**LAPORAN PRAKTIKUM**

**PEMROGRAMAN I**

**MODUL 6**

****

**Array**

**Oleh:**

**Rizky Putra Afhani NIM. 2410817210013**

**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 : Rizky Putra Afhani

NIM : 2410817210013

|  |  |
| --- | --- |
| Menyetujui,  Asisten Praktikum  Randy Febrian  NIM. 2310817110013 | Mengetahui,  Dosen Penanggung Jawab Praktikum  Ir. Eka Setya Wijaya S.T., M.Kom.  NIP. 198205082008011010 |

# DAFTAR ISI

[LEMBAR PENGESAHAN 2](#_Toc185681020)

[DAFTAR ISI 3](#_Toc185681021)

[DAFTAR GAMBAR 4](#_Toc185681022)

[DAFTAR TABEL 4](#_Toc185681023)

[SOAL 1 6](#_Toc185681024)

[A. Source Code 7](#_Toc185681025)

[B. Output Program 8](#_Toc185681026)

[C. Pembahasan 9](#_Toc185681027)

[SOAL 2 11](#_Toc185681028)

[A. Source Code 11](#_Toc185681029)

[B. Output Program 13](#_Toc185681030)

[C. Pembahasan 13](#_Toc185681031)

[SOAL 3 14](#_Toc185681032)

[A. Source Code 14](#_Toc185681033)

[B. Output Program 16](#_Toc185681034)

[C. Pembahasan 16](#_Toc185681035)

[SOAL 4 17](#_Toc185681036)

[A. Source Code 18](#_Toc185681037)

[B. Output Program 19](#_Toc185681038)

[C. Pembahasan 20](#_Toc185681039)

[SOAL 5 21](#_Toc185681040)

[A. Source Code 21](#_Toc185681041)

[B. Output Program 23](#_Toc185681042)

[C. Pembahasan 24](#_Toc185681043)

[TAUTAN GIT 25](#_Toc185681044)

# DAFTAR GAMBAR

[Gambar 1. Screenshot Output PRAK601 Bahasa C 8](#_Toc185681114)

[Gambar 2. Screenshot Output PRAK601 Bahasa Python 9](#_Toc185681115)

[Gambar 3. Screenshot Output PRAK602 Bahasa C 13](#_Toc185681116)

[Gambar 4. Screenshot Output PRAK602 Bahasa Python 13](#_Toc185681117)

[Gambar 5. Screenshot Output PRAK603 Bahasa C 16](#_Toc185681118)

[Gambar 6. Screenshot Output PRAK603 Bahasa Python 16](#_Toc185681119)

[Gambar 7. Screenshot Output PRAK604 Bahasa C 19](#_Toc185681120)

[Gambar 8. Screenshot Output PRAK604 Bahasa Python 20](#_Toc185681121)

[Gambar 9. Screenshot Output PRAK605 Bahasa C 23](#_Toc185681122)

[Gambar 10. Screenshot Output PRAK605 Bahasa Python 24](#_Toc185681123)

# DAFTAR TABEL

[Tabel 1 Source Code PRAK601 Bahasa C 7](#_Toc185681148)

[Tabel 2 Source Code PRAK601 Bahasa Python 8](#_Toc185681149)

[Tabel 3 Source Code PRAK602 Bahasa C 12](#_Toc185681150)

[Tabel 4 Source Code PRAK602 Bahasa Python 12](#_Toc185681151)

[Tabel 5 Source Code PRAK603 Bahasa C 15](#_Toc185681152)

[Tabel 6 Source Code PRAK603 Bahasa Python 15](#_Toc185681153)

[Tabel 7 Source Code PRAK604 Bahasa C 18](#_Toc185681154)

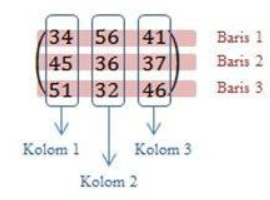
[Tabel 8 Source Code PRAK604 Bahasa Python 19](#_Toc185681155)

[Tabel 9 Source Code PRAK605 Bahasa C 22](#_Toc185681156)

[Tabel 10 Source Code PRAK605 Bahasa Python 23](#_Toc185681157)

# 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

|  |  |
| --- | --- |
| 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>  #include <stdlib.h>  void buat\_matriks() {  int baris, kolom;  printf("");  scanf("%d %d", &baris, &kolom);  int total\_elemen = baris \* kolom;  int \*elemen = (int \*)malloc(total\_elemen \* sizeof(int));  if (elemen == NULL) {  printf("Memori tidak cukup!\n");  return;  }  printf("");  for (int i = 0; i < total\_elemen; i++) {  scanf("%d", &elemen[i]);  }  printf("n");  for (int i = 0; i < baris; i++) {  for (int j = 0; j < kolom; j++) {  printf("%d ", elemen[i \* kolom + j]);  }  printf("\n");  }  free(elemen);  }  int main() {  buat\_matriks();  return 0;  } |

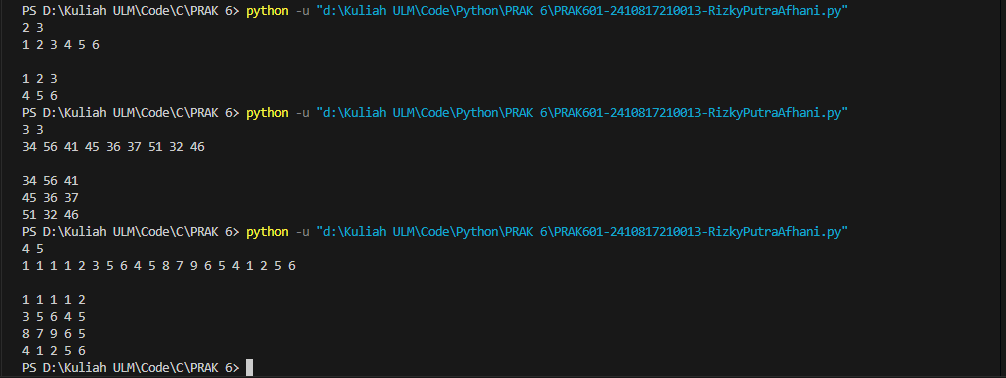
Tabel 1 Source Code PRAK601 Bahasa C

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | def buat\_matriks():    baris, kolom = map(int, input("").split())    elemen = list(map(int, input("").split()))  if len(elemen) != baris \* kolom:  print("")  return  print("")  matriks = []  for i in range(baris):  row = elemen[i \* kolom:(i + 1) \* kolom]  matriks.append(row)  print(" ".join(map(str, row)))  return matriks  buat\_matriks() |

Tabel 2 Source Code PRAK601 Bahasa Python

## Output Program

Gambar 1. Screenshot Output PRAK601 Bahasa C



Gambar 2. Screenshot Output PRAK601 Bahasa Python

## Pembahasan

* **Bahasa C:**

Program di atas dibuat dalam bahasa C untuk membuat dan menampilkan matriks berdasarkan input dari pengguna. Pertama, program meminta pengguna untuk memasukkan jumlah baris dan kolom matriks. Setelah itu, program menghitung jumlah total elemen matriks dan mengalokasikan memori secara dinamis menggunakan fungsi malloc. Dengan memori ini, elemen-elemen matriks yang diinputkan dapat disimpan. Jika terjadi kegagalan dalam alokasi memori, program akan menampilkan pesan kesalahan dan menghentikan prosesnya. Setelah semua elemen dimasukkan, program menampilkan matriks dalam format baris dan kolom sesuai dengan jumlah yang telah ditentukan sebelumnya. Akses ke elemen-elemen matriks dilakukan berdasarkan indeks, yang dihitung berdasarkan posisi baris dan kolomnya. Ketika proses pencetakan selesai, program membebaskan kembali memori yang telah dialokasikan menggunakan fungsi free agar tidak terjadi kebocoran memori. Fungsi utama (main) bertanggung jawab memanggil fungsi buat\_matriks, yang menjalankan seluruh proses pembuatan dan pencetakan matriks sebelum program selesai. Program ini menggunakan alokasi memori dinamis untuk mengelola data dengan ukuran yang bervariasi, sebuah pendekatan yang efisien dan umum dalam pemrograman C.

* **Python:**

Kode di atas berfungsi untuk membuat dan menampilkan matriks berdasarkan input pengguna. Pertama, pengguna diminta untuk memasukkan jumlah baris dan kolom matriks, yang kemudian dipisahkan menggunakan fungsi map(int, input().split()). Setelah itu, pengguna diminta untuk memasukkan elemen- elemen matriks dalam bentuk satu daftar. Jika jumlah elemen yang dimasukkan tidak sesuai dengan ukuran matriks yang telah ditentukan (yaitu, baris dikali kolom), maka fungsi akan berhenti dan tidak melanjutkan proses. Jika jumlah elemen valid, kode akan membagi daftar elemen tersebut menjadi baris-baris

matriks sesuai ukuran yang diinginkan, dan menampilkan setiap baris dengan format yang rapi menggunakan join. Matriks yang terbentuk kemudian dikembalikan sebagai hasil fungsi.

# 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

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | #include <stdio.h>  void hitung\_zetsu\_putih() {  int jumlah\_ruangan;  printf("");  scanf("%d", &jumlah\_ruangan);  int zetsu\_awal[jumlah\_ruangan];  int zetsu\_setelah\_belah[jumlah\_ruangan];  printf("");  for (int i = 0; i < jumlah\_ruangan; i++) {  scanf("%d", &zetsu\_awal[i]);  } |

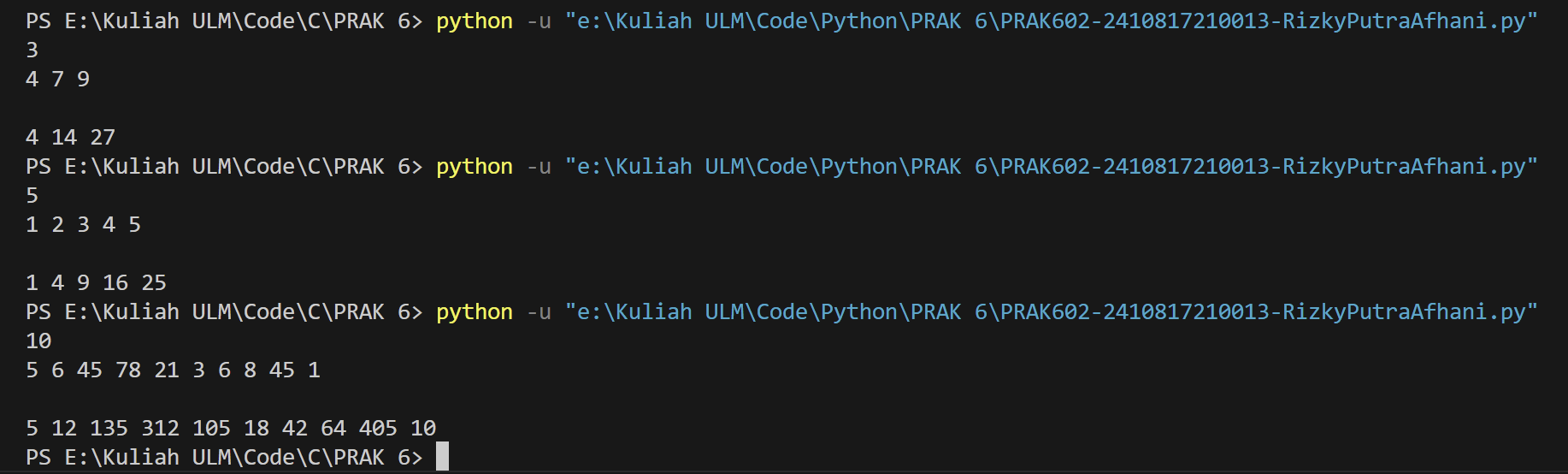
|  |  |
| --- | --- |
| 16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | for (int i = 0; i < jumlah\_ruangan; i++) {  zetsu\_setelah\_belah[i] = zetsu\_awal[i] \* (i + 1);  }  printf("");  for (int i = 0; i < jumlah\_ruangan; i++) {  printf("%d ", zetsu\_setelah\_belah[i]);  }  printf("\n");  }  int main() {  hitung\_zetsu\_putih();  return 0;  } |

Tabel 3 Source Code PRAK602 Bahasa C

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | def hitung\_zetsu\_putih():        jumlah\_ruangan = int(input(""))        zetsu\_awal = list(map(int, input("").split()))          if len(zetsu\_awal) != jumlah\_ruangan:          print("Jumlah input tidak sesuai dengan jumlah ruangan!")          return      zetsu\_setelah\_belah = [zetsu\_awal[i] \* (i + 1) for i in range(jumlah\_ruangan)]      print("")      print(" ".join(map(str, zetsu\_setelah\_belah)))  hitung\_zetsu\_putih() |

Tabel 4 Source Code PRAK602 Bahasa Python

## Output Program

Gambar 3. Screenshot Output PRAK602 Bahasa C

Gambar 4. Screenshot Output PRAK602 Bahasa Python

## Pembahasan

* **Bahasa C:**

Program ini menghitung perubahan nilai dalam sebuah array berdasarkan input pengguna. Fungsi hitung\_zetsu\_putih meminta jumlah ruang dan nilai awal untuk setiap ruang, lalu mengalikan nilai tersebut dengan urutan elemen (indeks + 1) dan menyimpan hasilnya di array baru. Setelah itu, hasil perhitungan ditampilkan ke layar. Program ini cukup sederhana, berfokus pada pengolahan array dengan operasi perkalian. Fungsi utama hanya memanggil fungsi hitung\_zetsu\_putih untuk menjalankan proses tersebut.

* **Python:**

Program ini meminta pengguna memasukkan jumlah ruangan dan nilai-nilai awal dalam bentuk daftar. Setelah memverifikasi bahwa jumlah input sesuai, program mengalikan setiap nilai dengan indeksnya (ditambah satu) untuk menghasilkan array baru. Hasil perhitungan kemudian ditampilkan ke layar. Intinya, program menerima input, memeriksa kesesuaian data, melakukan perhitungan, dan menampilkan hasilnya.

# 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 :

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

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | #include <stdio.h>  void hitung\_perkalian\_baris() {      int n1, n2;      printf("");      scanf("%d %d", &n1, &n2);  if (n1 != n2) { |

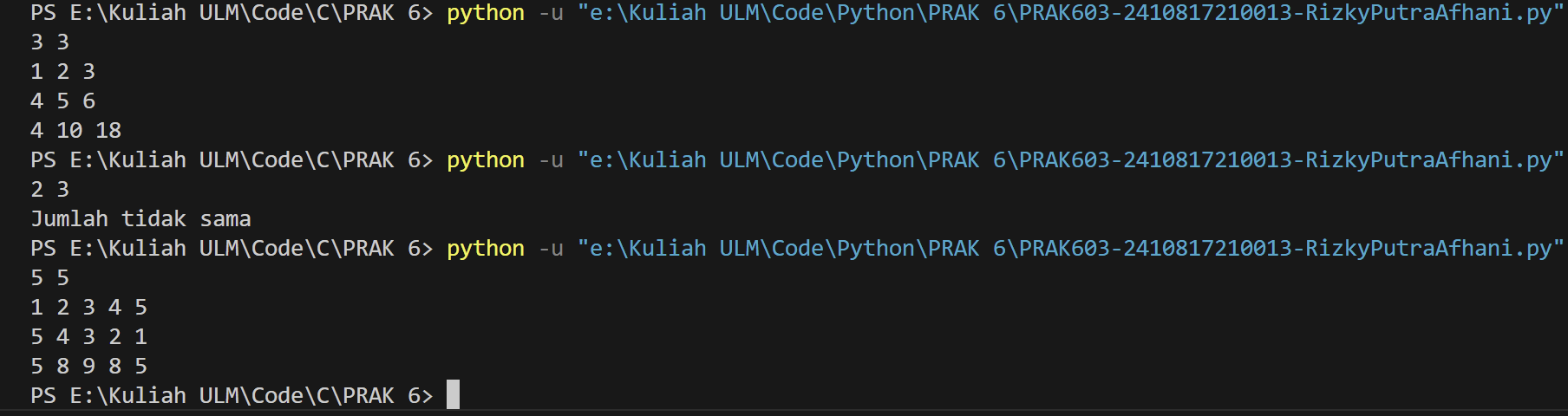
|  |  |
| --- | --- |
| 9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | printf("Jumlah tidak sama\n");          return;      }      int baris1[n1], baris2[n2];      printf("");      for (int i = 0; i < n1; i++) {          scanf("%d", &baris1[i]);      }      printf("");      for (int i = 0; i < n2; i++) {          scanf("%d", &baris2[i]);      }      printf("");      for (int i = 0; i < n1; i++) {          printf("%d ", baris1[i] \* baris2[i]);      }      printf("\n");  }  int main() {      hitung\_perkalian\_baris();      return 0;  } |

Tabel 5 Source Code PRAK603 Bahasa C

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | def hitung\_perkalian\_baris():        n1, n2 = map(int, input().split())        if n1 != n2:          print("Jumlah tidak sama")          return      baris1 = list(map(int, input().split()))        baris2 = list(map(int, input().split()))      hasil = [baris1[i] \* baris2[i] for i in range(n1)]      print(" ".join(map(str, hasil)))  hitung\_perkalian\_baris() |

Tabel 6 Source Code PRAK603 Bahasa Python

## Output Program

Gambar 5. Screenshot Output PRAK603 Bahasa C

Gambar 6. Screenshot Output PRAK603 Bahasa Python

## Pembahasan

* **Bahasa C:**

Program ini menghitung hasil perkalian elemen-elemen dua baris array dengan panjang yang sama. Fungsi hitung\_perkalian\_baris dimulai dengan meminta pengguna untuk memasukkan dua angka yang menunjukkan panjang kedua baris. Jika panjang kedua baris tidak sama, program akan menampilkan pesan kesalahan dan berhenti. Jika panjangnya sama, program melanjutkan dengan meminta pengguna untuk memasukkan nilai untuk kedua baris array. Setelah itu, program mengalikan elemen-elemen pada kedua baris yang berada pada posisi yang sama dan menampilkan hasilnya.

* **Python:**

Program ini menghitung hasil perkalian elemen-elemen dua baris array yang panjangnya harus sama. Fungsi hitung\_perkalian\_baris dimulai dengan meminta dua angka yang menunjukkan panjang kedua baris. Jika panjang kedua baris tidak sama, program akan menampilkan pesan kesalahan dan berhenti. Jika panjangnya sama, program kemudian meminta input dua baris array, lalu mengalikan elemen-elemen yang berada pada posisi yang sama di kedua baris menggunakan list comprehension. Hasil perkalian setiap elemen akan ditampilkan dalam bentuk daftar yang dipisahkan oleh spasi.

# 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

|  |  |
| --- | --- |
| 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 | #include <stdio.h>  #include <string.h>  void shikamaru() {      char kode\_shikamaru[100];      char pesan\_diterima[100];      printf("");      scanf("%[^\n]%\*c", kode\_shikamaru);      printf("");      scanf("%[^\n]%\*c", pesan\_diterima);      if (strlen(kode\_shikamaru) != strlen(pesan\_diterima)) {          printf("Panjang kalimat berbeda, pesan palsu\n");          return;      }      char hasil[100];      int bintang = 0, pagar = 0;      for (int i = 0; i < strlen(kode\_shikamaru); i++) {          if (kode\_shikamaru[i] == pesan\_diterima[i]) {              hasil[i] = '\*';              bintang++;          } else {              hasil[i] = '#';              pagar++;          }      }      hasil[strlen(kode\_shikamaru)] = '\0';      printf("%s\n", hasil);      printf("\* = %d\n", bintang);      printf("# = %d\n", pagar);      if (bintang >= pagar) {          printf("Pesan Asli\n");      } else {          printf("Pesan Palsu\n");      }  }  int main() {      shikamaru();      return 0;  } |

Tabel 7 Source Code PRAK604 Bahasa C

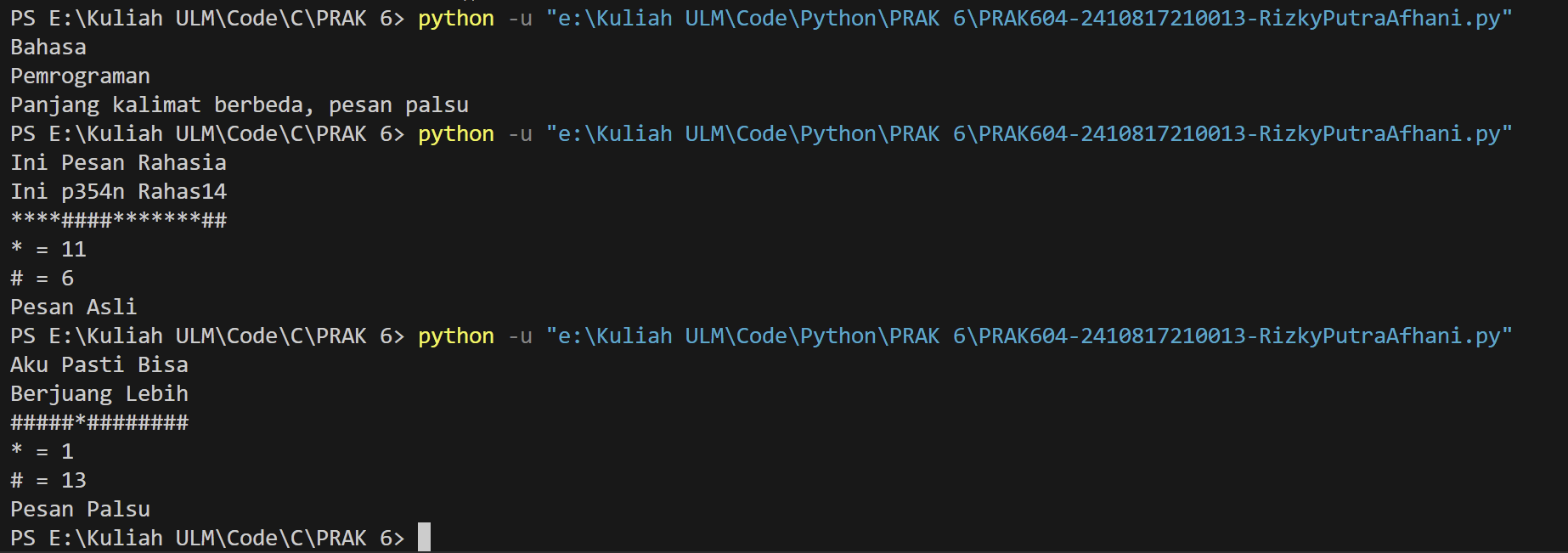
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | def shikamaru():      kode\_shikamaru = input("")        pesan\_diterima = input("")      if len(kode\_shikamaru) != len(pesan\_diterima):          print("Panjang kalimat berbeda, pesan palsu")          return      hasil = []      bintang = 0      pagar = 0      for k, p in zip(kode\_shikamaru, pesan\_diterima):          if k == p:              hasil.append('\*')              bintang += 1          else:              hasil.append('#')              pagar += 1      print("".join(hasil))      print(f"\* = {bintang}")      print(f"# = {pagar}")        if bintang >= pagar:          print("Pesan Asli")      else:          print("Pesan Palsu")  shikamaru() |

Tabel 8 Source Code PRAK604 Bahasa Python

## Output Program

Gambar 7. Screenshot Output PRAK604 Bahasa C

Gambar 8. Screenshot Output PRAK604 Bahasa Python



## Pembahasan

* **Bahasa C:**

Program ini digunakan untuk memverifikasi apakah dua pesan yang diberikan adalah pesan asli atau palsu, berdasarkan kesamaan karakter pada setiap posisi. Fungsi shikamaru dimulai dengan menerima dua input berupa string: kode\_shikamaru dan pesan\_diterima. Jika panjang kedua pesan berbeda, program akan mengeluarkan pesan "Panjang kalimat berbeda, pesan palsu" dan berhenti. Jika panjang pesan sama, program membandingkan karakter-karakter pada posisi yang sama antara kedua pesan. Jika karakter pada posisi yang sama cocok, program menandainya dengan simbol \*, jika tidak cocok, ditandai dengan simbol #. Program juga menghitung berapa banyak \* (karakter yang cocok) dan # (karakter yang tidak cocok), lalu menampilkan hasil tersebut. Akhirnya, jika jumlah \* lebih banyak atau sama dengan jumlah #, pesan dianggap "Asli", jika tidak, dianggap "Palsu". Program ini memberikan output berupa hasil perbandingan, jumlah kecocokan, dan jenis pesan.

* **Python:**

Program ini membandingkan dua pesan karakter demi karakter. Jika panjang pesan berbeda, akan muncul pesan kesalahan. Jika panjangnya sama, program menandai kecocokan karakter dengan \* dan ketidakcocokan dengan #, lalu menghitung jumlahnya. Jika jumlah \* lebih banyak atau sama dengan #, pesan dianggap "Asli", jika tidak, dianggap "Palsu".

# 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

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | #include <stdio.h>  void input\_matriks(int n, int matriks[][n], char nama) {      printf("Matriks %c\n", nama);      for (int i = 0; i < n; i++) {          for (int j = 0; j < n; j++) {              scanf("%d", &matriks[i][j]);          }      }  }  void cetak\_matriks(int n, int matriks[][n], char nama) {      printf("Matriks %cXB\n", nama);      for (int i = 0; i < n; i++) { |

|  |  |
| --- | --- |
| 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 | for (int j = 0; j < n; j++) {              printf("%d ", matriks[i][j]);          }          printf("\n");      }  }  void perkalian\_matriks(int n, int A[][n], int B[][n], int hasil[][n]) {      for (int i = 0; i < n; i++) {          for (int j = 0; j < n; j++) {              hasil[i][j] = 0;              for (int k = 0; k < n; k++) {                  hasil[i][j] += A[i][k] \* B[k][j];              }          }      }  }  int main() {      int n;      scanf("%d", &n);      int A[n][n], B[n][n], hasil[n][n];      input\_matriks(n, A, 'A');      input\_matriks(n, B, 'B');      perkalian\_matriks(n, A, B, hasil);      cetak\_matriks(n, hasil, 'A');      return 0;  } |

Tabel 9 Source Code PRAK605 Bahasa C

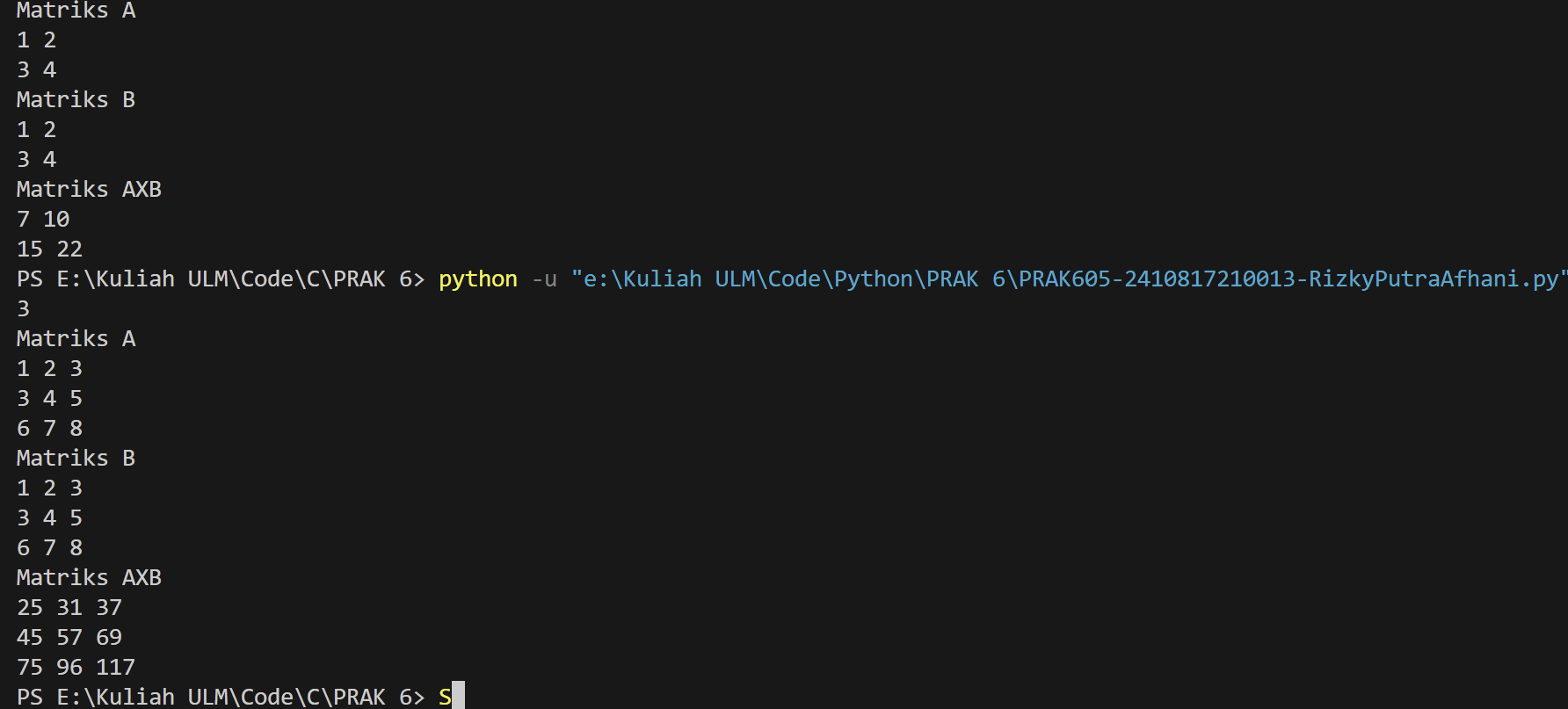
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | def input\_matriks(n, nama):      print(f"Matriks {nama}")      matriks = []      for i in range(n):          row = list(map(int, input().split()))          matriks.append(row)      return matriks  def cetak\_matriks(matriks, nama):      print(f"Matriks {nama}") |

|  |  |
| --- | --- |
| 10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | for row in matriks:          print(" ".join(map(str, row)))  def perkalian\_matriks(A, B, n):      hasil = [[0] \* n for \_ in range(n)]      for i in range(n):          for j in range(n):              for k in range(n):                  hasil[i][j] += A[i][k] \* B[k][j]      return hasil  def main():      n = int(input(""))      matriks\_A = input\_matriks(n, "A")      matriks\_B = input\_matriks(n, "B")      hasil\_kali = perkalian\_matriks(matriks\_A, matriks\_B, n)      cetak\_matriks(hasil\_kali, "AXB")  if \_\_name\_\_ == "\_\_main\_\_":      main() |

Tabel 10 Source Code PRAK605 Bahasa Python

## Output Program

Gambar 9. Screenshot Output PRAK605 Bahasa C



Gambar 10. Screenshot Output PRAK605 Bahasa Python

## Pembahasan

* **Bahasa C:**

Program ini melakukan perkalian dua matriks persegi berukuran n x n. Di dalam fungsi main, program pertama-tama meminta input ukuran matriks n. Kemudian, dua matriks, A dan B, diinputkan melalui fungsi input\_matriks. Setelah itu, fungsi perkalian\_matriks digunakan untuk mengalikan matriks A dan B, dan hasil perkalian disimpan dalam matriks hasil. Proses perkalian matriks dilakukan dengan cara menjumlahkan hasil perkalian elemen baris matriks A dengan kolom matriks B untuk setiap elemen matriks hasil. Setelah perkalian selesai, hasil matriks ditampilkan dengan menggunakan fungsi cetak\_matriks. Matriks yang ditampilkan adalah hasil perkalian dari matriks A dan B, yang disebut sebagai "Matriks AXB".

* **Python:**

Program ini melakukan perkalian dua matriks persegi berukuran n x n. Pertama, program meminta input ukuran matriks n. Kemudian, dua matriks A dan B dimasukkan melalui fungsi input\_matriks, yang meminta pengguna memasukkan elemen-elemen baris per baris untuk masing-masing matriks. Fungsi perkalian\_matriks digunakan untuk menghitung hasil perkalian kedua matriks. Perkalian dilakukan dengan menjumlahkan hasil perkalian elemen baris matriks A dengan kolom matriks B untuk setiap elemen hasil. Matriks hasil perkalian disimpan dalam variabel hasil\_kali. Setelah perkalian selesai, fungsi cetak\_matriks digunakan untuk menampilkan matriks hasil tersebut dalam format yang sesuai. Program kemudian menampilkan matriks hasil perkalian dengan label "Matriks AXB".

# TAUTAN GIT

<https://github.com/rizkyputraafhani26/Pemrograman-1>