**LAPORAN PRAKTIKUM**

**PEMROGRAMAN I**

**MODUL 6**

****

**Array**

**Oleh:**

**Putri Fatima Az’hara NIM. 2410817120001**

**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 Web II. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Putri Fatima Az’hara

NIM : 2410817120001

|  |  |
| --- | --- |
| 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 ii](#_Toc184997059)

[DAFTAR ISI iii](#_Toc184997060)

[DAFTAR GAMBAR iv](#_Toc184997061)

[DAFTAR TABEL v](#_Toc184997062)

[SOAL 1 6](#_Toc184997063)

[A. Source Code 6](#_Toc184997064)

[B. Output Program 7](#_Toc184997065)

[C. Pembahasan 9](#_Toc184997066)

[SOAL 2 11](#_Toc184997067)

[A. Source Code 12](#_Toc184997068)

[B. Output Program 13](#_Toc184997069)

[C. Pembahasan 14](#_Toc184997070)

[SOAL 3 16](#_Toc184997071)

[A. Source Code 17](#_Toc184997072)

[B. Output Program 18](#_Toc184997073)

[C. Pembahasan 20](#_Toc184997074)

[SOAL 4 23](#_Toc184997075)

[A. Source Code 24](#_Toc184997076)

[B. Output Program 24](#_Toc184997077)

[C. Pembahasan 26](#_Toc184997078)

[SOAL 5 28](#_Toc184997079)

[A. Source Code 29](#_Toc184997080)

[B. Output Program 29](#_Toc184997081)

[C. Pembahasan 31](#_Toc184997082)

# DAFTAR GAMBAR

[Gambar 1. Screenshot Hasil Jawaban Soal 1 C 8](#_Toc185015830)

[Gambar 2. Screenshot Hasil Jawaban Soal 1 Python 8](#_Toc185015831)

[Gambar 3. Screenshot Hasil Jawaban Soal 2 C 13](#_Toc185015832)

[Gambar 4. Screenshot Hasil Jawaban Soal 2 python 14](#_Toc185015833)

[Gambar 5. Screenshot Hasil Jawaban Soal 3 C 19](#_Toc185015834)

[Gambar 6. Screenshot Hasil Jawaban Soal 3 python 19](#_Toc185015835)

[Gambar 7. Screenshot Hasil Jawaban Soal 4 C 25](#_Toc185015836)

[Gambar 8. Screenshot Hasil Jawaban Soal 4 python 25](#_Toc185015837)

[Gambar 9. Screenshot Hasil Jawaban Soal 5 C 30](#_Toc185015838)

[Gambar 10. Screenshot Hasil Jawaban Soal 5 python 30](#_Toc185015839)

# DAFTAR TABEL

[Tabel 1Source Code Soal 1 C 7](#_Toc184998358)

[Tabel 2 Source Code Soal 1 Python 7](#_Toc184998359)

[Tabel 3 Source Code Soal 2 C 13](#_Toc184998360)

[Tabel 4 Source Code Soal 2 python 13](#_Toc184998361)

[Tabel 5 Source Code Soal 3 C 18](#_Toc184998362)

[Tabel 6 Source Code Soal 3 python 18](#_Toc184998363)

[Tabel 7 Source Code Soal 4 C 24](#_Toc184998364)

[Tabel 8 Source Code Soal 4 python 24](#_Toc184998365)

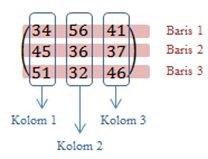
[Tabel 9 Source Code C Soal 5 C 29](#_Toc184998366)

[Tabel 10 Source Code Soal 5 python 29](#_Toc184998367)

# SOAL 1

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 |

## Source Code

* C

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

Tabel 1Source Code Soal 1 C

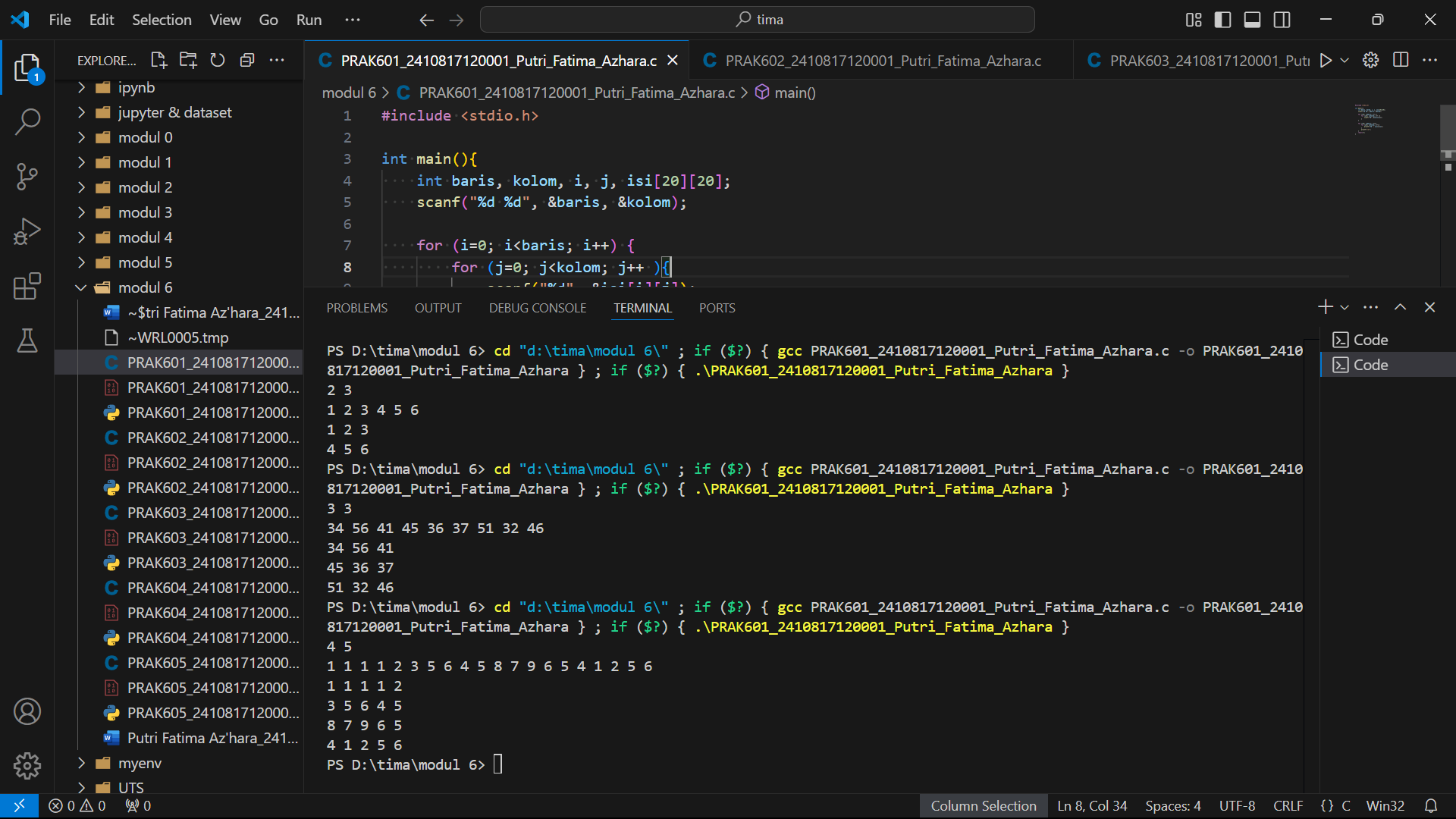
* Python

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | baris, kolom = map(int, input().split())  isi = list (map(int, input().split()))  angka = []  x = 0  for i in range (baris):      angka.append(isi[x:x+kolom])      x += kolom  for baris in angka:      for isi in baris:          print(isi, end=' ')      print() |

Tabel 2 Source Code Soal 1 Python

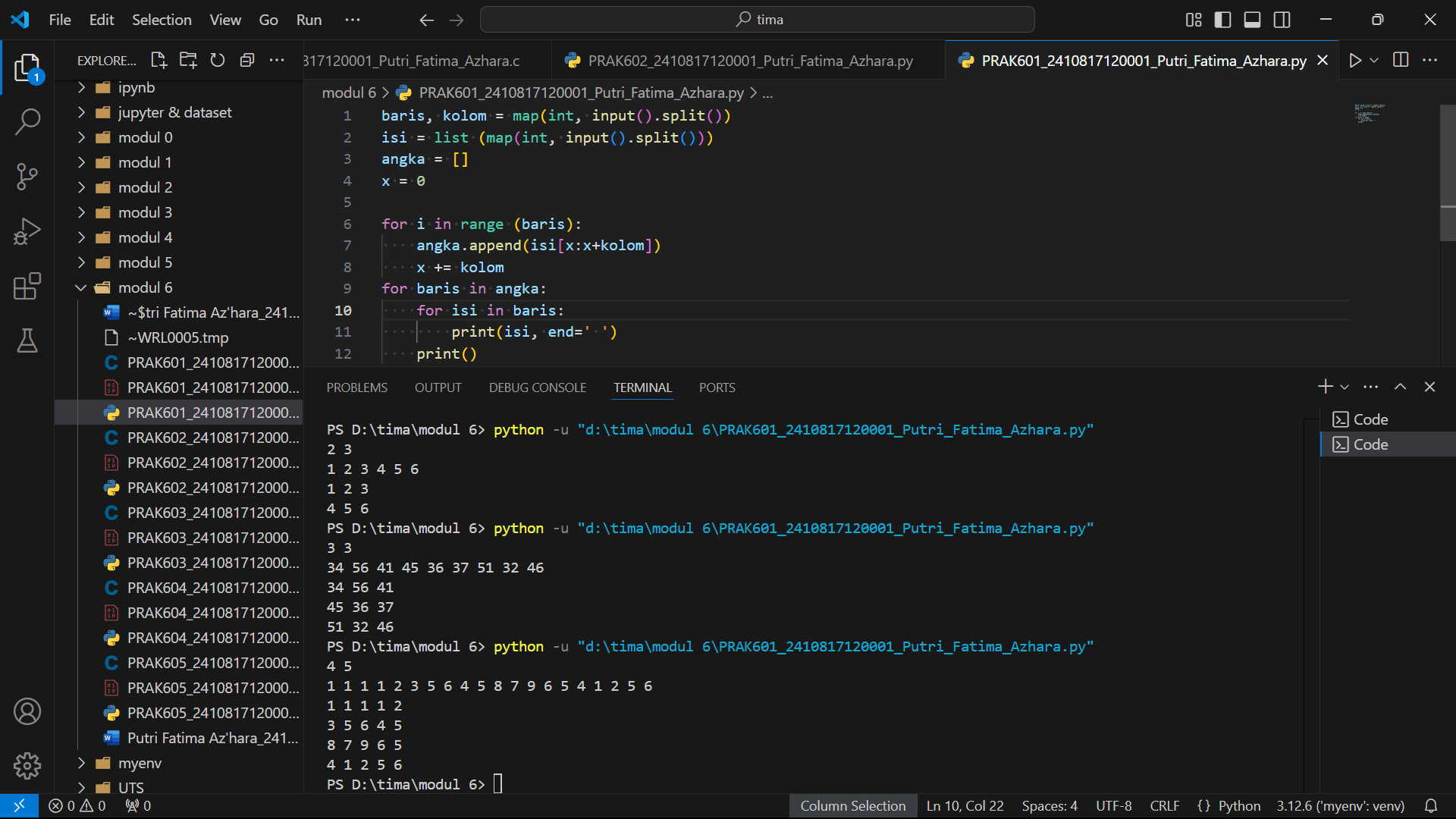
## Output Program

* C



Gambar 1. Screenshot Hasil Jawaban Soal 1 C

* Python



Gambar 2. Screenshot Hasil Jawaban Soal 1 Python

## Pembahasan

1. Bahasa C:

* #include <stdio.h> : Memasukkan file header stdio.h, yang berisi

deklarasi fungsi-fungsi untuk input dan output.

* int main() : Fungsi utama dalam program C, di mana eksekusi program

dimulai. Kata kunci int menunjukkan bahwa fungsi ini akan mengembalikan nilai integer.

* Tanda {} : Berisi kumpulan perintah (*statement*) agar dieksekusi

berurutan

* Int ; : Mendeklarasikan variabel baris, kolom, i, j, dan isi dengan tipe

data integer.

* isi[20][20] : Matriks 2 dimensi dengan kapasitas maksimum 20x20 elemen,

bertipe integer.

* Scanf () : Fungsi untuk membaca input dari pengguna.
* %d: Format spesifier untuk membaca integer.
* &baris: Input disimpan ke dalam variabel baris.
* &kolom : Input disimpan ke dalam variabel kolom.
* &isi[i][j] : Menunjukkan elemen matriks pada baris ke-i dan kolom ke-j.

Nilai ini diperoleh dari input pengguna.

* for() : Pernyataan loop yang digunakan untuk mengulang suatu blok

kode dengan jumlah iterasi tertentu.

* (i = 0; i < baris; i++) :1) Variabel i bertipe integer diinisialisasi dengan nilai 0

sebagai angka memulai loop.

2) Variabel i akan berhenti melakukan loop jika angka sudah sampai kurang dari nilai variabel baris.

3) Nilai i akan bertambah 1, dikarenakan Operator i++ adalah *shorthand* untuk i = i + 1.

* (j = 0; i < kolom; j++) :1) Variabel j bertipe integer diinisialisasi dengan nilai 0

sebagai angka memulai loop.

2) Variabel j akan berhenti melakukan loop jika angka sudah sampai kurang dari nilai variabel kolom.

3) Nilai i akan bertambah 1, dikarenakan Operator i++ adalah *shorthand* untuk i = i + 1.

* \n : Membuat baris baru.
* printf (“”); : Sebuah perintah (*statement*) untuk menampilkan output.
* return 0; : Mengembalikan nilai 0, yang menunjukan program

telah berjalan tanpa kesalahan (*error*).

1. Bahasa Python:

* baris, kolom = Input() :Berfungsi untuk mengambil input dari pengguna, kemudian

menyimpan nilai input pengguna di variabel baris dan kolom.

* isi = Input() :Berfungsi untuk mengambil input dari pengguna, kemudian

menyimpan nilai input pengguna di variabel baris

* split() :Berfungsi memisahkan input pengguna berdasarkan spasi.
* map() : mengubah setiap isi input yang dipisahkan spasi menjadi

bilangan bulat.

* list() : Mengubah hasil map () menjadi data list.
* angka = [] : Berfungsi untuk mendeklarasikan daftar kosong yang akan

menyimpan baris-baris matriks dalam bentuk dua dimensi.

* x = 0 : Berfungsi untuk menyimpan indeks awal elemen yang akan

diambil dari daftar isi.

* for: :Pernyataan loop yang digunakan untuk mengulang suatu blok

kode dengan jumlah iterasi tertentu.

* for i in range(baris) : Membuat daftar bilangan dari 0 hingga baris-1, digunakan

untuk iterasi sebanyak jumlah baris.

* angka.append() : Menambahkan hasil slicing tersebut ke dalam daftar angka

sebagai sebuah baris matriks.

* isi[x:x+kolom] : Mengambil elemen dari daftar isi mulai dari indeks x hingga

x+kolom.

* for baris in angka : Mengulang setiap elemen di dalam daftar angka yang berisi

baris-baris matriks.

* for isi in baris : Mengulang setiap elemen dalam baris matriks.
* end=' ' : Mengatur agar output dicetak dalam satu baris dengan spasi

antar elemen.

* print(“”) : Sebuah perintah (*statement*) untuk menampilkan output.

# SOAL 2

1. 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 |

## Source Code

* C

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | #include <stdio.h>  int main(){      int ruang, isi[20], i, hasil;      scanf("%d", &ruang);      for (i=0; i<ruang; i++){          scanf("%d",&isi[i]);      }      for (i=0; i<ruang; i++){          hasil = isi[i]\*(i+1);          printf("%d ", hasil);      }      return 0;  } |

Tabel 3 Source Code Soal 2 C

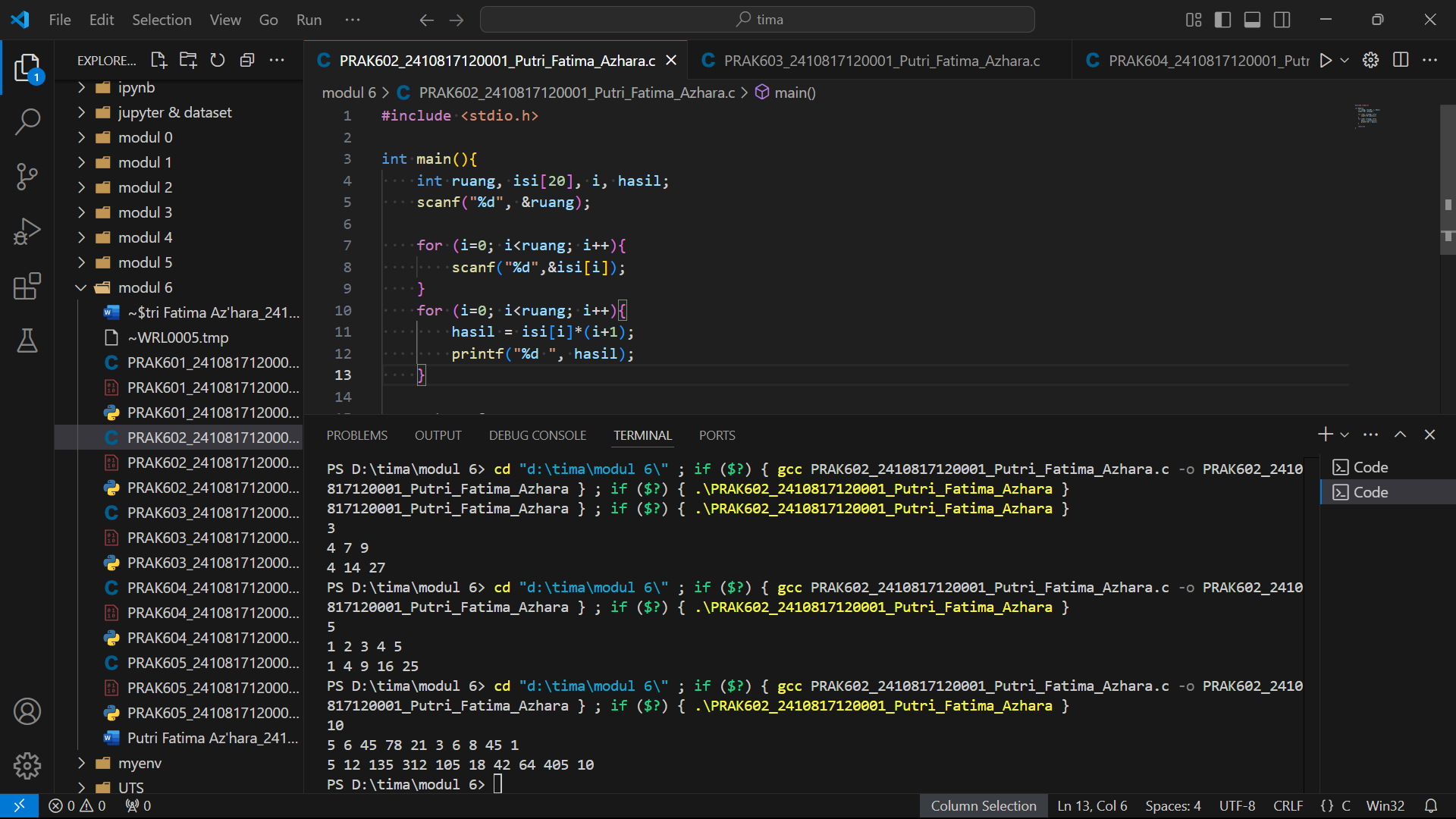
* Python

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | ruang = int(input())  isi = list(map(int, input().split()))  for i in range(ruang):      hasil = isi[i]\*(i+1)      print(hasil, end=" ") |

Tabel 4 Source Code Soal 2 python

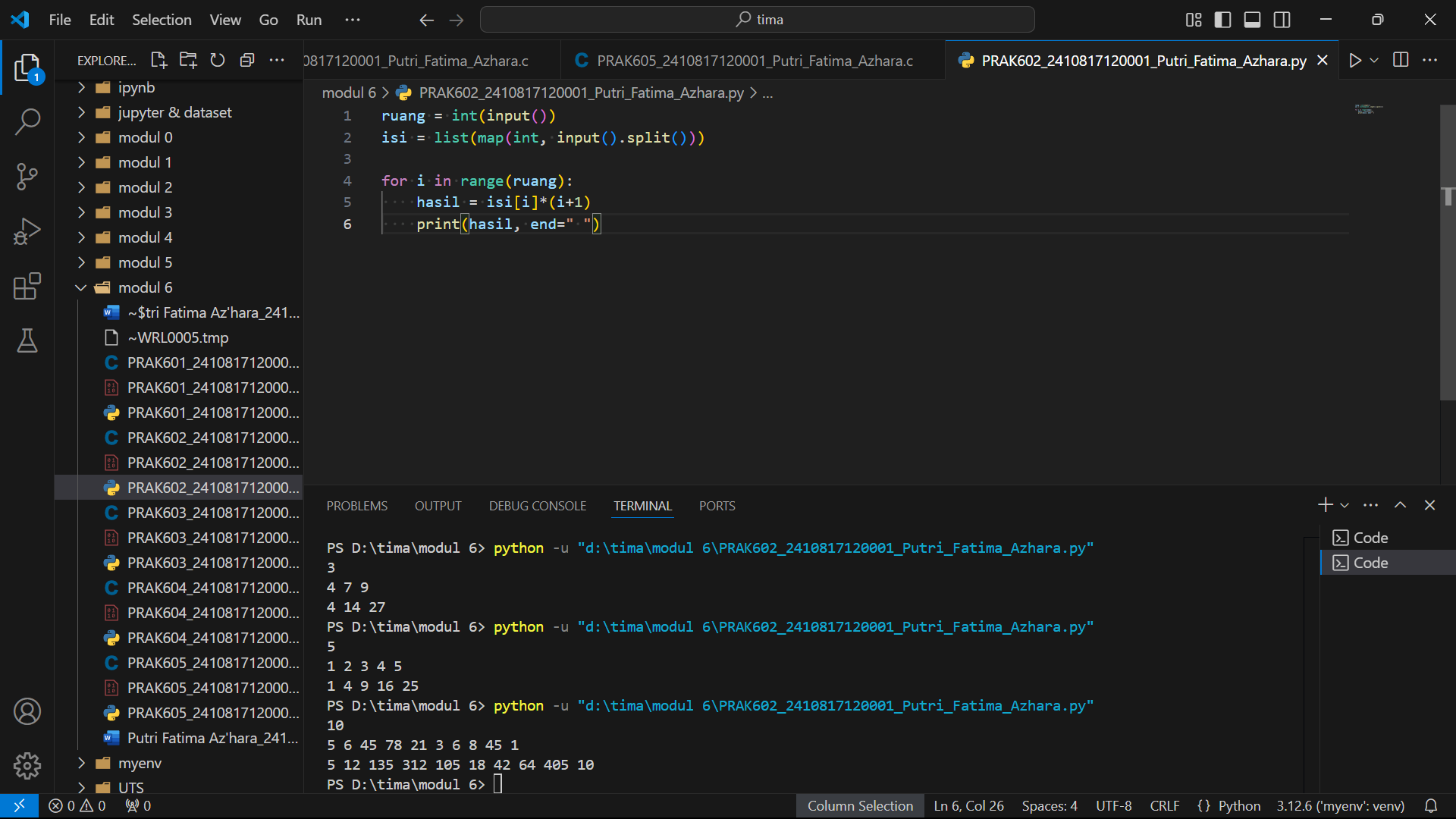
## Output Program

* C



Gambar 3. Screenshot Hasil Jawaban Soal 2 C

* Python



Gambar 4. Screenshot Hasil Jawaban Soal 2 python

## C. Pembahasan

1. Bahasa C:

* # include <stdio.h> : Memasukkan file header stdio.h, yang berisi deklarasi fungsi-

fungsi untuk input dan output.

* int main() : Fungsi utama dalam pemograman C. Int dalam int main()

menunjukkan fungsi ini mengembalikan nilai integer.

* Tanda {} :Berisi kumpulan perintah (*statement*) agar dieksekusi

Berurutan.

* Int ; : Mendeklarasikan variabel ruang, isi[20], i, dan hasil dengan

tipe data integer.

* Scanf () : Fungsi untuk membaca input dari pengguna.
* %d : Format spesifier untuk menampilkan integer pada output.
* &ruang : Input akan dimasukkan dan disimpan di variabel ruang.
* for() : Pernyataan loop yang digunakan untuk mengulang suatu blok

kode dengan jumlah iterasi tertentu.

* (i = 0; i < ruang; i++) :1) Variabel i bertipe integer diinisialisasi dengan nilai 0

sebagai angka memulai loop.

2) Variabel i akan berhenti melakukan loop jika angka sudah sampai kurang dari nilai variabel ruang.

3) Nilai i akan bertambah 1, dikarenakan Operator i++ adalah *shorthand* untuk i = i + 1

* &isi[i] : Membaca elemen bilangan bulat dari pengguna dan

menyimpannya ke dalam array isi pada indeks i.

* Variabel hasil : Menyimpan hasil perhitungan dari elemen array isi pada

indeks ke-i dikali nilai elemen dengan indeksnya yang dimulai dari 1.

* printf (“”) :Sebuah perintah (*statement*) untuk menampilkan output.
* return 0; :Mengembalikan nilai 0, yang menunjukan program

telah berjalan tanpa kesalahan (*error*).

1. Bahasa Python:

* ruang = Input() :Berfungsi untuk mengambil input dari pengguna, kemudian

menyimpan nilai input pengguna di variabel ruang.

* isi = Input() :Berfungsi untuk mengambil input dari pengguna, kemudian

menyimpan nilai input pengguna di variabel isi.

* split() :Berfungsi memisahkan input pengguna berdasarkan spasi.
* map() : mengubah setiap isi input yang dipisahkan spasi menjadi

bilangan bulat.

* list() : Mengubah hasil map () menjadi data list.
* for() : Pernyataan loop yang digunakan untuk mengulang suatu blok

kode dengan jumlah iterasi tertentu.

* for i in range(ruang) : Membuat daftar bilangan dari 0 hingga ruang-1, digunakan

untuk iterasi sebanyak jumlah isi.

* Variabel hasil : Menyimpan hasil perhitungan dari elemen array isi pada

indeks ke-i dikali nilai elemen dengan indeksnya yang dimulai dari 1.

* end=' ' : Mengatur agar output dicetak dalam satu baris dengan spasi

antar elemen.

* print(“”) : Sebuah perintah (*statement*) untuk menampilkan output.

# SOAL 3

1. 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 :

x

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 |

## 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 n1,n2,baris1[50], baris2[50], hasil, i;      scanf("%d %d", &n1,&n2);      if (n1==n2){          for (i=0; i<n1; i++){              scanf("%d", &baris1[i]);          }          for (i=0; i<n2; i++){              scanf("%d", &baris2[i]);          }          for (i=0; i<n1; i++){              hasil = baris1[i]\*baris2[i];              printf("%d ", hasil);          }      }      else {          printf("Jumlah tidak sama");      }  } |

Tabel 5 Source Code Soal 3 C

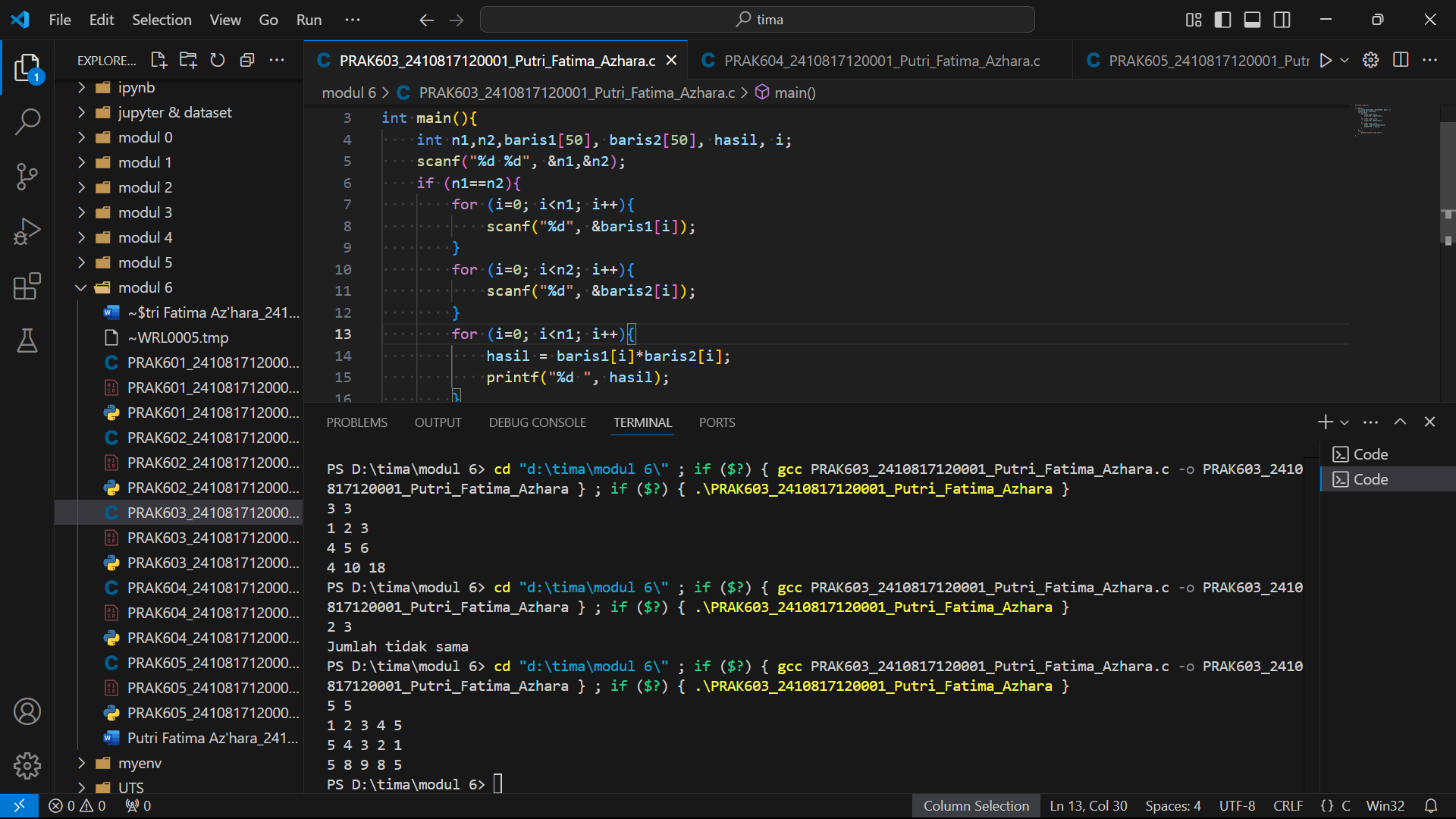
* Python

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | n1, n2 = map(int, input().split())  if (n1==n2):      baris1 = list (map(int, input().split()))      baris2 = list (map(int, input().split()))      for i in range(n1):          hasil = baris1[i] \* baris2[i]          print(hasil, end=" ")  else:      print("Jumlah tidak sama") |

Tabel 6 Source Code Soal 3 python

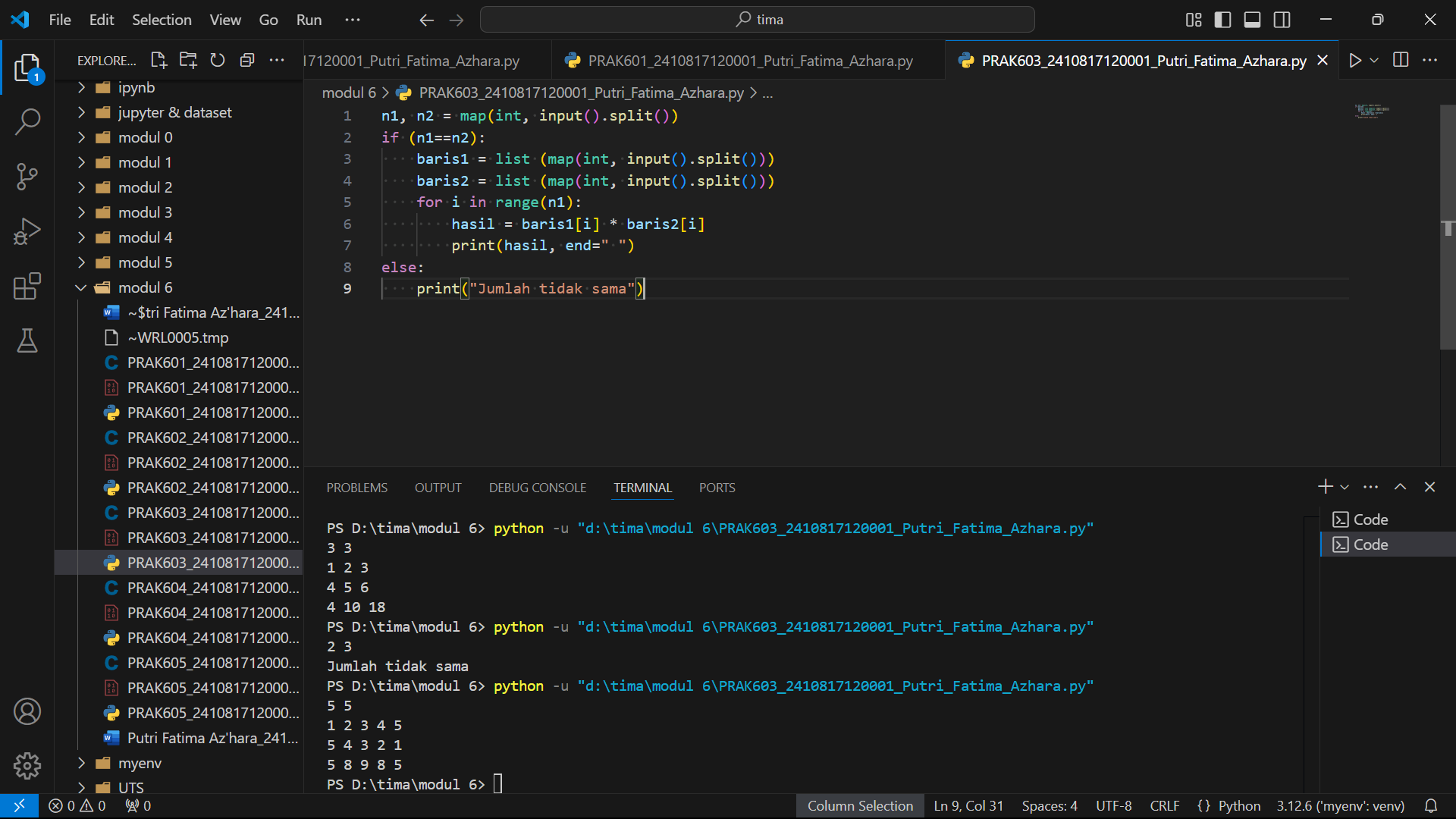
## Output Program

* C



Gambar 5. Screenshot Hasil Jawaban Soal 3 C

* Python



Gambar 6. Screenshot Hasil Jawaban Soal 3 python

## Pembahasan

1. Bahasa C:

* # include <stdio.h> : Memasukkan file header stdio.h, yang berisi

deklarasi fungsi-fungsi untuk input dan output.

* int main() : Fungsi utama dalam program C, di mana eksekusi program

dimulai. Kata kunci int menunjukkan bahwa fungsi ini akan mengembalikan nilai integer.

* Tanda {} : Berisi kumpulan perintah (*statement*) agar dieksekusi

Berurutan.

* Int ; : Mendeklarasikan variabel n1, n2, baris1[50], baris2[50], hasil,

dan i dengan tipe data integer.

* Variabel baris1[50] : Berisi array yang dapat menyimpan hingga 50 elemen.
* Variabel baris2[50] : Berisi array yang dapat menyimpan hingga 50 elemen.
* Scanf () : Fungsi untuk membaca input dari pengguna.
* %d: Format spesifier untuk membaca integer.
* &n1: Input disimpan ke dalam variabel n1.
* &n2 : Input disimpan ke dalam variabel n2.
* if :Pernyataan kondisional yang digunakan untuk mengeksekusi

baris kode hanya jika suatu kondisi tertentu bernilai benar (*true*).

* if(n1==n2) : Kode tersebut berisikan syarat/aturan sebuah output dapat

dieksekusi. Syarat berupa jika nilai variabel n1 sama dengan nilai variabel n2.

* for() : Pernyataan loop yang digunakan untuk mengulang suatu blok

kode dengan jumlah iterasi tertentu.

* (i = 0; i < n1; i++) :1) Variabel i bertipe integer diinisialisasi dengan nilai 0

sebagai angka memulai loop.

2) Variabel i akan berhenti melakukan loop jika angka sudah sampai kurang dari nilai variabel n1.

3) Nilai i akan bertambah 1, dikarenakan Operator i++ adalah *shorthand* untuk i = i + 1.

* (i = 0; i < n2; i++) :1) Variabel i bertipe integer diinisialisasi dengan nilai 0

sebagai angka memulai loop.

2) Variabel i akan berhenti melakukan loop jika angka sudah sampai kurang dari nilai variabel n2.

3) Nilai i akan bertambah 1, dikarenakan Operator i++ adalah *shorthand* untuk i = i + 1.

* &baris1[i] : Menginput elemen array baris1 pada indeks i dari input

pengguna.

* &baris2[i] : Menginput elemen array baris2 pada indeks i dari input

pengguna.

* Variabel hasil : Mengalikan elemen array baris1 dan baris2 pada indeks yang

sama, kemudian menyimpan hasilnya di variabel hasil.

* else : Biasanya diletakkan di akhir struktur percabangan

pengkondisian. Else akan dieksekusi ketika semua kondisi sebelumnya baik if bernilai salah.

* printf (“”) :Sebuah perintah (*statement*) untuk menampilkan output.
* return 0; :Mengembalikan nilai 0, yang menunjukan program

telah berjalan tanpa kesalahan (*error*).

1. Bahasa Python:

* n1, n2 = Input() :Berfungsi untuk mengambil input dari pengguna, kemudian

menyimpan nilai input pengguna di variabel isi.

* split() :Berfungsi memisahkan input pengguna berdasarkan spasi.
* map(int, ...) : mengubah setiap isi input yang dipisahkan spasi menjadi

bilangan bulat.

* if :Pernyataan kondisional yang digunakan untuk mengeksekusi

baris kode hanya jika suatu kondisi tertentu bernilai benar (*true*).

* if(n1==n2) : Kode tersebut berisikan syarat/aturan sebuah output dapat

dieksekusi. Syarat berupa jika nilai variabel n1 sama dengan nilai variabel n2.

* baris1 = Input() :Berfungsi untuk mengambil input dari pengguna, kemudian

menyimpan nilai input pengguna di variabel baris1.

* baris2 = Input() :Berfungsi untuk mengambil input dari pengguna, kemudian

menyimpan nilai input pengguna di variabel baris2.

* split() :Berfungsi memisahkan input pengguna berdasarkan spasi.
* map(int, ...) : mengubah setiap isi input yang dipisahkan spasi menjadi

bilangan bulat.

* list() : Mengubah hasil map () menjadi data list.
* for() : Pernyataan loop yang digunakan untuk mengulang suatu blok

kode dengan jumlah iterasi tertentu.

* for i in range(n1) : Membuat daftar bilangan dari 0 hingga n1-1, digunakan untuk

iterasi sebanyak jumlah elemen dalam baris1 dan baris2.

* Variabel hasil : Mengalikan elemen array baris1 dan baris2 pada indeks yang

sama, kemudian menyimpan hasilnya di variabel hasil.

* else : Biasanya diletakkan di akhir struktur percabangan

pengkondisian. Else akan dieksekusi ketika semua kondisi sebelumnya baik if bernilai salah.

* end=' ' : Mengatur agar output dicetak dalam satu baris dengan spasi

antar elemen.

* print(“”) : Sebuah perintah (*statement*) untuk menampilkan output.

# SOAL 4

1. 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 |

## 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>  int main(){      char asli[100];      char palsu[100];      scanf("%[^\n]%\*c", &asli);      scanf("%[^\n]%\*c", &palsu);      int kode, pesan, x, i=0, j=0;      kode = strlen(asli);      pesan = strlen(palsu);      if(kode != pesan){          printf("Panjang kalimat berbeda, pesan palsu");      }      else{          for(x = 0; x < kode; x++){              if(asli[x]==palsu[x]){                  printf("\*");                  i++;              }              else{                  printf("#");                  j++;              }          }          printf("\n\* = %d", i);          printf("\n# = %d", j);          if(i >= j){              printf("\nPesan Asli");          }          else if(i < j){          printf("\nPesan Palsu");          }      }  } |

Tabel 7 Source Code Soal 4 C

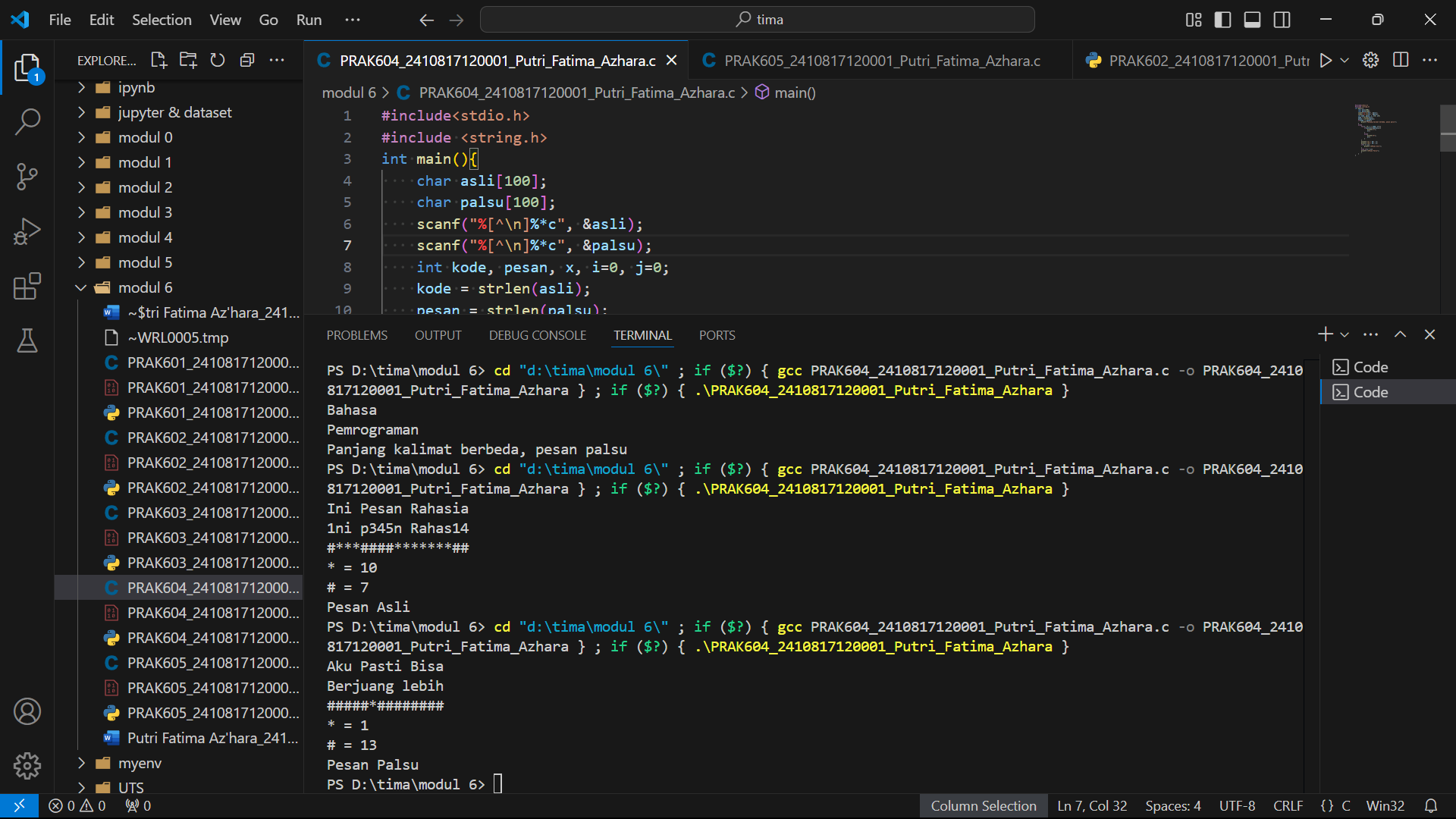
* Python

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | asli = input()  palsu = input()  kode = len(asli)  pesan = len(palsu)  x = 0  y = 0  if len(asli) != len(palsu):      print('Panjang kalimat berbeda, pesan palsu')  else:      for i in range(len(asli)):          if asli[i] == palsu[i]:              print('\*', end='')              x+=1          else:              print('#', end='')              y+=1      print('\n\* = {}' .format(x))      print('# = {}' .format(y))      if x >= y:          print('Pesan Asli')      elif x < y:          print('Pesan Palsu') |

Tabel 8 Source Code Soal 4 python

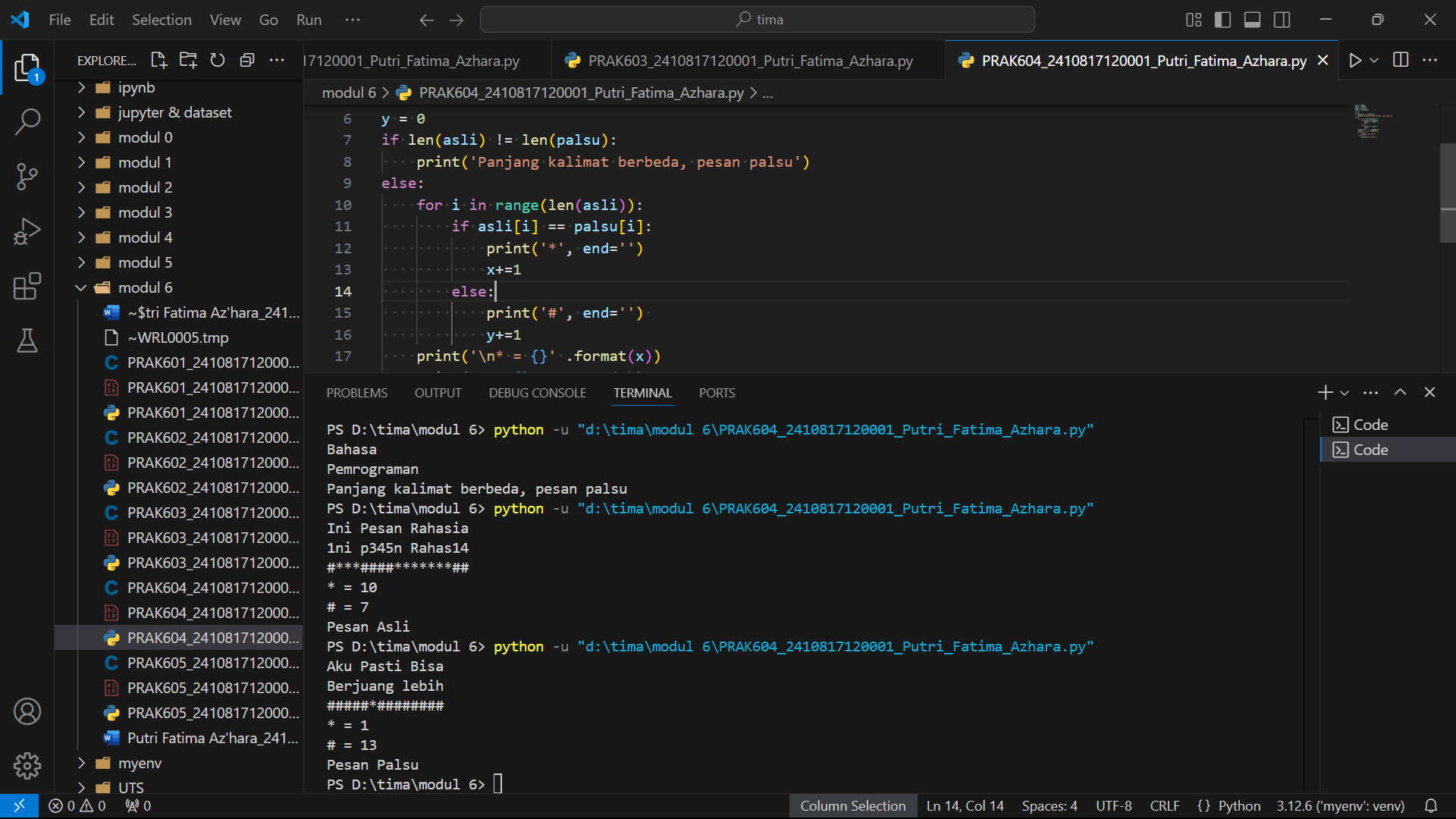
## Output Program

* C



Gambar 7. Screenshot Hasil Jawaban Soal 4 C

* Python



Gambar 8. Screenshot Hasil Jawaban Soal 4 python

## C. Pembahasan

1. Bahasa C:

* # include <stdio.h> : Memasukkan file header stdio.h, yang berisi

deklarasi fungsi-fungsi untuk input dan output.

* #include<string.h> : Berfungsi untuk memasukkan file header string.h, yang

menyediakan fungsi manipulasi string.

* int main() : Fungsi utama dalam program C, di mana eksekusi program

dimulai. Kata kunci int menunjukkan bahwa fungsi ini akan mengembalikan nilai integer.

* Tanda {} : Berisi kumpulan perintah (*statement*) agar dieksekusi

Berurutan.

* char : Tipe data untuk menyimpan karakter atau array karakter

(string).

* asli[100] : Array untuk menyimpan string variabel asli yang dimasukkan

pengguna, dengan maksimum 100 karakter.

* palsu[100] : Array untuk menyimpan string variabel palsu yang dimasukkan

pengguna, dengan maksimum 100 karakter.

* scanf () : Fungsi untuk membaca input dari pengguna.
* %[^\n]%\*c : Format spesifier untuk membaca seluruh kalimat hingga

menemukan karakter *newline* (\n).

* &asli : Input akan dimasukkan dan disimpan di variabel asli
* &palsu : Input akan dimasukkan dan disimpan di variabel palsu
* Int ; : Mendeklarasikan variabel kode, pesan, x, i, dan j dengan tipe

data integer.

* Variabel i dan j : Berisikan nilai 0.
* kode = strlen(asli) : Menghitung panjang string asli dan menyimpannya di

variabel kode.

* pesan = strlen(palsu) : Menghitung panjang string palsu dan menyimpannya di

variabel pesan.

* if :Pernyataan kondisional yang digunakan untuk mengeksekusi

baris kode hanya jika suatu kondisi tertentu bernilai benar (*true*).

* else if :Pernyataan kondisional lainnya setelah if yang dibutuhkan jika

memiliki lebih dari satu kondisi dan akan dieksekusi jika kondisi sebelumnya salah.

* else : Biasanya diletakkan di akhir struktur percabangan

pengkondisian. Else akan dieksekusi ketika semua kondisi sebelumnya baik if maupun else if bernilai salah.

* if(kode != pesan) : Kode tersebut berisikan syarat/aturan sebuah output dapat

dieksekusi. Syarat berupa jika variabel kode tidak sama dengan variabel pesan.

* for() : Pernyataan loop yang digunakan untuk mengulang suatu blok

kode dengan jumlah iterasi tertentu.

* (x = 0; x < kode; x++) :1) Variabel x bertipe integer diinisialisasi dengan nilai 0

sebagai angka memulai loop.

2) Variabel x akan berhenti melakukan loop jika angka sudah sampai kurang dari nilai variabel kode.

3) Nilai x akan bertambah 1, dikarenakan Operator x++ adalah *shorthand* untuk x = x + 1.

* if(asli[x] == palsu[x]) : Memeriksa apakah karakter pada indeks x di asli sama dengan

karakter pada indeks yang sama di palsu. Jika sama, program akan mencetak \*.

* i++ : Nilai i akan bertambah 1 setiap iterasi (shorthand untuk i = i

+ 1).

* j++ : Nilai j akan bertambah 1 setiap iterasi (shorthand untuk j = j

+ 1).

* \n : Membuat baris baru.
* %d : Format spesifier untuk menampilkan integer pada output.
* if(i >= j) : Kode tersebut berisikan syarat/aturan sebuah output dapat

dieksekusi. Syarat berupa jika variabel i lebih besar sama dengan variabel j.

* else if(i < j) : Kode tersebut berisikan syarat/aturan sebuah output dapat

dieksekusi. Syarat berupa jika variabel i kurang dari j.

* printf (“”) :Sebuah perintah (*statement*) untuk menampilkan output.
* return 0; :Mengembalikan nilai 0, yang menunjukan program

telah berjalan tanpa kesalahan (*error*).

1. Bahasa Python:

* asli = input() : Mengambil input dari pengguna dan menyimpannya dalam

variabel asli.

* palsu = input() : Mengambil input dari pengguna dan menyimpannya dalam

variabel palsu.

* kode = len(asli) : Menghitung panjang string asli dan menyimpannya dalam

variabel kode.

* pesan = len(palsu) : Menghitung panjang string asli dan menyimpannya dalam

variabel pesan.

* len() : Berfungsi menghitung panjang karakter dalam string.
* Variabel x dan y : Berisikan nilai 0.
* if :Pernyataan kondisional yang digunakan untuk mengeksekusi

baris kode hanya jika suatu kondisi tertentu bernilai benar (*true*).

* elif :Pernyataan kondisional lainnya setelah if yang dibutuhkan jika

memiliki lebih dari satu kondisi dan akan dieksekusi jika kondisi sebelumnya salah.

* else : Biasanya diletakkan di akhir struktur percabangan

pengkondisian. Else akan dieksekusi ketika semua kondisi sebelumnya baik if maupun else if bernilai salah.

* if len(palsu) != len(asli) : Kode tersebut berisikan syarat/aturan sebuah output dapat

dieksekusi. Syarat berupa jika panjang string variabel palsu tidak sama dengan panjang string variabel asli.

* for() : Pernyataan loop yang digunakan untuk mengulang suatu blok

kode dengan jumlah iterasi tertentu.

* for i in range(len(asli)) : Loop yang digunakan untuk menghasilkan urutan indeks yang

sesuai dengan panjang string asli.

* if asli[i] == palsu[i] : Memeriksa apakah karakter pada indeks i di string asli sama

dengan karakter pada indeks yang sama di string palsu.

* end=' ' : Mengatur agar output dicetak dalam satu baris dengan spasi

antar elemen.

* x += 1 : Jika sama, mencetak \* dan menambah nilai x sebanyak 1.
* y += 1 : Jika tidak sama, mencetak # dan menambah nilai y sebanyak 1.
* \n : Membuat baris baru.
* .format() : digunakan untuk memasukkan nilai ke dalam string pada

tempat yang ditentukan.

* if x >= y : Jika jumlah karakter yang sama (x) lebih banyak atau sama

dengan karakter yang berbeda (y), maka program mencetak “pesan asli“.

* elif x < y : Jika jumlah karakter yang sama (x) lebih banyak atau sama

dengan karakter yang berbeda (y), maka program mencetak “pesan palsu“.

* print(“”) :Sebuah perintah (*statement*) untuk menampilkan output.

# SOAL 5

1. 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 |

## 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 | #include <stdio.h>  int main(){      int Matriks\_A[10][10], Matriks\_B[10][10], Matriks\_AXB[10][10];      int i, j, k, n, jumlah = 0;      scanf("%d", &n);      printf("Matriks A\n");      for(i = 0; i < n; i++){          for(j = 0; j < n; j++){              scanf("%d", &Matriks\_A[i][j]);          }      }      printf("Matriks B\n");      for(i = 0; i < n; i++){          for(j = 0; j < n; j++){              scanf("%d", &Matriks\_B[i][j]);          }      }      for(i = 0; i < n; i++){          for(j = 0; j < n; j++){              for(k = 0; k < n; k++){                  jumlah = jumlah + Matriks\_A[i][k] \* Matriks\_B[k][j];              }              Matriks\_AXB[i][j] = jumlah;              jumlah = 0;          }      }      printf("Matriks AXB\n");      for(i = 0; i < n; i++){          for(j = 0; j < n; j++){              printf("%d ", Matriks\_AXB[i][j]);          }          printf("\n");      }  } |

Tabel 9 Source Code C Soal 5 C

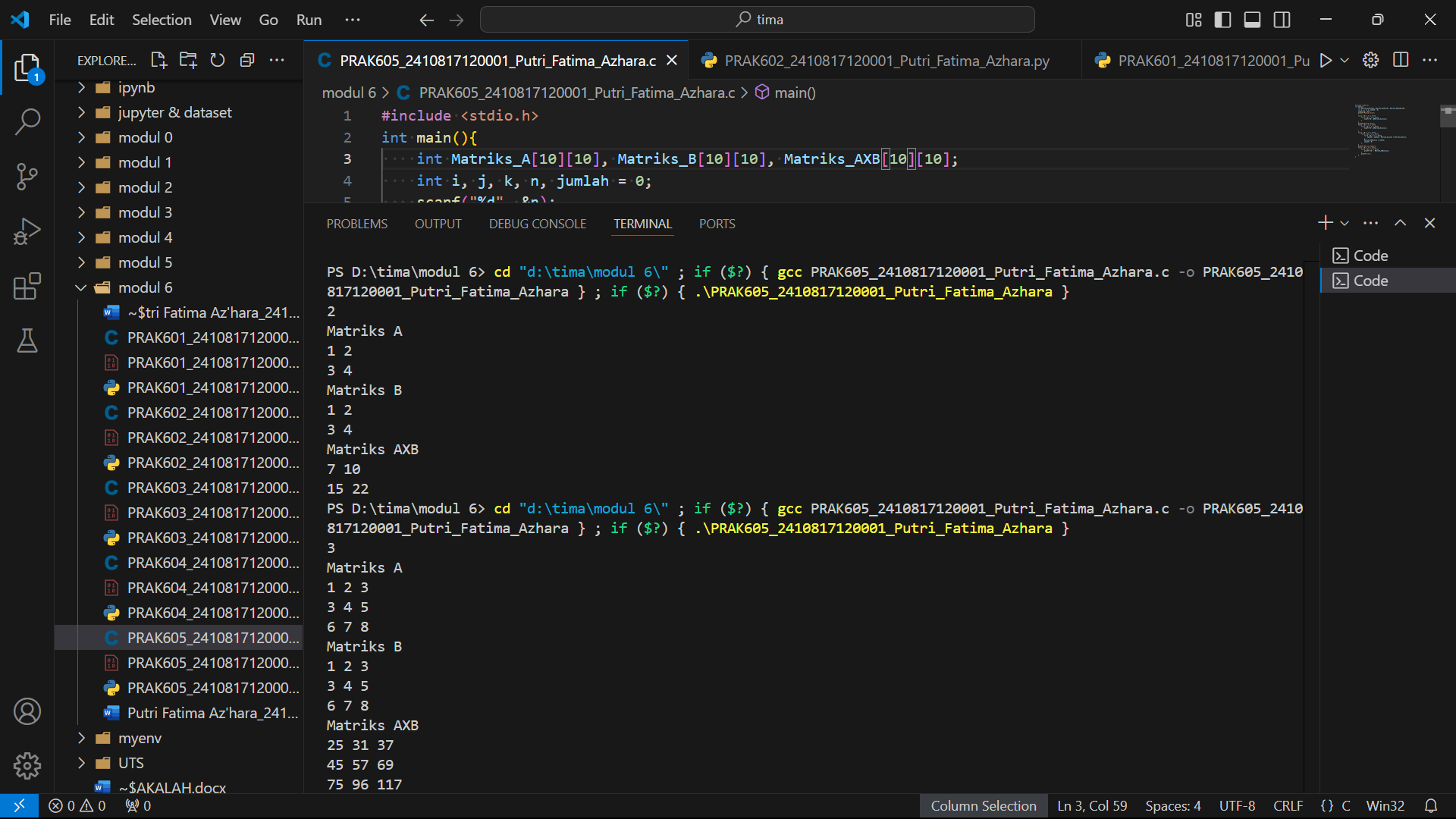
* Python

|  |  |
| --- | --- |
| 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 | x = int(input(""))  MatriksA = []  MatriksB = []  MatriksAB = []  print("Matriks A")  for i in range(x):      k = list(map(int, input("").split()))      MatriksA.append(k)  print("Matriks B")  for i in range(x):      k = list(map(int, input("").split()))      MatriksB.append(k)  print("Matriks AXB")  for i in range(x):      MatriksAB.append([])      for j in range(x):          Hasil = 0          for k in range(x):              Hasil += MatriksA[i][k] \* MatriksB[k][j]          MatriksAB[i].append(Hasil)  for number in MatriksAB:      for k in number:          print(k, end=" ")      print() |

Tabel 10 Source Code Soal 5 python

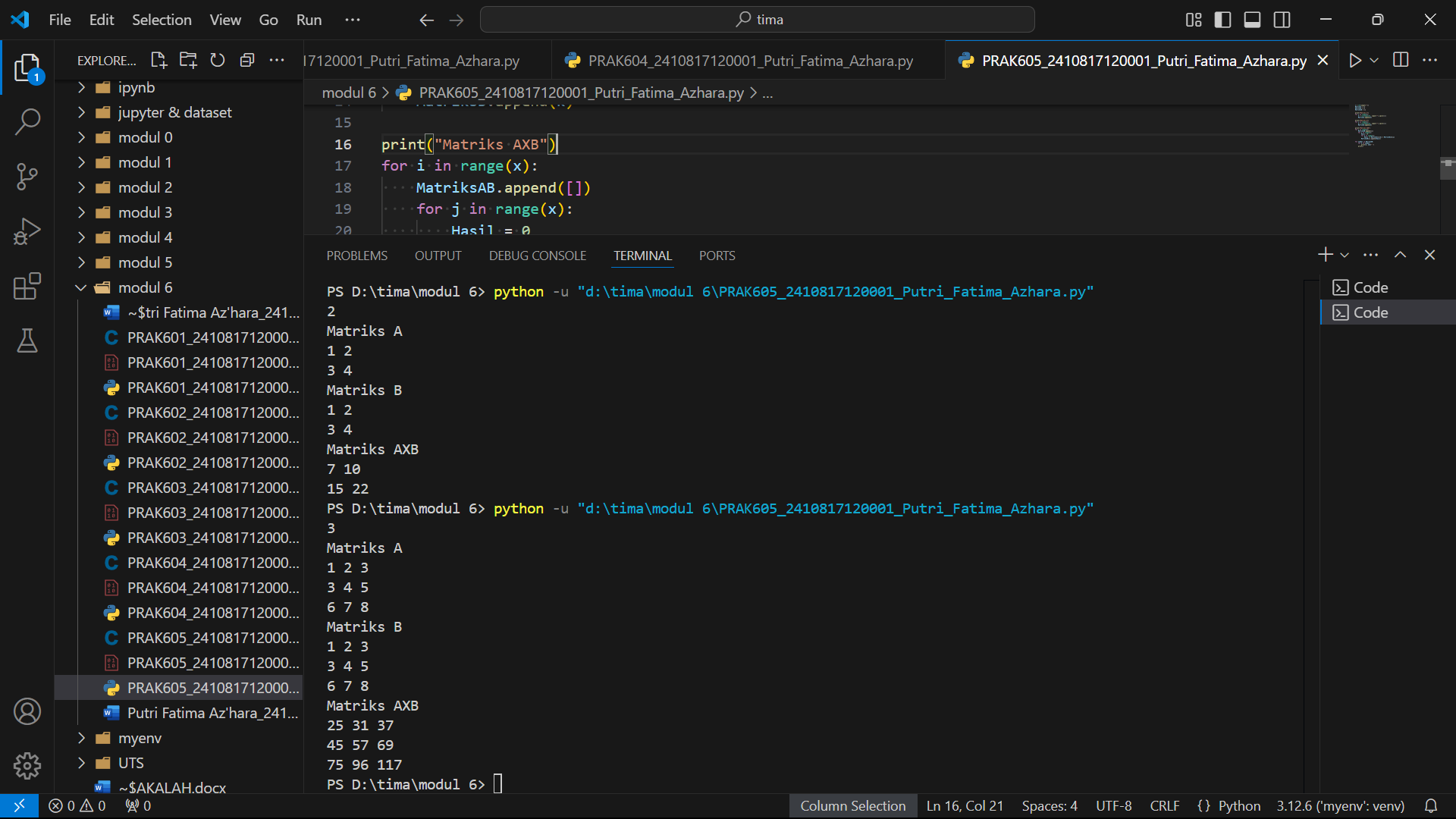
## Output Program

* C



Gambar 9. Screenshot Hasil Jawaban Soal 5 C

* Python



Gambar 10. Screenshot Hasil Jawaban Soal 5 python

## Pembahasan

1. Bahasa C:

* # include <stdio.h> : Memasukkan file header stdio.h, yang berisi

deklarasi fungsi-fungsi untuk input dan output.

* int main() : Fungsi utama dalam program C, di mana eksekusi program

dimulai. Kata kunci int menunjukkan bahwa fungsi ini akan mengembalikan nilai integer.

* Tanda {} : Berisi kumpulan perintah (*statement*) agar dieksekusi

Berurutan.

* Int ; : Mendeklarasikan variabel Matriks\_A[10][10],

Matriks\_B[10][10], Matriks\_AXB[10][10], i, j, k, n, dan jumlah dengan tipe data integer.

* Matriks\_A[10][10] : Deklarasi array 2 dimensi untuk menyimpan elemen matriks A.
* Matriks\_B[10][10] : Deklarasi array 2 dimensi untuk menyimpan elemen matriks B.
* Matriks\_AXB[10][10] : Deklarasi array 2 dimensi untuk menyimpan elemen matriks

AXB.

* Scanf () : Fungsi untuk membaca input dari pengguna.
* %d: Format spesifier untuk membaca integer.
* &n: Input disimpan ke dalam variabel n.
* \n : Membuat garis baru.
* for() : Pernyataan loop yang digunakan untuk mengulang suatu blok

kode dengan jumlah iterasi tertentu.

* (i = 0; i < n; i++) :1) Variabel i bertipe integer diinisialisasi dengan nilai 0

sebagai angka memulai loop.

2) Variabel i akan berhenti melakukan loop jika angka sudah sampai kurang dari nilai variabel kode.

3) Nilai i akan bertambah 1, dikarenakan Operator i++ adalah *shorthand* untuk i = i + 1.

* (j = 0; j < n; j++) :1) Variabel j bertipe integer diinisialisasi dengan nilai 0

sebagai angka memulai loop.

2) Variabel j akan berhenti melakukan loop jika angka sudah sampai kurang dari nilai variabel kode.

3) Nilai j akan bertambah 1, dikarenakan Operator j++ adalah *shorthand* untuk j = j + 1.

* (k = 0; k < n; k++) :1) Variabel k bertipe integer diinisialisasi dengan nilai 0

sebagai angka memulai loop.

2) Variabel k akan berhenti melakukan loop jika angka sudah sampai kurang dari nilai variabel kode.

3) Nilai k akan bertambah 1, dikarenakan Operator k++ adalah *shorthand* untuk k = k + 1.

* &Matriks\_A[10][10]: Input disimpan ke dalam variabel Matriks\_A.
* &Matriks\_B[10][10]: Input disimpan ke dalam variabel Matriks\_B.
* Variabel jumlah : Berisi penghitungan elemen matriks AXB dengan rumus

perkalian matriks.

* jumlah = 0 : Mengatur ulang nilai jumlah untuk iterasi berikutnya.
* printf (“”) :Sebuah perintah (*statement*) untuk menampilkan output.
* return 0; :Mengembalikan nilai 0, yang menunjukan program

telah berjalan tanpa kesalahan (*error*).

1. Bahasa Python:

* def Biodata() : Membuat fungsi (*function*) dengan nama Biodata.
* x=input("") : Meminta pengguna untuk memasukkan ukuran matriks

(bilangan bulat).

* int() : Mengonversi input string menjadi bilangan bulat. Variabel x

menyimpan ukuran matriks (jumlah baris dan kolom).

* MatriksA = [] : Membut list kosong untuk menyimpan elemen Matriks A.
* MatriksB = [] : Membut list kosong untuk menyimpan elemen Matriks B.
* MatriksAB = [] : Membut list kosong untuk menyimpan hasil perkalian Matriks

A dan Matriks B.

* for i in range(x) : Loop untuk mengulangi sebanyak x kali (jumlah baris Matriks

A).

* input("").split() : Membaca satu baris input sebagai string, memisahkannya

menjadi elemen-elemen berdasarkan spasi.

* map(int, ...) : Mengonversi setiap elemen yang dipisahkan menjadi bilangan

bulat.

* list() : Mengubah hasil map menjadi list.
* MatriksA.append(k) : Menambahkan list hasil konversi (k) ke dalam Matriks A.
* MatriksB.append(k) : Menambahkan list hasil konversi (k) ke dalam Matriks B.
* MatriksAB.append(k) : Menambahkan Menambahkan list kosong untuk setiap baris

hasil perkalian.

* for i in range(x) : Iterasi untuk setiap baris Matriks A.
* for j in range(x) : Iterasi untuk setiap kolom Matriks B.
* Hasil = 0 : Inisialisasi variabel Hasil untuk menyimpan total hasil

perkalian elemen sejajar.

* for k in range(x) : Iterasi untuk elemen sejajar pada baris Matriks A dan kolom

Matriks B.

* Variabel hasil : Menghitung hasil perkalian elemen Matriks A dan Matriks B

pada indeks tertentu, lalu menambahkannya ke Hasil.

* MatriksAB[i].append(Hasil): Menambahkan hasil perkalian (Hasil) ke list baris hasil

perkalian Matriks AB.

* end=' ' : Mengatur agar output dicetak dalam satu baris dengan spasi

antar elemen.

* print (“”) :Sebuah perintah (*statement*) untuk menampilkan output.