dengan Spasi: " ".join(...) Menggabungkan elemen-elemen string hasil perkalian dengan spasi di antaranya. Mencetak Hasil: Hasil akhir langsung dicetak dalam satu baris. Contoh: Jika row1 = [1, 2, 3] dan row2 = [4, 5, 6]: Iterasi 1 (i = 0): 1 \* 4 = 4 Iterasi 2 (i = 1): 2 \* 5 = 10 Iterasi 3 (i = 2): 3 \* 6 = 18 Output: 4 10 18.

# SOAL 4

Shikamaru merupakan seorang anggota anbu Desa Konoha. Pada saat memata-matai Desa Iwagakure, Shikamaru harus memecahkan sebuah kode untuk memastikan keaslian pesan dari rekannya. Kode merupakan rangkaian karakter yang harus dicocokkan dengan rangkaian karakter yang dimiliki oleh Shikamaru. Karakter yang sama pada kode tersebut akan berubah menjadi tanda bintang (\*) sedangkan karakter yang berbeda akan berubah menjadi tanda (#). Pesan yang asli merupakan pesan yang memiliki jumlah bintang yang lebih dari atau sama dengan jumlah pagar, sedangkan pesan yang palsu merupakan pesan yang memiliki jumlah bintang yang kurang dari jumlah pagar atau panjang karakter tidak sama.

Buatlah sebuah program untuk membantu Shikamaru agar mengetahui pesan yang diterima merupakan pesan asli atau pesan palsu.

Format input:

Baris pertama merupakan kode yang dimiliki oleh Shikamaru Baris kedua merupakan pesan yang diterima oleh Shikamaru Format output:

Baris pertama merupakan rangkaian karakter yang telah berubah menjadi tanda bintang (\*) atau pagar (#)

Baris kedua merupakan jumlah tanda bintang Baris ketiga merupakan jumlah tanda pagar

Baris keempat merupakan pemberitahuan pesan asli atau pesan palsu

|  |  |
| --- | --- |
| **Input** | **Output** |
| Bahasa  Pemrograman | Panjang kalimat berbeda, pesan palsu |
| Ini Pesan Rahasia  1ni p354n Rahas14 | #\*\* ####\* \*\*\*\*\*##  \* = 8  # = 7  Pesan Asli |
| Aku Pasti Bisa  Berjuang lebih | #####\*########  \* = 1  # = 13  Pesan Palsu |

Tabel 10. Soal Nomor 4

## Source Code

### **Source Code C**

|  |  |
| --- | --- |
| 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 | #include <stdio.h>  #include <string.h>  #include <ctype.h>  int main()  {       char code[100], message[100];       int starCount = 0, hashCount = 0;       fgets(code, sizeof(code), stdin);       fgets(message, sizeof(message), stdin);       code[strcspn(code, "\n")] = 0;       message[strcspn(message, "\n")] = 0;       if (strlen(code) != strlen(message))       {            printf("Panjang kalimat berbeda, pesan palsu\n");            return 0;       }       for (int i = 0; code[i]; i++)       {            if (isspace(code[i]) && isspace(message[i]))                 putchar(' ');            else                 putchar(code[i] == message[i] ? '\*' : '#'),                     (code[i] == message[i]) ? starCount++ : hashCount++;       }       printf("\n\* = %d\n# = %d\n%s\n", starCount, hashCount,              starCount >= hashCount ? "Pesan Asli" : "Pesan Palsu");       return 0;  } |

Tabel 11. Source Code Soal 4 pada Bahasa Pemrograman C

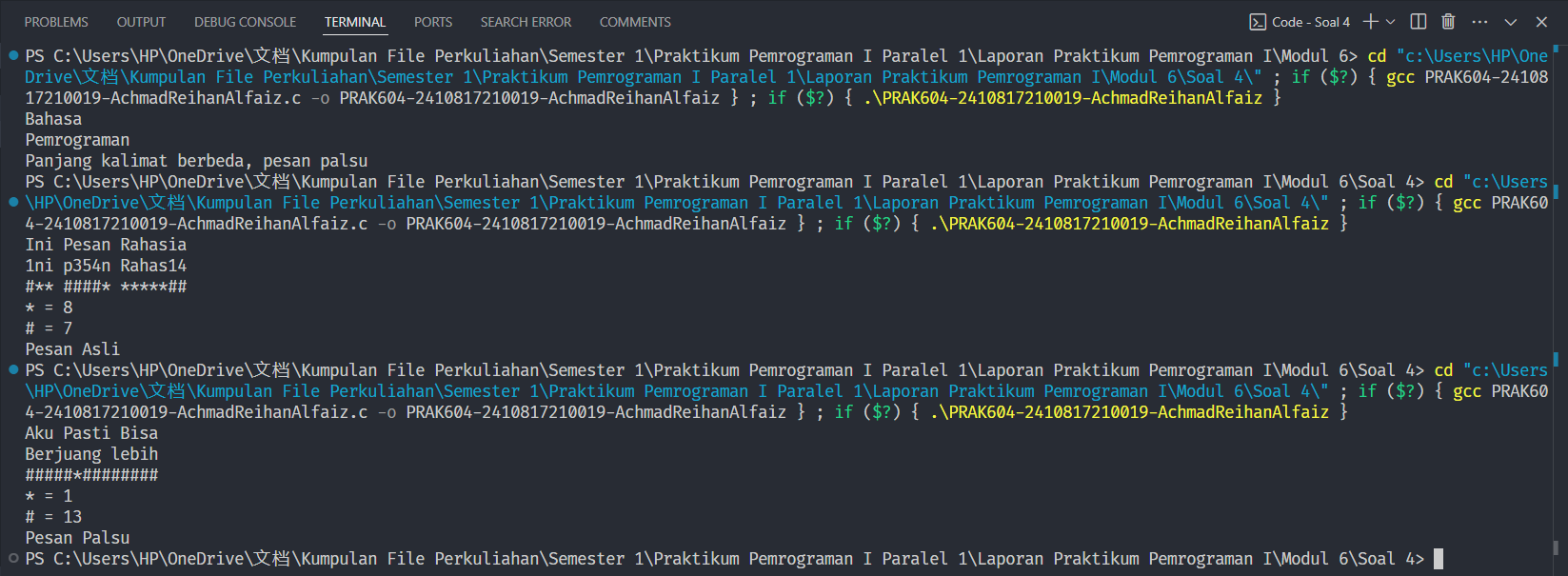
### **Source Code Python**

|  |  |
| --- | --- |
| 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 | code = input()  message = input()  code = code.rstrip("\n")  message = message.rstrip("\n")  if len(code) != len(message):      print("Panjang kalimat berbeda, pesan palsu")      exit()  star\_count = 0  hash\_count = 0  result = []  for c, m in zip(code, message):      if c.isspace() and m.isspace():          result.append(" ")      else:          result.append("\*" if c == m else "#")          if c == m:              star\_count += 1          else:              hash\_count += 1  print("".join(result))  print(f"\* = {star\_count}")  print(f"# = {hash\_count}")  print("Pesan Asli" if star\_count >= hash\_count else "Pesan Palsu") |

Tabel 12. Source Code Soal 4 pada Bahasa Pemrograman Python

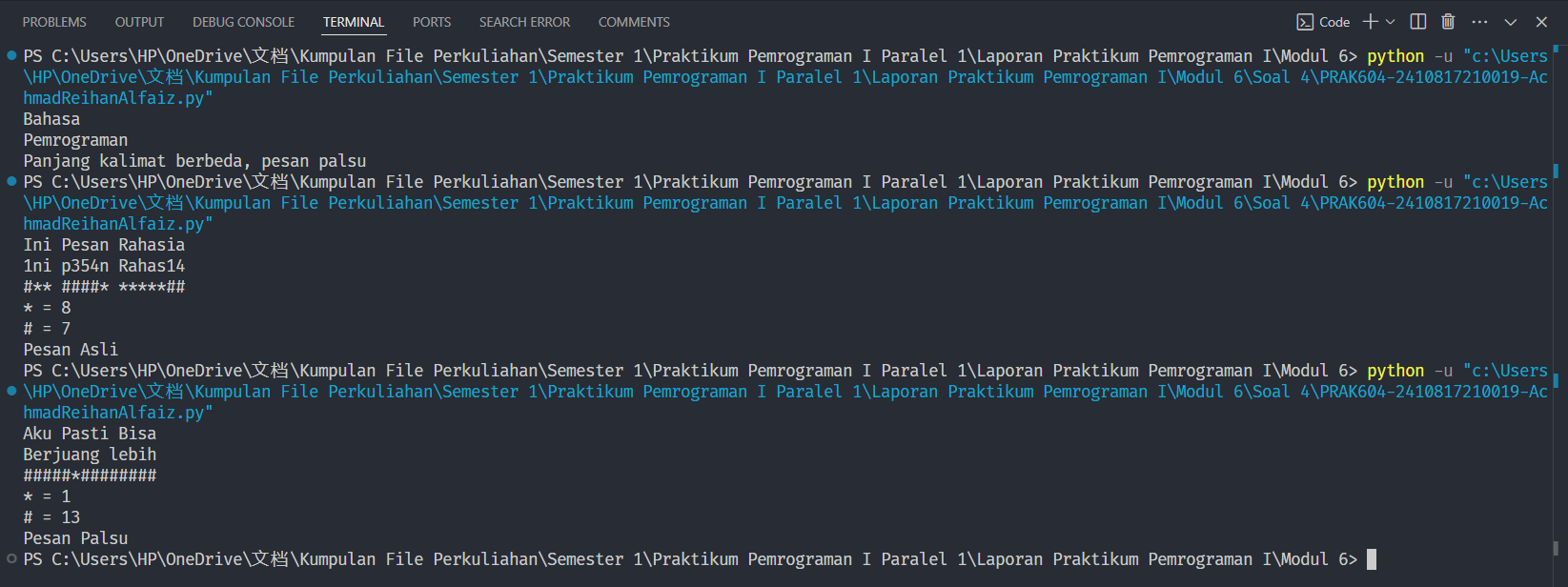
## Output Program

### **Output Program C**



Gambar 7. Screenshot Hasil Jawaban Soal 4 pada Bahasa Pemrograman C

### **Output Program Python**



Gambar 8. Screenshot Hasil Jawaban Soal 4 pada Bahasa Pemrograman Python

## Pembahasan

### **Pembahasan Code/Program C**

**Line 1:** Baris pertama ini menyertakan library standar input-output di C (<stdio.h>) yang menyediakan fungsi untuk melakukan operasi input dan output, seperti printf untuk mencetak teks ke layar dan fgets untuk membaca input dari pengguna.

**Line 2:** Baris ini menyertakan library *string.h* di C (#include <string.h>) yang menyediakan fungsi-fungsi untuk manipulasi string, seperti strlen untuk menghitung panjang string dan strcspn untuk mencari posisi karakter pertama dalam string yang cocok dengan karakter tertentu, lalu mengembalikan indeksnya.

**Line 3:** Baris ini menyertakan library *ctype.h* di C (#include <ctype.h>) yang menyediakan fungsi-fungsi untuk memeriksa dan memanipulasi karakter, seperti isspace untuk memeriksa apakah karakter adalah spasi atau karakter whitespace lainnya (seperti tab atau newline).

**Line 5:** Fungsi main() adalah titik masuk utama dari program. Fungsi main() merupakan standar dalam bahasa C yang menandakan bahwa program akan dimulai dari fungsi ini saat dieksekusi.

**Line 7–8:** code dan message: Menyimpan dua string yang akan dibandingkan. Maksimum panjang string adalah 99 karakter (karena buffer dialokasikan untuk 100 karakter termasuk null-terminator \0). starCount: Menyimpan jumlah karakter yang sama (ditandai dengan '\*'). hashCount: Menyimpan jumlah karakter yang berbeda (ditandai dengan '#').

**Line 10–11:** fgets: Membaca input string termasuk spasi, hingga panjang maksimum buffer (99 karakter) atau hingga ditemukan newline (\n). sizeof(code): Membatasi jumlah karakter yang dibaca ke panjang buffer code (yaitu 100). Contoh Input: Jika pengguna memasukkan:

secret code

secret c0de

Maka: code = "secret code\n" message = "secret c0de\n".

**Line 12–13:** strcspn(code, "\n"): Mencari posisi pertama dari karakter newline (\n) dalam string code. Menghapus Newline: Mengganti karakter \n dengan null-terminator \0, sehingga string menjadi bersih dari newline. Contoh: Setelah operasi ini: code = "secret code" message = "secret c0de".

**Line 15–19:** strlen: Menghitung panjang string (tidak termasuk null-terminator \0). Kondisi if: Jika panjang code dan message berbeda, program langsung mencetak: Panjang kalimat berbeda, pesan palsu. Dan berhenti dengan return 0.

**Line 21–28:** Perulangan for: Iterasi melalui setiap karakter dalam code dan message. Memeriksa Spasi: if (isspace(code[i]) && isspace(message[i]))

putchar(' '); Jika kedua karakter pada posisi i adalah spasi, mencetak ' ' (spasi). Membandingkan Karakter: putchar(code[i] == message[i] ? '\*' : '#'); Jika karakter code[i] sama dengan message[i], mencetak '\*'. Jika berbeda, mencetak '#'. Menghitung '\*' dan '#': (code[i] == message[i]) ? starCount++ : hashCount++; Menambahkan 1 ke starCount jika sama, atau ke hashCount jika berbeda.

**Line 30–31:** Jumlah '\*' dan '#': Program mencetak jumlah karakter yang sama (starCount) dan jumlah karakter yang berbeda (hashCount). Keputusan Pesan: starCount >= hashCount ? "Pesan Asli" : "Pesan Palsu" Jika jumlah karakter yang sama ('\*') lebih besar atau sama dengan jumlah karakter yang berbeda ('#'), maka pesan dianggap "Asli". Jika tidak, pesan dianggap "Palsu".

**Line 33:** Baris terakhir return 0; digunakan untuk menunjukkan bahwa program telah selesai dijalankan tanpa kesalahan. Di C, nilai 0 biasanya menandakan eksekusi sukses, sementara nilai non-zero dapat menunjukkan bahwa ada masalah yang terjadi.

### **Pembahasan Code/Program Python**

**Line 1–2:** input(): Membaca dua baris input dari pengguna. Variabel code dan message: code: Menyimpan string pertama (kode asli). message: Menyimpan string kedua (pesan yang akan dibandingkan).

**Line 4–5:** rstrip("\n"): Menghapus karakter newline (\n) di akhir string, jika ada.

**Line 7–9:** len(): Menghitung panjang string. Kondisi if: Jika panjang code tidak sama dengan panjang message, program mencetak: Panjang kalimat berbeda, pesan palsu Dan menghentikan eksekusi dengan exit().

**Line 11–13:** star\_count: Menyimpan jumlah karakter yang sama ('\*'). hash\_count: Menyimpan jumlah karakter yang berbeda ('#'). result: Daftar kosong untuk menyimpan hasil perbandingan karakter demi karakter dalam bentuk '\*', '#', atau spasi.

**Line 15–23:** zip(code, message): Menggabungkan dua string menjadi pasangan karakter bersesuaian. Contoh: code = "abc" message = "abd" Hasil: zip(code, message) → [('a', 'a'), ('b', 'b'), ('c', 'd')] Perbandingan Spasi: if c.isspace() and m.isspace():

result.append(" ")

Jika kedua karakter adalah spasi, tambahkan spasi (' ') ke daftar result. Perbandingan Karakter Lain: result.append("\*" if c == m else "#"). Jika karakter sama, tambahkan '\*' ke result. Jika berbeda, tambahkan '#'. Menghitung '\*' dan '#':

if c == m:

star\_count += 1

else:

hash\_count += 1

Jika karakter sama ('\*'), tambahkan 1 ke star\_count. Jika karakter berbeda ('#'), tambahkan 1 ke hash\_count.

**Line 25–28:**Mencetak Hasil Perbandingan: print("".join(result)) Menggabungkan elemen dalam daftar result menjadi satu string, lalu mencetaknya. Mencetak Jumlah '\*' dan '#':

print(f"\* = {star\_count}")

print(f"# = {hash\_count}")

Menampilkan jumlah karakter yang sama (star\_count) dan jumlah karakter yang berbeda (hash\_count). Keputusan Pesan: print("Pesan Asli" if star\_count >= hash\_count else "Pesan Palsu") Jika jumlah '\*' lebih besar atau sama dengan '#', pesan dianggap "Asli". Jika tidak, pesan dianggap "Palsu".

# SOAL 5

Buatlah program untuk menghitung hasil dari perkalian 2 buah matriks persegi.

Format input:

Baris pertama merupakan ordo matriks n×n

Baris kedua merupakan isi matriks A

Baris ketiga merupakan isi matriks B

Format output:

Output merupakan hasil kali dari matriks A×B

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  Matriks A  1 2  3 4  Matriks B  1 2  3 4 | Matriks AXB  7 10  15 22 |
| 3  Matriks A  1 2 3  3 4 5  6 7 8  Matriks B  1 2 3  3 4 5  6 7 8 | Matriks AXB  25 31 37  45 57 69  75 96 117 |

Tabel 13. Soal Nomor 5

## Source Code

### **Source Code C**

|  |  |
| --- | --- |
| 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  50 | #include <stdio.h>  int main()  {       int n;       scanf("%d", &n);       int A[n][n], B[n][n], C[n][n];       printf("Matriks A\n");       for (int i = 0; i < n; i++)       {            for (int j = 0; j < n; j++)            {                 scanf("%d", &A[i][j]);            }       }       printf("Matriks B\n");       for (int i = 0; i < n; i++)       {            for (int j = 0; j < n; j++)            {                 scanf("%d", &B[i][j]);            }       }       for (int i = 0; i < n; i++)       {            for (int j = 0; j < n; j++)            {                 C[i][j] = 0;                 for (int k = 0; k < n; k++)                 {                      C[i][j] += A[i][k] \* B[k][j];                 }            }       }       printf("Matriks AxB:\n");       for (int i = 0; i < n; i++)       {            for (int j = 0; j < n; j++)            {                 printf("%d ", C[i][j]);            }            printf("\n");       }       return 0;  } |

Tabel 14. Source Code Soal 5 pada Bahasa Pemrograman C

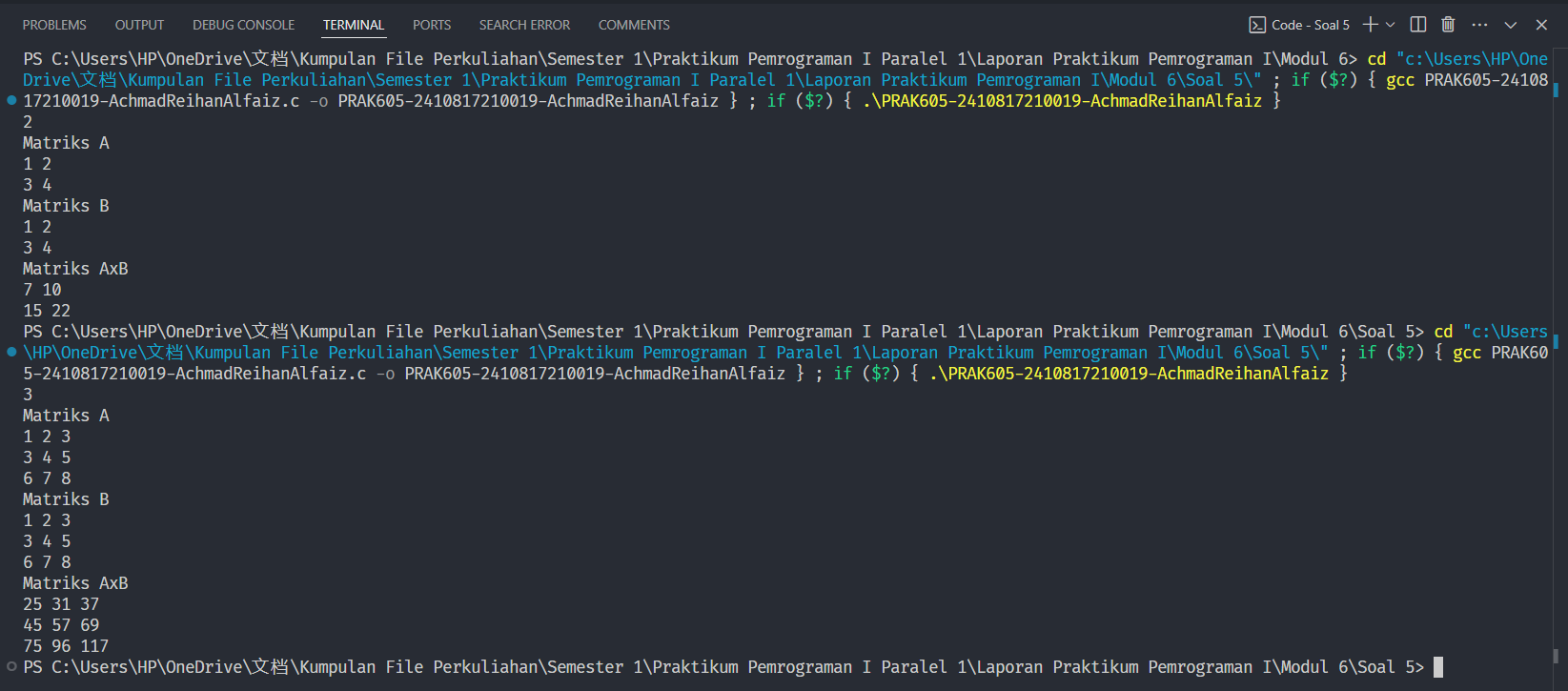
### **Source Code Python**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | n = int(input())  A = []  B = []  C = [[0] \* n for \_ in range(n)]  print("Matriks A")  for i in range(n):      row = list(map(int, input().split()))      A.append(row)  print("Matriks B")  for i in range(n):      row = list(map(int, input().split()))      B.append(row)  for i in range(n):      for j in range(n):          for k in range(n):              C[i][j] += A[i][k] \* B[k][j]  print("Matriks AxB")  for row in C:      print(" ".join(map(str, row))) |

Tabel 15. Source Code Soal 5 pada Bahasa Pemrograman Python

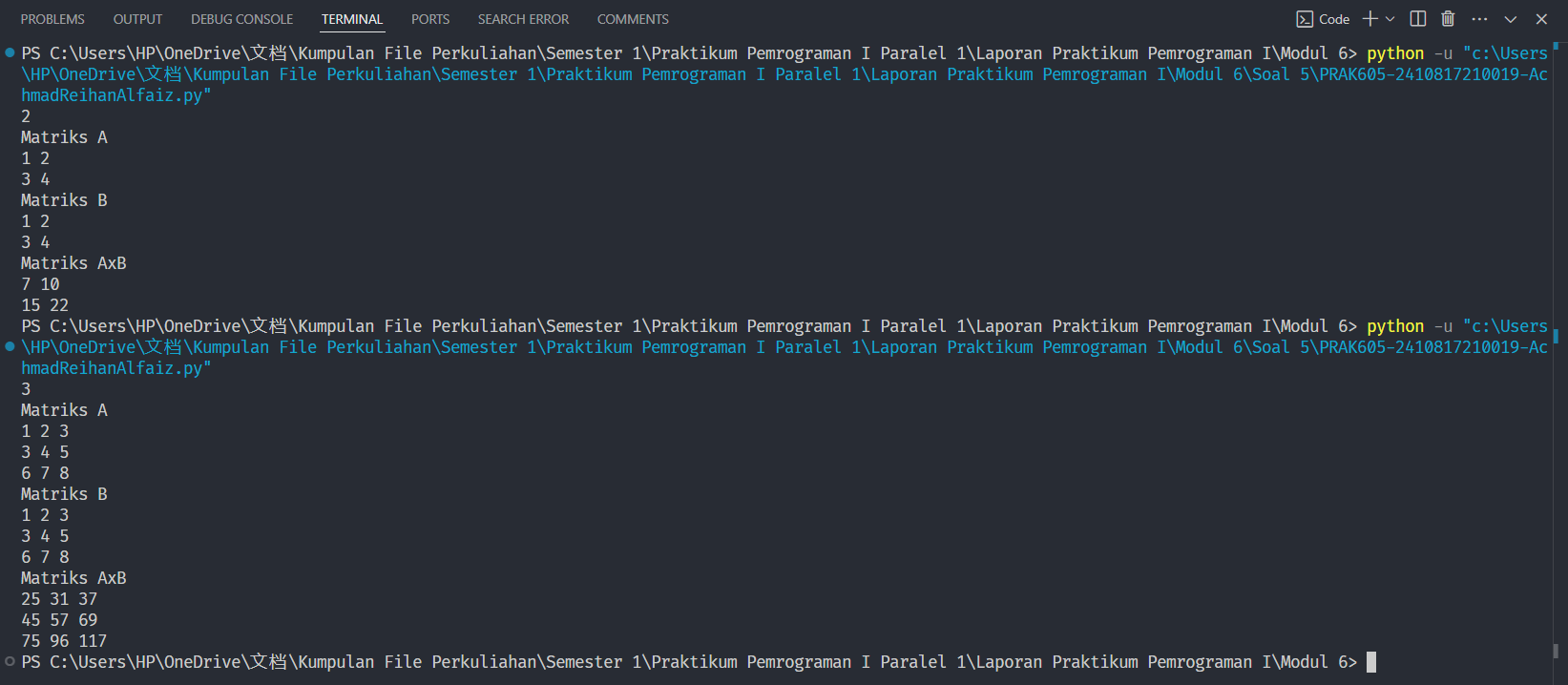
## Output Program

### **Output Program C**



Gambar 9. Screenshot Hasil Jawaban Soal 5 pada Bahasa Pemrograman C

### **Output Program Python**



Gambar 10. Screenshot Hasil Jawaban Soal 5 pada Bahasa Pemrograman Python

## Pembahasan

### **Pembahasan Code/Program C**

**Line 1:** Baris pertama ini menyertakan library standar input-output di C (<stdio.h>) yang menyediakan fungsi untuk melakukan operasi input dan output, seperti printf untuk mencetak teks ke layar dan fgets untuk membaca input dari pengguna.

**Line 3:** Fungsi main() adalah titik masuk utama dari program. Fungsi main() merupakan standar dalam bahasa C yang menandakan bahwa program akan dimulai dari fungsi ini saat dieksekusi.

**Line 5–8:** n: Menyimpan ukuran matriks persegi (n × n), yang ditentukan oleh input pengguna. A[n][n], B[n][n], C[n][n]: Matriks A dan B menyimpan elemen-elemen input dari pengguna. Matriks C digunakan untuk menyimpan hasil perkalian A × B.

**Line 10–17:** Perulangan Bersarang: Iterasi pertama (for i) mengakses setiap baris i pada matriks. Iterasi kedua (for j) mengakses setiap elemen dalam baris i, yaitu kolom j. scanf: Membaca elemen-elemen matriks baris per baris.

**Line 18–25:** Prosesnya identik dengan membaca Matriks A, tetapi elemen-elemen disimpan dalam Matriks B.

**Line 27–37:** Perulangan Bersarang: for i: Mengakses baris i dari Matriks A dan Matriks C. for j: Mengakses kolom j dari Matriks B dan Matriks C. for k: Menghitung hasil perkalian Matriks A dan Matriks B untuk elemen C[i][j]. Proses Perkalian: Matriks A dan B dikalikan sesuai definisi perkalian matriks: Elemen C[i][j] dihitung sebagai:

**Line 39–47:** Perulangan Bersarang: Iterasi pertama (for i) mengakses setiap baris dari Matriks C. Iterasi kedua (for j) mencetak elemen-elemen dalam baris i. Output Matriks C: Hasil perkalian Matriks A dan B dicetak baris per baris.

**Line 49:** Baris terakhir return 0; digunakan untuk menunjukkan bahwa program telah selesai dijalankan tanpa kesalahan. Di C, nilai 0 biasanya menandakan eksekusi sukses, sementara nilai non-zero dapat menunjukkan bahwa ada masalah yang terjadi.

### **Pembahasan Code/Program Python**

**Line 1:** input(): Membaca satu baris input dari pengguna. int(): Mengonversi input string menjadi bilangan bulat. Variabel n: Menyimpan ukuran matriks persegi (𝑛 × 𝑛).

**Line 3–5:** A dan B: Daftar kosong yang akan diisi dengan elemen-elemen Matriks A dan B. C: Matriks hasil perkalian, diinisialisasi dengan nilai nol. Struktur ini adalah daftar bersarang, dengan setiap elemen awalnya bernilai 0. List Comprehension: [[0] \* n for \_ in range(n)] Membuat matriks n × n, di mana setiap elemen awalnya adalah 0.

**Line 7–10:** Perulangan for: Iterasi melalui setiap baris i dari Matriks A. Membaca Baris: row = list(map(int, input().split())) Membaca elemen-elemen dalam satu baris, dipisahkan oleh spasi, lalu mengonversi elemen-elemen tersebut menjadi bilangan bulat. Menambahkan ke Matriks A: A.append(row) Baris yang baru dibaca ditambahkan ke Matriks A.

**Line 12–15:** Prosesnya identik dengan membaca Matriks A, tetapi elemen-elemen disimpan dalam Matriks B.

**Line 17–20:** Perulangan Bersarang: for i: Mengakses setiap baris i dari Matriks A dan Matriks C. for j: Mengakses setiap kolom j dari Matriks B dan Matriks C. for k: Melakukan operasi perkalian dan penjumlahan elemen Matriks A dan Matriks B untuk menghitung elemen C[i][j]. Definisi Perkalian Matriks: Elemen C[i][j] dihitung dengan rumus:

**Line 22–24:** Perulangan for: Iterasi melalui setiap baris dalam Matriks C. " ".join(map(str, row)): Menggabungkan elemen-elemen dalam baris row menjadi string, dipisahkan oleh spasi. Mencetak Baris: Setiap baris hasil perkalian dicetak dalam format yang rapi.