

**LAPORAN PRAKTIKUM
PEMROGRAMAN I
MODUL 6**



Array

Oleh:

Ashma Salimah

NIM. 2410817220015

**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 : Ashma Salimah
NIM : 2410817220015

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Ryan Rizky Rahmadi
NIM. 2210817310001

Helda Yunita, S.Kom., M.Kom.
NIP. 199106192024062001

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI	3
DAFTAR GAMBAR.....	5
DAFTAR TABEL	6
SOAL 1	7
A. Source Code	8
• C.....	8
• Python	8
B. Output Program.....	9
• C.....	9
• Python	9
C. Pembahasan.....	10
• C.....	10
• Python	11
SOAL 2.....	12
A. Source Code	13
• C.....	13
• Python	13
B. Output Program.....	14
• C.....	14
• Python	14
C. Pembahasan.....	15
• C.....	15
• Python	16
SOAL 3.....	17
A. Source Code	18
• C.....	18
• Python	18
B. Output Program.....	19

• C.....	19
• Python	19
C. Pembahasan.....	20
• C.....	20
• Python	21
SOAL 4.....	23
A. Source Code	24
• C.....	24
• Python	25
B. Output Program.....	26
• C.....	26
• Python	26
C. Pembahasan.....	27
• C.....	27
• Python	29
SOAL 5.....	30
A. Source Code	31
• C.....	31
• Python	32
B. Output Program.....	33
• C.....	33
• Python	33
C. Pembahasan.....	34
• C.....	34
• Python	35

DAFTAR GAMBAR

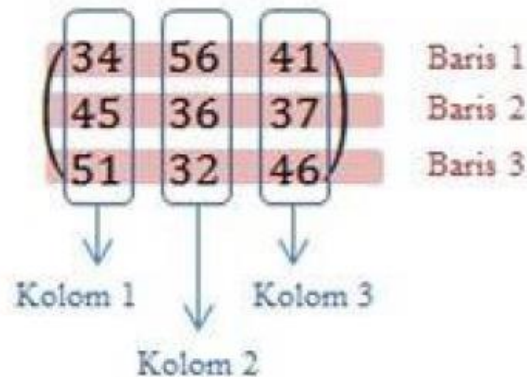
Gambar 1 Screenshot 1 Hasil Jawaban Soal 1	9
Gambar 2 Screenshot 2 Hasil Jawaban Soal 1	9
Gambar 3 Screenshot 1 Hasil Jawaban Soal 2	14
Gambar 4 Screenshot 2 Hasil Jawaban Soal 2	14
Gambar 5 Screenshot 1 Hasil Jawaban Soal 3	19
Gambar 6 Screenshot 2 Hasil Jawaban Soal 3	19
Gambar 7 Screenshot 1 Hasil Jawaban Soal 4	26
Gambar 8 Screenshot 2 Hasil Jawaban Soal 4	26
Gambar 9 Screenshot 1 Hasil Jawaban Soal 5	33
Gambar 10 Screenshot 2 Hasil Jawaban Soal 5	33

DAFTAR TABEL

Table 1 Source Code 1 Soal 1	8
Table 2 Source Code 2 Soal 1	8
Table 3 Source Code 1 Soal 2	13
Table 4 Source 2 Code Soal 2	13
Table 5 Source Code 1 Soal 3	18
Table 6 Source Code 2 Soal 3	18
Table 7 Source Code 1 Soal 4	24
Table 8 Source Code 2 Soal 4	25
Table 9 Source Code 1 Soal 5	31
Table 10 Source Code 2 Soal 5	32

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
1 2 3 4 5 6	4 5 6
3 3	34 56 41
34 56 41 45 36 37 51 32 46	45 36 37
	51 32 46
4 5	1 1 1 1 2
1 1 1 1 2 3 5 6 4 5 8 7 9 6 5 4 1 2 5 6	3 5 6 4 5
	8 7 9 6 5
	4 1 2 5 6

A. Source Code

- C

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

Table 1 Source Code 1 Soal 1

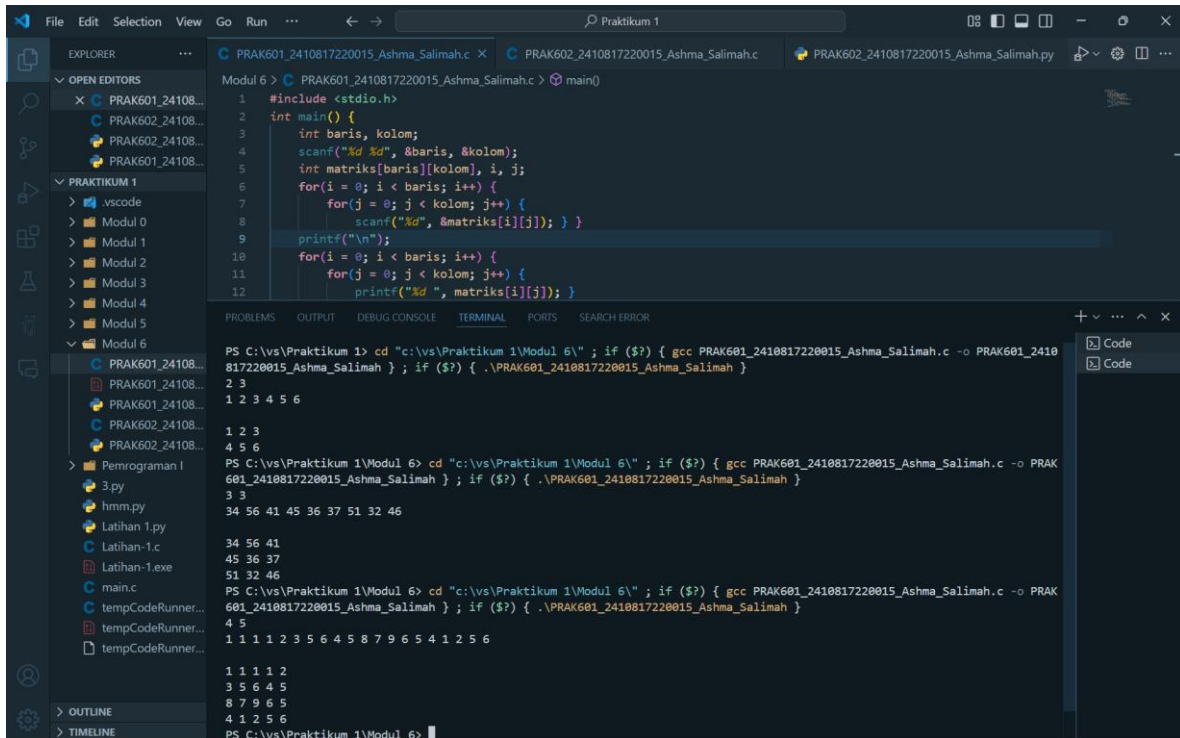
- Python

1	baris, kolom = map(int, input("").split())
2	angka = list(map(int, input("").split()))
3	matriks = []
4	for i in range (baris) :
5	matriks.append(angka[i * kolom:(i + 1) * kolom])
6	print("")
7	for i in range (baris) :
8	for j in range (kolom) :
9	print(f"{matriks[i][j]}", end=" ")
10	print("")

Table 2 Source Code 2 Soal 1

B. Output Program

• C



The screenshot shows the Visual Studio Code interface with a C program in the editor and its output in the terminal. The C code defines a 5x5 matrix and prints it row by row. The terminal shows the execution of the program, displaying the matrix values.

```
#include <stdio.h>

int main() {
    int baris, kolom;
    scanf("%d %d", &baris, &kolom);
    int matriks[baris][kolom];
    for(i = 0; i < baris; i++) {
        for(j = 0; j < kolom; j++) {
            scanf("%d", &matriks[i][j]);
        }
        printf("\n");
    }
    for(i = 0; i < baris; i++) {
        for(j = 0; j < kolom; j++) {
            printf("%d ", matriks[i][j]);
        }
    }
}
```

```
PS C:\vs\Praktikum 1> cd "c:\vs\Praktikum 1\Modul 6\" ; if ($?) { gcc PRAK601_2410817220015_Ashma_Salimah.c -o PRAK601_2410817220015_Ashma_Salimah }
2 3
1 2 3 4 5 6

1 2 3
4 5 6

PS C:\vs\Praktikum 1\Modul 6> cd "c:\vs\Praktikum 1\Modul 6\" ; if ($?) { gcc PRAK601_2410817220015_Ashma_Salimah.c -o PRAK601_2410817220015_Ashma_Salimah }
3 3
34 56 41 45 36 37 51 32 46

34 56 41
45 36 37
51 32 46

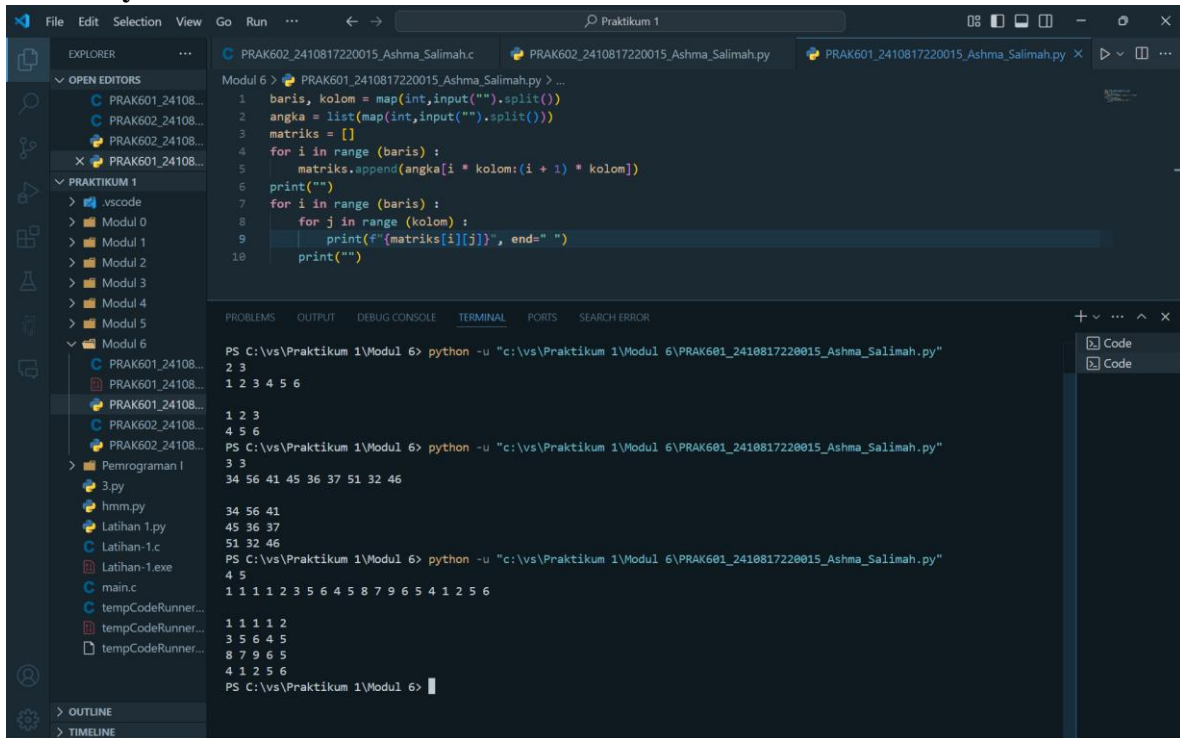
PS C:\vs\Praktikum 1\Modul 6> cd "c:\vs\Praktikum 1\Modul 6\" ; if ($?) { gcc PRAK601_2410817220015_Ashma_Salimah.c -o PRAK601_2410817220015_Ashma_Salimah }
4 5
1 1 1 1 2 3 5 6 4 5 8 7 9 6 5 4 1 2 5 6

1 1 1 1 2
3 5 6 4 5
8 7 9 6 5
4 1 2 5 6

PS C:\vs\Praktikum 1\Modul 6>
```

Gambar 1 Screenshot 1 Hasil Jawaban Soal 1

• Python



The screenshot shows the Visual Studio Code interface with a Python program in the editor and its output in the terminal. The Python code uses list comprehension to create a 5x5 matrix and prints it row by row. The terminal shows the execution of the program, displaying the matrix values.

```
baris, kolom = map(int, input("").split())
angka = list(map(int, input("").split()))
matriks = []
for i in range (baris) :
    matriks.append(angka[i * kolom:(i + 1) * kolom])
print("")
for i in range (baris) :
    for j in range (kolom) :
        print(f"{matriks[i][j]}", end=" ")
    print("")
```

```
PS C:\vs\Praktikum 1\Modul 6> python -u "c:\vs\Praktikum 1\Modul 6\PRAK601_2410817220015_Ashma_Salimah.py"
2 3
1 2 3 4 5 6

1 2 3
4 5 6

PS C:\vs\Praktikum 1\Modul 6> python -u "c:\vs\Praktikum 1\Modul 6\PRAK601_2410817220015_Ashma_Salimah.py"
3 3
34 56 41 45 36 37 51 32 46

34 56 41
45 36 37
51 32 46

PS C:\vs\Praktikum 1\Modul 6> python -u "c:\vs\Praktikum 1\Modul 6\PRAK601_2410817220015_Ashma_Salimah.py"
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

PS C:\vs\Praktikum 1\Modul 6>
```

Gambar 2 Screenshot 2 Hasil Jawaban Soal 1

C. Pembahasan

- C

Pada baris pertama, terdapat `#include <stdio.h>` yang memiliki fungsi menyertakan library standar, juga sebagai kode input maupun output yang akan dilibatkan dalam program.

Pada baris kedua, terdapat `int main() { . int` berfungsi sebagai penunjuk bahwa `main()` membalikkan nilai integer. Fungsi inilah yang menjadi titik awal dan titik akhir eksekusi program. Tanda kurung kurawal `({})` berfungsi untuk mendefinisikan blok kode, sebagai pendefinisian tubuh fungsi. `{` berfungsi sebagai awal eksekusi program dan `}` berfungsi sebagai akhir tubuh atau akhir eksekusi program.

`scanf` berfungsi untuk membaca string pada input. `%d` berfungsi untuk menerima tipe data integer (bilangan bulat). Tanda `&` sebelum variabel adalah operator alamat yang berfungsi untuk memberi alamat memori dari input akan dibaca dan disimpan pada variabel, tanda tersebut dibutuhkan karena fungsi `scanf` membutuhkan alamat memori untuk menempatkan data yang dimasukkan.

```
int matriks[baris][kolom], i, j;
```

`int` untuk memasukkan fungsi nilai integer atau bilangan bulat. `matriks[baris][kolom]` adalah array dengan jumlah dimensi `[baris]` dan `[kolom]`.

`i` dan `j` merupakan variabel loop untuk literasi melalui `baris` dan `kolom`.

```
for(i = 0; i < baris; i++) {  
    for(j = 0; j < kolom; j++) {
```

`for` adalah loop yang berfungsi untuk melakukan pengulangan blok kode.

Pada `i = 0`, variabel `i` diinisialisasikan dengan 0, berarti pengulangan dimulai dari 0.

Kemudian pada `i < baris;`, loop akan terus berjalan selama `i` kurang dari `baris`, jika telah mencapai, loop akan berhenti.

Pada `i++`, loop akan meningkat sebanyak satu unit.

`for(j = 0; j < kolom; j++)` adalah nested loop, yaitu loop yang berjalan di dalam loop. `i` dan `j` mewakili koordinat dalam matriks.

```
printf("%d", matriks[i][j]);
```

`printf("%d", matriks[i][j]);` berfungsi untuk menampilkan suatu keluaran atau mencetak output pada layar. `%d` berfungsi untuk menerima tipe data integer (bilangan bulat) pada variabel yang terletak setelah tanda koma.

; pada setiap akhir fungsi, berfungsi untuk menandai akhir dari setiap pernyataan atau statement dan bukan sebagai pemisah antara dua pernyataan.

Pada `return 0;`, `return` berfungsi sebagai pembalik nilai dari variabel pada tempat di mana fungsi dipanggil.

- **Python**

```
baris, kolom = map(int, input("").split())
```

```
angka = list(map(int, input("").split()))
```

`list` berfungsi untuk menyimpan kumpulan elemen dalam satu variabel.

`map(int, ...)` mengkonversi elemen dalam string menjadi integer.

`input("")` meminta pengguna untuk menginput data atau string.

`.split()` memisahkan inputan berdasarkan spasi dan menghasilkan daftar string.

```
matriks = []
```

membuat daftar kosong untuk menyimpan elemen dalam bentuk array (list).

```
for i in range (baris) :
```

Loop yang membuat iterasi dari 0 sampai `baris - 1`.

```
    for j in range (kolom) :
```

adalah loop yang berjalan dalam loop.

```
matriks.append(angka[i * kolom:(i + 1) * kolom])
```

```
append(angka[i * kolom:(i + 1) * kolom]),
```

`append` berfungsi untuk menambahkan elemen baru ke dalam daftar `matriks`.

```
print(f"{matriks[i][j]}", end=" ")
```

berfungsi untuk menampilkan suatu keluaran atau mencetak output pada layar. Tanda `f` adalah fungsi f-string, string literal yang terdapat sebelum tanda petik. Tanda kurung kurawal berfungsi untuk menempatkan variabel ke dalam string, dengan memasukkan variabel fungsi yang telah dibuat sebelumnya.

```
end=" ")
```

memisahkan output berdasarkan spasi.

SOAL 2

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

A. Source Code

- **C**

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

Table 3 Source Code 1 Soal 2

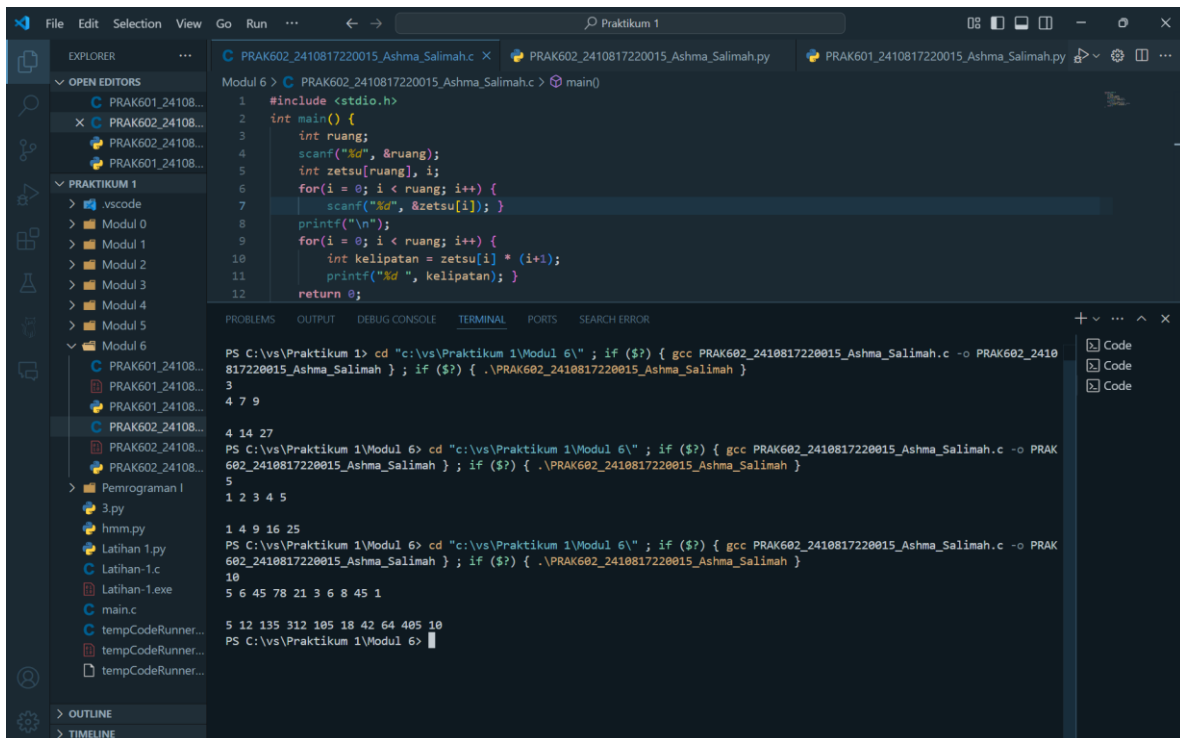
- **Python**

1	ruang = int(input(""))
2	zetsu = list(map(int,input("").split()))
3	print("")
4	for i in range (ruang) :
5	kelipatan = zetsu[i] * (i+1)
6	print(f"{kelipatan}", end=" ")

Table 4 Source 2 Code Soal 2

B. Output Program

- C



The screenshot shows a VS Code editor with a C program in `PRAK602_2410817220015_Ashma_Salimah.c`. The code defines an array `ruang` of size 10, reads 10 integers from input, and calculates the sum of products of `zetsu[i] * (i+1)` for `i` from 0 to 9. The terminal output shows the execution results for three different input sets.

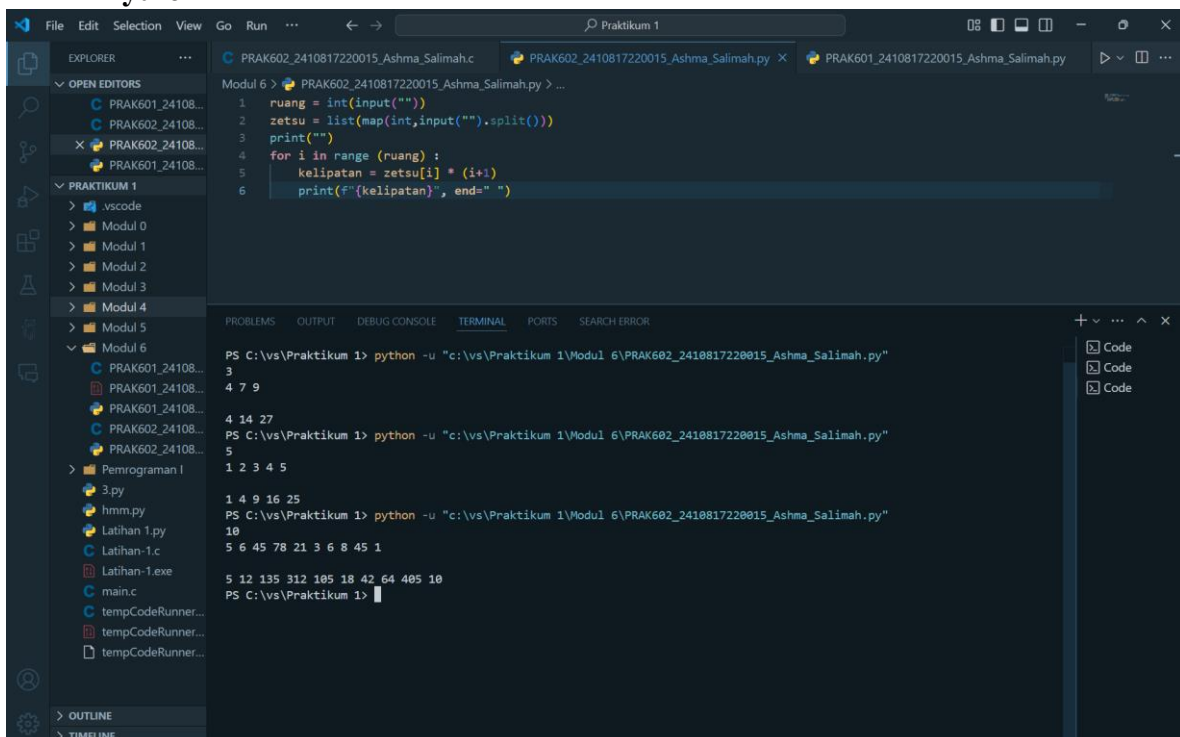
```
#include <stdio.h>
int main() {
    int ruang;
    scanf("%d", &ruang);
    int zetsu[ruang], i;
    for(i = 0; i < ruang; i++) {
        scanf("%d", &zetsu[i]);
    }
    printf("\n");
    for(i = 0; i < ruang; i++) {
        int kelipatan = zetsu[i] * (i+1);
        printf("%d ", kelipatan);
    }
    return 0;
}
```

Terminal Output:

```
PS C:\vs\Praktikum 1> cd "c:\vs\Praktikum 1\Modul 6\" ; if ($?) { gcc PRAK602_2410817220015_Ashma_Salimah.c -o PRAK602_2410817220015_Ashma_Salimah } ; if ($?) { .\PRAK602_2410817220015_Ashma_Salimah }
3
4 7 9
5
4 14 27
PS C:\vs\Praktikum 1\Modul 6> cd "c:\vs\Praktikum 1\Modul 6\" ; if ($?) { gcc PRAK602_2410817220015_Ashma_Salimah.c -o PRAK602_2410817220015_Ashma_Salimah } ; if ($?) { .\PRAK602_2410817220015_Ashma_Salimah }
5
1 2 3 4 5
6
1 4 9 16 25
PS C:\vs\Praktikum 1\Modul 6> cd "c:\vs\Praktikum 1\Modul 6\" ; if ($?) { gcc PRAK602_2410817220015_Ashma_Salimah.c -o PRAK602_2410817220015_Ashma_Salimah } ; if ($?) { .\PRAK602_2410817220015_Ashma_Salimah }
10
5 6 45 78 21 3 6 8 45 1
7
5 12 135 312 105 18 42 64 405 10
PS C:\vs\Praktikum 1\Modul 6>
```

Gambar 3 Screenshot 1 Hasil Jawaban Soal 2

- Python



The screenshot shows a VS Code editor with a Python program in `PRAK602_2410817220015_Ashma_Salimah.py`. The code reads an integer `ruang` and a list of integers `zetsu`, then calculates the sum of products of `zetsu[i] * (i+1)` for `i` from 0 to `ruang-1`. The terminal output shows the execution results for three different input sets.

```
ruang = int(input(""))
zetsu = list(map(int, input("").split()))
print("")
for i in range (ruang) :
    kelipatan = zetsu[i] * (i+1)
    print(f"{kelipatan}", end=" ")
```

Terminal Output:

```
PS C:\vs\Praktikum 1> python -u "c:\vs\Praktikum 1\Modul 6\PRAK602_2410817220015_Ashma_Salimah.py"
3
4 7 9
5
4 14 27
PS C:\vs\Praktikum 1> python -u "c:\vs\Praktikum 1\Modul 6\PRAK602_2410817220015_Ashma_Salimah.py"
5
1 2 3 4 5
6
1 4 9 16 25
PS C:\vs\Praktikum 1> python -u "c:\vs\Praktikum 1\Modul 6\PRAK602_2410817220015_Ashma_Salimah.py"
10
5 6 45 78 21 3 6 8 45 1
7
5 12 135 312 105 18 42 64 405 10
PS C:\vs\Praktikum 1>
```

Gambar 4 Screenshot 2 Hasil Jawaban Soal 2

C. Pembahasan

- C

Pada baris pertama, terdapat `#include <stdio.h>` yang memiliki fungsi menyertakan library standar, juga sebagai kode input maupun output yang akan dilibatkan dalam program.

Pada baris kedua, terdapat `int main() { .int` berfungsi sebagai penunjuk bahwa `main()` membalikkan nilai integer. Fungsi inilah yang menjadi titik awal dan titik akhir eksekusi program. Tanda kurung kurawal `({})` berfungsi untuk mendefinisikan blok kode, sebagai pendefinisian tubuh fungsi. `{` berfungsi sebagai awal eksekusi program dan `}` berfungsi sebagai akhir tubuh atau akhir eksekusi program.

`int ruang;` pada baris ke-3, `int` memiliki fungsi sama, yaitu membalikkan nilai integer tetapi pada variabel `ruang`.

```
scanf("%d", &ruang);
```

`scanf` berfungsi untuk membaca string pada input. `%d` berfungsi untuk menerima tipe data integer (bilangan bulat). Tanda `&` sebelum variabel berfungsi untuk memberi alamat memori dari input akan dibaca dan disimpan pada variabel.

```
int zetsu[ruang], i;
```

`int` untuk memasukkan fungsi nilai integer atau bilangan bulat. `zetsu[ruang]` adalah array dengan jumlah dimensi `[ruang]`.

`i` dan `j` merupakan variabel loop untuk literasi melalui `ruang`.

```
scanf("%d", &zetsu[i]);
```

`scanf` berfungsi untuk membaca string pada input. `%d` berfungsi untuk menerima tipe data integer (bilangan bulat). Tanda `&` sebelum variabel adalah operator alamat yang berfungsi untuk memberi alamat memori dari input akan dibaca dan disimpan pada variabel, tanda tersebut dibutuhkan karena fungsi `scanf` membutuhkan alamat memori untuk menempatkan data yang dimasukkan.

`zetsu` adalah nama array, `[i]` menunjukkan indeks array yang menjadi tempat nilai dimasukkan dan disimpan.

```
for(i = 0; i < ruang; i++) {
```

`for` adalah loop yang berfungsi untuk melakukan pengulangan blok kode.

Pada `i = 0`, variabel `i` diinisialisasikan dengan 0, berarti pengulangan dimulai dari 0.

Kemudian pada `i < ruang;`, loop akan terus berjalan selama `i` kurang dari `ruang`, jika telah mencapai, loop akan berhenti.

Pada `i++`, loop akan meningkat sebanyak satu unit.

`printf("%d", kelipatan);` berfungsi untuk menampilkan suatu keluaran atau mencetak output pada layar. `%d` berfungsi untuk menerima tipe data integer (bilangan bulat) pada variabel yang terletak setelah tanda koma.

`;` pada setiap akhir fungsi, berfungsi untuk menandai akhir dari setiap pernyataan atau statement dan bukan sebagai pemisah antara dua pernyataan.

Pada `return 0;`, `return` berfungsi sebagai pembalik nilai dari variabel pada tempat di mana fungsi dipanggil.

- **Python**

```
ruang = int(input(""))
```

```
zetsu = list(map(int,input("").split()))
```

`list` berfungsi untuk menyimpan kumpulan elemen dalam satu variabel.

`map(int, ...)` mengkonversi elemen dalam string menjadi integer.

`input("")` meminta pengguna untuk menginput data atau string.

`.split()` memisahkan inputan berdasarkan spasi dan menghasilkan daftar string.

```
for i in range (ruang) :
```

Loop yang membuat iterasi dari 0 sampai `ruang - 1`.

`print(f"{kelipatan}", end=" ")` berfungsi untuk menampilkan suatu keluaran atau mencetak output pada layar. Tanda `f` adalah fungsi f-string, string literal yang terdapat sebelum tanda petik. Tanda kurung kurawal berfungsi untuk menempatkan variabel ke dalam string, dengan memasukkan variabel fungsi yang telah dibuat sebelumnya.

`end=" "` memisahkan output berdasarkan spasi.

SOAL 3

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

A. Source Code

• C

1	#include <stdio.h>
2	int main() {
3	int baris, kolom;
4	scanf("%d %d", &baris, &kolom);
5	if (baris != kolom) {
6	printf("Jumlah tidak sama"); }
7	else {
8	int matriks1[baris], matriks2[kolom];
9	for(int i = 0; i < baris; i++) {
10	scanf("%d", &matriks1[i]); }
11	for(int i = 0; i < kolom; i++) {
12	scanf("%d", &matriks2[i]); }
13	printf("\n");
14	for(int i = 0; i < baris; i++) {
15	int hasil = matriks1[i] * matriks2[i];
16	printf("%d ", hasil); } }
17	return 0;
18	}

Table 5 Source Code 1 Soal 3

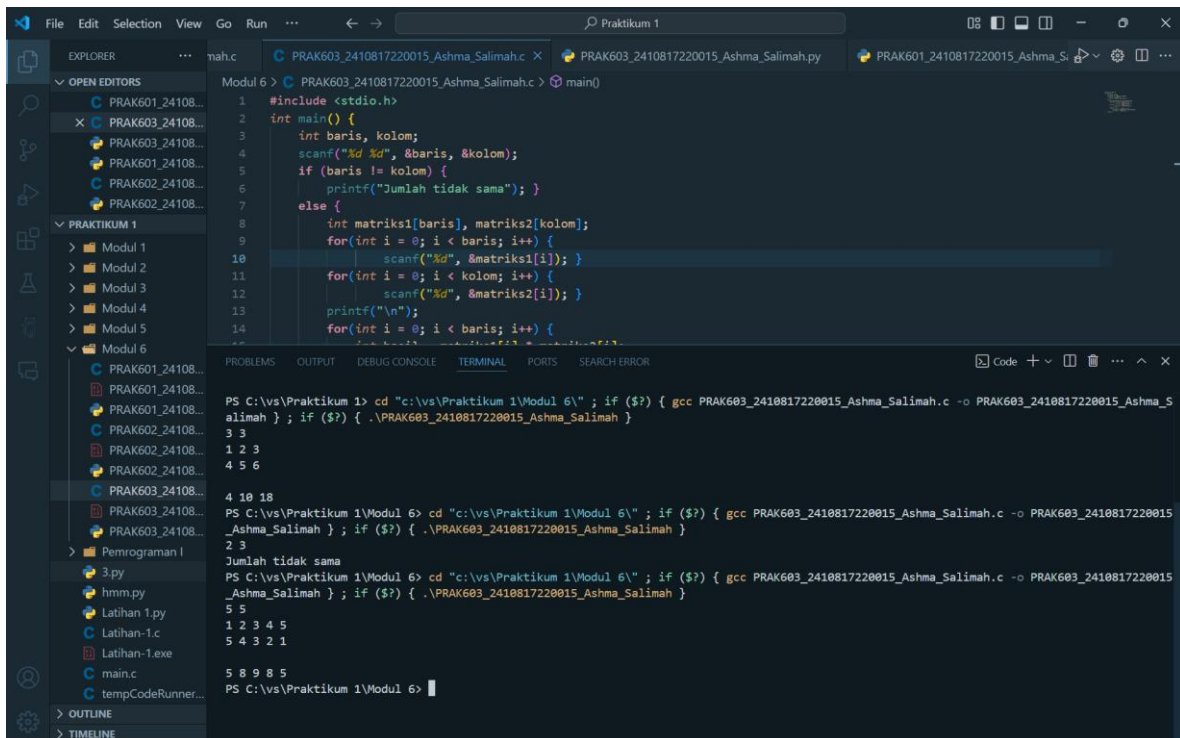
• Python

1	baris, kolom = map(int, input("").split())
2	if baris != kolom :
3	print("Jumlah tidak sama")
4	else :
5	matriks1 = list(map(int, input("").split()))
6	matriks2 = list(map(int, input("").split()))
7	print("")
8	for i in range (baris) :
9	hasil = matriks1[i] * matriks2[i]
10	print(f"{hasil}", end=" ")

Table 6 Source Code 2 Soal 3

B. Output Program

- C



The screenshot shows the Visual Studio Code interface with a C program in the main editor. The program checks if the number of rows and columns are equal. If not, it prints "Jumlah tidak sama". If yes, it multiplies two matrices and prints the result. The terminal shows the execution of the program with inputs 3 3 and 4 5 6, resulting in the output 1 2 3 and 4 5 6. The Explorer panel shows the project structure with files like main.c and tempCodeRunner.c.

```
#include <stdio.h>

int main() {
    int baris, kolom;
    scanf("%d %d", &baris, &kolom);
    if (baris != kolom) {
        printf("Jumlah tidak sama");
    }
    else {
        int matriks1[baris], matriks2[kolom];
        for(int i = 0; i < baris; i++) {
            scanf("%d", &matriks1[i]);
        }
        for(int i = 0; i < kolom; i++) {
            scanf("%d", &matriks2[i]);
        }
        printf("\n");
        for(int i = 0; i < baris; i++) {
            int hasil = 0;
            for(int j = 0; j < kolom; j++) {
                hasil += matriks1[i] * matriks2[j];
            }
            printf("%d ", hasil);
        }
        printf("\n");
    }
}
```

Terminal Output:

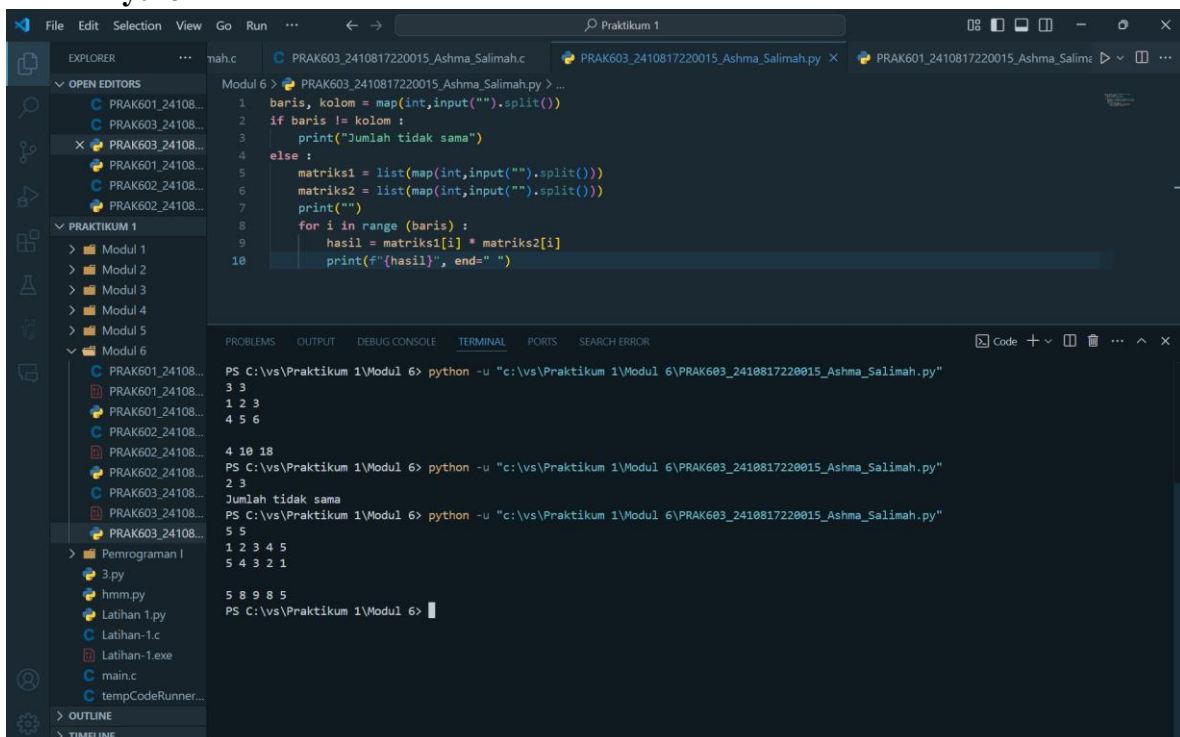
```
PS C:\vs\Praktikum 1> cd "c:\vs\Praktikum 1\Modul 6\"; if ($?) { gcc PRAK603_2410817220015_Ashma_Salimah.c -o PRAK603_2410817220015_Ashma_Salimah.exe; if ($?) { .\PRAK603_2410817220015_Ashma_Salimah.exe }
3 3
1 2 3
4 5 6

4 10 18
PS C:\vs\Praktikum 1\Modul 6> cd "c:\vs\Praktikum 1\Modul 6\"; if ($?) { gcc PRAK603_2410817220015_Ashma_Salimah.c -o PRAK603_2410817220015_Ashma_Salimah.exe; if ($?) { .\PRAK603_2410817220015_Ashma_Salimah.exe }
2 3
Jumlah tidak sama
PS C:\vs\Praktikum 1\Modul 6> cd "c:\vs\Praktikum 1\Modul 6\"; if ($?) { gcc PRAK603_2410817220015_Ashma_Salimah.c -o PRAK603_2410817220015_Ashma_Salimah.exe; if ($?) { .\PRAK603_2410817220015_Ashma_Salimah.exe }
5 5
1 2 3 4 5
5 4 3 2 1

5 8 9 8 5
PS C:\vs\Praktikum 1\Modul 6>
```

Gambar 5 Screenshot 1 Hasil Jawaban Soal 3

- Python



The screenshot shows the Visual Studio Code interface with a Python program in the main editor. The program checks if the number of rows and columns are equal. If not, it prints "Jumlah tidak sama". If yes, it multiplies two matrices and prints the result. The terminal shows the execution of the program with inputs 3 3 and 4 5 6, resulting in the output 1 2 3 and 4 5 6. The Explorer panel shows the project structure with files like main.c and tempCodeRunner.c.

```
baris, kolom = map(int, input("").split())
if baris != kolom:
    print("Jumlah tidak sama")
else:
    matriks1 = list(map(int, input("").split()))
    matriks2 = list(map(int, input("").split()))
    print("")
    for i in range (baris):
        hasil = matriks1[i] * matriks2[i]
        print(f"{hasil}", end=" ")
    print("\n")
```

Terminal Output:

```
PS C:\vs\Praktikum 1\Modul 6> python -u "c:\vs\Praktikum 1\Modul 6\PRAK603_2410817220015_Ashma_Salimah.py"
3 3
1 2 3
4 5 6

4 10 18
PS C:\vs\Praktikum 1\Modul 6> python -u "c:\vs\Praktikum 1\Modul 6\PRAK603_2410817220015_Ashma_Salimah.py"
2 3
Jumlah tidak sama
PS C:\vs\Praktikum 1\Modul 6> python -u "c:\vs\Praktikum 1\Modul 6\PRAK603_2410817220015_Ashma_Salimah.py"
5 5
1 2 3 4 5
5 4 3 2 1

5 8 9 8 5
PS C:\vs\Praktikum 1\Modul 6>
```

Gambar 6 Screenshot 2 Hasil Jawaban Soal 3

C. Pembahasan

- C

Pada baris pertama, terdapat `#include <stdio.h>` yang memiliki fungsi menyertakan library standar, juga sebagai kode input maupun output yang akan dilibatkan dalam program.

Pada baris ke-2, terdapat `int main() { .int` berfungsi sebagai penunjuk bahwa `main()` membalikkan nilai integer. Fungsi inilah yang menjadi titik awal dan titik akhir eksekusi program. Tanda kurung kurawal (`{}`) berfungsi untuk mendefinisikan blok kode, sebagai pendefinisian tubuh fungsi. `{` berfungsi sebagai awal eksekusi program dan `}` berfungsi sebagai akhir tubuh atau akhir eksekusi program.

`int baris, kolom;` pada baris ke-3, `int` memiliki fungsi sama, yaitu membalikkan nilai integer tetapi pada variabel `baris, kolom`.

```
scanf("%d", &ruang);
```

`scanf` berfungsi untuk membaca string pada input. `%d` berfungsi untuk menerima tipe data integer (bilangan bulat). Tanda `&` sebelum variabel berfungsi untuk memberi alamat memori dari input akan dibaca dan disimpan pada variabel.

```
int matriks1[baris], matriks2[kolom];
```

`int` untuk memasukkan fungsi nilai integer atau bilangan bulat. `matriks1, matriks2` adalah array dengan jumlah dimensi `[baris]` dan `[ruang]`.

```
scanf("%d", &matriks1[i]);
```

```
scanf("%d", &matriks2[i]);
```

`scanf` berfungsi untuk membaca string pada input. `%d` berfungsi untuk menerima tipe data integer (bilangan bulat). Tanda `&` sebelum variabel adalah operator alamat yang berfungsi untuk memberi alamat memori dari input akan dibaca dan disimpan pada variabel, tanda tersebut dibutuhkan karena fungsi `scanf` membutuhkan alamat memori untuk menempatkan data yang dimasukkan.

`matriks1` dan `matriks2` adalah nama array, `[i]` menunjukkan indeks array yang menjadi tempat nilai dimasukkan dan disimpan.

`if()` dan `else` adalah fungsi kondisional yang berguna untuk mengontrol alur program berdasarkan kondisi tertentu.

`if()` untuk memeriksa kondisi. Jika kondisi tersebut benar, maka kode di dalamnya akan dieksekusi.

`else` berfungsi untuk menangani semua kasus yang tidak terpenuhi oleh kondisi yang ada di `if()`.

```
for(i = 0; i < baris; i++) {  
for(i = 0; i < kolom; i++) {
```

for adalah loop yang berfungsi untuk melakukan pengulangan blok kode.

Pada `i = 0` , variabel `i` diinisialisasikan dengan 0, berarti pengulangan dimulai dari 0.

Kemudian pada `i < baris;` , loop akan terus berjalan selama `i` kurang dari `baris`, jika telah mencapai, loop akan berhenti.

Pada `i++`, loop akan meningkat sebanyak satu unit.

`printf("%d", hasil);` berfungsi untuk menampilkan suatu keluaran atau mencetak output pada layar. `%d` berfungsi untuk menerima tipe data integer (bilangan bulat) pada variabel yang terletak setelah tanda koma.

`;` pada setiap akhir fungsi, berfungsi untuk menandai akhir dari setiap pernyataan atau statement dan bukan sebagai pemisah antara dua pernyataan.

Pada `return 0;` , `return` berfungsi sebagai pembalik nilai dari variabel pada tempat di mana fungsi dipanggil.

- **Python**

```
baris, kolom = map(int,input("").split())
```

```
matriks1 = list(map(int,input("").split()))
```

```
matriks2 = list(map(int,input("").split()))
```

`list` berfungsi untuk menyimpan kumpulan elemen dalam satu variabel.

`map(int,` mengkonversi elemen dalam string menjadi integer.

`input("")` meminta pengguna untuk menginput data atau string.

`.split())` memisahkan inputan berdasarkan spasi dan menghasilkan daftar string.

`if`, dan `else` adalah fungsi kondisional yang berguna untuk mengontrol alur program berdasarkan kondisi tertentu.

`if` untuk memeriksa kondisi. Jika kondisi tersebut benar, maka kode di dalamnya akan dieksekusi.

`else` berfungsi untuk menangani semua kasus yang tidak terpenuhi oleh kondisi yang ada di `if`.

```
for i in range (baris) :
```

Loop yang membuat iterasi dari 0 sampai `baris - 1`.

`print(f"{hasil}", end=" ")` berfungsi untuk menampilkan suatu keluaran atau mencetak output pada layar. Tanda `f` adalah fungsi f-string, string literal yang terdapat sebelum tanda petik. Tanda kurung kurawal berfungsi untuk menempatkan variabel ke dalam string, dengan memasukkan variabel fungsi yang telah dibuat sebelumnya.

`end=" ")` memisahkan output berdasarkan spasi.

SOAL 4

4. Shikamaru merupakan seorang anggota anbu Desa Konoha. Pada saat memata-matai Desa Iwakure, 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 Ini p354n Rahas14	*** #####* *****## * = 8 # = 7 Pesan Asli
Aku Pasti Bisa Berjuang lebih	#####*##### * = 1 # = 13 Pesan Palsu

A. Source Code

• C

1	#include <stdio.h>
2	#include <string.h>
3	int main() {
4	char kode[50], pesan[50];
5	scanf("%[^\\n]*c", &pesan);
6	scanf("%[^\\n]*c", &kode);
7	printf("\\n");
8	int bintang = 0;
9	int pagar = 0;
10	if (strlen(pesan) == strlen(kode)) {
11	for(int i = 0; i < strlen(pesan); i++) {
12	if (pesan[i] == ' ' && kode[i] == ' ') {
13	printf(" "); }
14	else if (pesan[i] == kode[i]) {
15	printf("*");
16	bintang++; }
17	else if (pesan[i] != kode[i]) {
18	printf("#");
19	pagar++; }
20	else {} }
21	printf("\\n* = %d \\n", bintang);
22	printf("# = %d \\n", pagar);
23	if (bintang >= pagar) { printf("Pesan Asli"); }
24	else { printf("Pesan Palsu"); } }
25	else { printf("Panjang kalimat berbeda, pesan
26	palsu"); }
27	return 0;
27	}

Table 7 Source Code 1 Soal 4

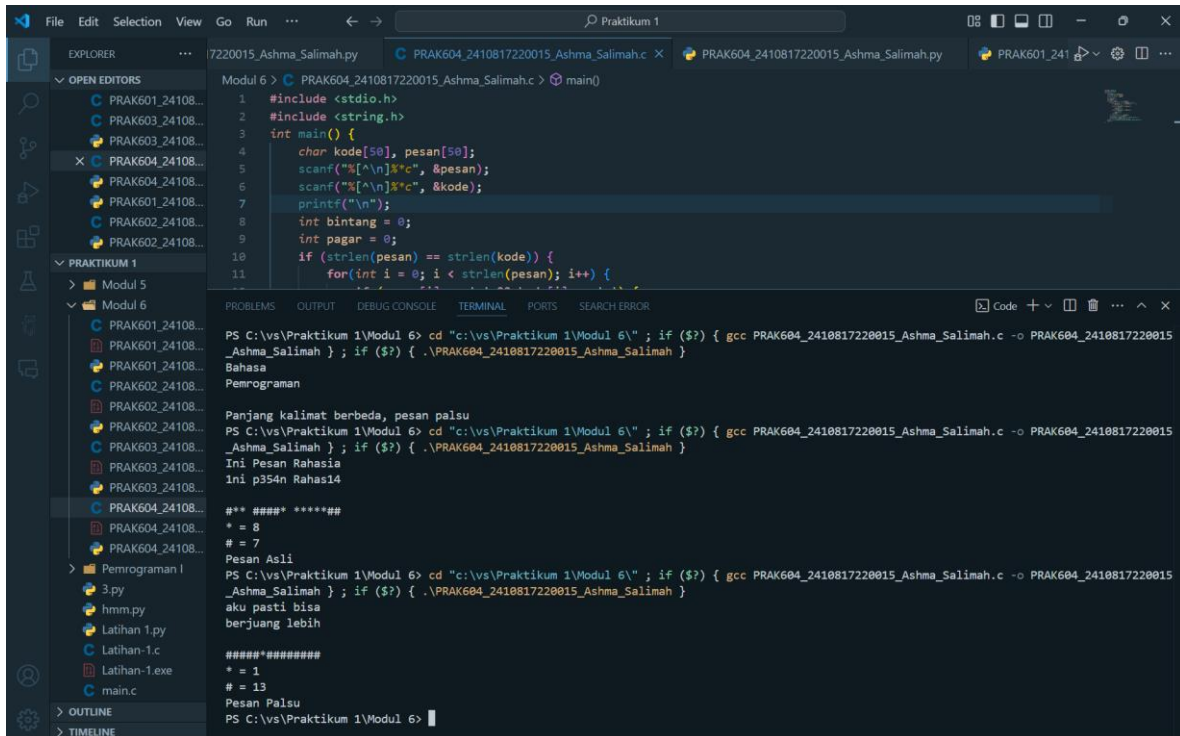
- Python

```
1 pesan = input("")
2 kode = input("")
3 print(" ")
4 karakter = ""
5 bintang = 0
6 pagar = 0
7 if len(pesan) == len(kode) :
8     for i in range (len(pesan)) :
9         if pesan[i] == ' ' and kode[i] == ' ' :
10             karakter += " "
11         elif pesan[i] == kode[i] :
12             karakter += "*"
13             bintang += 1
14         else :
15             karakter += "#"
16             pagar += 1
17     print(karakter)
18     print(f"* = {bintang}")
19     print(f"# = {pagar}")
20     if bintang >= pagar :
21         print("Pesan Asli")
22     else :
23         print("Pesan Palsu")
24 else :
25     print("Panjang kalimat berbeda, pesan palsu")
```

Table 8 Source Code 2 Soal 4

B. Output Program

- C



The screenshot shows the Visual Studio Code interface with a C program open in the editor. The program is named `PRAK604_2410817220015_Ashma_Salimah.c`. The code includes `<stdio.h>` and `<string.h>`. It defines a `main` function that takes a message and a code as input, checks if their lengths are equal, and then prints the message. The output window shows the results of running the program, including the message "Bahasa Pemrograman" and the code "p354n Rahas14".

```
#include <stdio.h>
#include <string.h>
int main() {
    char kode[50], pesan[50];
    scanf("%[^\n]%c", &pesan);
    scanf("%[^\n]%c", &kode);
    printf("\n");
    int bintang = 0;
    int pagar = 0;
    if (strlen(pesan) == strlen(kode)) {
        for(int i = 0; i < strlen(pesan); i++) {
```

PS C:\vs\Praktikum 1\Modul 6> cd "c:\vs\Praktikum 1\Modul 6\" ; if (\$?) { gcc PRAK604_2410817220015_Ashma_Salimah.c -o PRAK604_2410817220015_Ashma_Salimah } ; if (\$?) { .\PRAK604_2410817220015_Ashma_Salimah }

Bahasa
Pemrograman

Panjang kalimat berbeda, pesan palsu
PS C:\vs\Praktikum 1\Modul 6> cd "c:\vs\Praktikum 1\Modul 6\" ; if (\$?) { gcc PRAK604_2410817220015_Ashma_Salimah.c -o PRAK604_2410817220015_Ashma_Salimah } ; if (\$?) { .\PRAK604_2410817220015_Ashma_Salimah }

Ini Pesan Rahasia
Ini p354n Rahas14

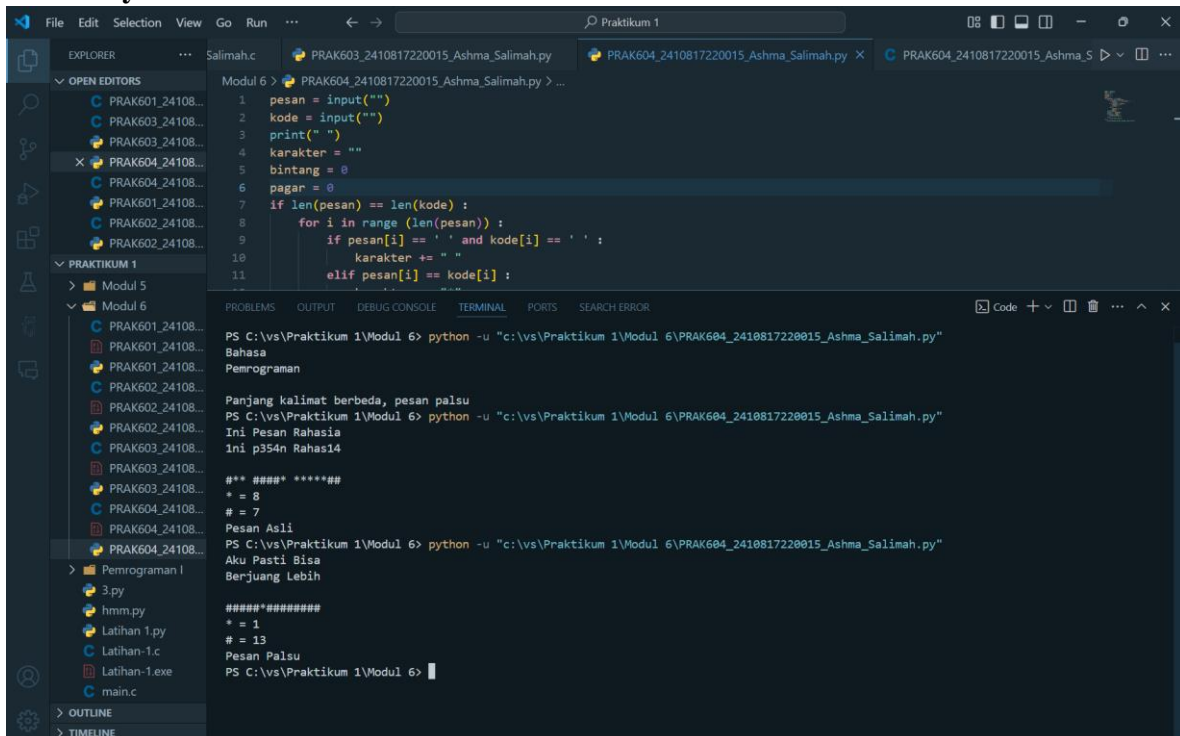
*** #####
* = 8
= 7
Pesan Asli
PS C:\vs\Praktikum 1\Modul 6> cd "c:\vs\Praktikum 1\Modul 6\" ; if (\$?) { gcc PRAK604_2410817220015_Ashma_Salimah.c -o PRAK604_2410817220015_Ashma_Salimah } ; if (\$?) { .\PRAK604_2410817220015_Ashma_Salimah }

aku pasti bisa
berjuang lebih

* = 1
= 13
Pesan Palsu
PS C:\vs\Praktikum 1\Modul 6>

Gambar 7 Screenshot 1 Hasil Jawaban Soal 4

- Python



The screenshot shows the Visual Studio Code interface with a Python program open in the editor. The program is named `PRAK604_2410817220015_Ashma_Salimah.py`. The code takes a message and a code as input, checks if their lengths are equal, and then prints the message. The output window shows the results of running the program, including the message "Bahasa Pemrograman" and the code "p354n Rahas14".

```
pesan = input("")
kode = input("")
print("")
karakter = ""
bintang = 0
pagar = 0
if len(pesan) == len(kode) :
    for i in range (len(pesan)) :
        if pesan[i] == ' ' and kode[i] == ' ' :
            karakter += " "
        elif pesan[i] == kode[i] :
```

PS C:\vs\Praktikum 1\Modul 6> python -u "c:\vs\Praktikum 1\Modul 6\PRAK604_2410817220015_Ashma_Salimah.py"

Bahasa
Pemrograman

Panjang kalimat berbeda, pesan palsu
PS C:\vs\Praktikum 1\Modul 6> python -u "c:\vs\Praktikum 1\Modul 6\PRAK604_2410817220015_Ashma_Salimah.py"

Ini Pesan Rahasia
Ini p354n Rahas14

*** #####
* = 8
= 7
Pesan Asli
PS C:\vs\Praktikum 1\Modul 6> python -u "c:\vs\Praktikum 1\Modul 6\PRAK604_2410817220015_Ashma_Salimah.py"

Aku Pasti Bisa
Berjuang Lebih

* = 1
= 13
Pesan Palsu
PS C:\vs\Praktikum 1\Modul 6>

Gambar 8 Screenshot 2 Hasil Jawaban Soal 4

C. Pembahasan

- C

Pada baris pertama, terdapat `#include <stdio.h>` yang memiliki fungsi menyertakan library standar, juga sebagai kode input maupun output yang akan dilibatkan dalam program.

Pada baris ke-2, terdapat `#include <string.h>` berfungsi sebagai pustaka standar yang menyediakan fungsi string (karakter array) dan memori, berisi berbagai fungsi operasi seperti membandingkan, menyalin, mencari, dan mengubah string dan memori.

`int main() {` berfungsi sebagai penunjuk bahwa `main()` membalikkan nilai integer. Fungsi inilah yang menjadi titik awal dan titik akhir eksekusi program. Tanda kurung kurawal `{ }` berfungsi untuk mendefinisikan blok kode, sebagai pendefinisian tubuh fungsi. `{` berfungsi sebagai awal eksekusi program dan `}` berfungsi sebagai akhir tubuh atau akhir eksekusi program.

```
int bintang = 0;
```

```
int pagar = 0;
```

`int` memiliki fungsi sama, yaitu membalikkan nilai integer tetapi pada variabel `bintang`, `pagar`.

```
char kode[50], pesan[50];
```

tipe data `char` berfungsi untuk menyimpan karakter yang direpresentasikan sebagai nilai ASCII (1 byte per-karakter).

`kode[50]` dan `pesan[50]` adalah array yang dapat menyimpan hingga 50 elemen bertipe `char` pada variabel `kode` dan `pesan`.

```
scanf("%d", &ruang);
```

`scanf` berfungsi untuk membaca string pada input. `%d` berfungsi untuk menerima tipe data integer (bilangan bulat). Tanda `&` sebelum variabel berfungsi untuk memberi alamat memori dari input akan dibaca dan disimpan pada variabel.

```
strlen(pesan)
```

```
strlen(kode)
```

`strlen` berfungsi untuk menghitung panjang string, fungsi ini masuk ke dalam pustaka `<string.h>`.

```
scanf("%[^\n]*c", &pesan);
```

```
scanf("%[^\n]*c", &kode);
```

Pada `scanf("%[^\n]*c", &pesan);`, tanda `%[]` berfungsi untuk menandai input khusus, tanda `^` artinya tidak termasuk, `%*c` berfungsi untuk membaca sebuah

karakter, tetapi tidak menyimpannya karena terdapat tanda *, yang berarti fungsi \n dihapus dalam inputan dan tidak mempengaruhi input berikutnya.

`scanf` berfungsi untuk membaca string pada input. `%d` berfungsi untuk menerima tipe data integer (bilangan bulat). Tanda `&` sebelum variabel adalah operator alamat yang berfungsi untuk memberi alamat memori dari input akan dibaca dan disimpan pada variabel, tanda tersebut dibutuhkan karena fungsi `scanf` membutuhkan alamat memori untuk menempatkan data yang dimasukkan.

`if()`, `else if()`, dan `else` adalah fungsi kondisional yang berguna untuk mengontrol alur program berdasarkan kondisi tertentu.

`if` untuk memeriksa kondisi. Jika kondisi tersebut benar, maka kode di dalamnya akan dieksekusi.

`else if` digunakan untuk memeriksa kondisi tambahan jika kondisi sebelumnya salah.

`else` berfungsi untuk menangani semua kasus yang tidak terpenuhi oleh kondisi yang ada di `if` dan `else if`.

```
for(int i = 0; i < strlen(pesan); i++) {
```

`for` adalah loop yang berfungsi untuk melakukan pengulangan blok kode.

Pada `i = 0`, variabel `i` diinisialisasikan dengan 0, berarti pengulangan dimulai dari 0.

Pada `i < strlen(pesan);`, loop akan terus berjalan selama `i` kurang dari jumlah panjang string dari variabel `baris`, jika telah mencapai variabel tersebut, loop akan berhenti.

Pada `i++`, loop akan meningkat sebanyak satu unit.

```
printf("\n* = %d \n", bintang);
```

```
printf("# = %d \n", pagar);
```

berfungsi untuk menampilkan suatu keluaran atau mencetak output pada layar. `%d` berfungsi untuk menerima tipe data integer (bilangan bulat) pada variabel yang terletak setelah tanda koma.

`\n` berfungsi untuk menambahkan baris baru pada output. Jika tidak ada `\n` sebelum tanda petik maka output akan tetap berada pada baris yang sama walaupun dalam inputan berada pada baris baru.

`;` pada setiap akhir fungsi, berfungsi untuk menandai akhir dari setiap pernyataan atau statement dan bukan sebagai pemisah antara dua pernyataan.

Pada `return 0;`, `return` berfungsi sebagai pembalik nilai dari variabel pada tempat di mana fungsi dipanggil.

- **Python**

```
pesan = input("")
```

```
kode = input("")
```

`input("")` berfungsi untuk meminta pengguna untuk menginput data atau string.

```
len(pesan)
```

```
len(kode)
```

`len` berfungsi untuk menghitung panjang jumlah elemen objek, seperti string, list, tuple, dictionary, dan lainnya.

```
if len(pesan) == len(kode) :
```

```
if pesan[i] == ' ' and kode[i] == ' ' :
```

```
elif pesan[i] == kode[i] :
```

```
if bintang >= pagar :
```

```
else :
```

`if()`, `elif()`, dan `else` adalah fungsi kondisional yang berguna untuk mengontrol alur program berdasarkan kondisi tertentu.

`if` untuk memeriksa kondisi. Jika kondisi tersebut benar, maka kode di dalamnya akan dieksekusi.

`elif` digunakan untuk memeriksa kondisi tambahan jika kondisi sebelumnya salah.

`else` berfungsi untuk menangani semua kasus yang tidak terpenuhi oleh kondisi yang ada di `if` dan `elif`.

```
for i in range (len(pesan)) :
```

Loop yang membuat iterasi dari 0 sampai panjang elemen dari variabel `pesan - 1`.

```
print(f"* = {bintang}")
```

```
print(f"# = {pagar}")
```

berfungsi untuk menampilkan suatu keluaran atau mencetak output pada layar. Tanda `f` adalah fungsi f-string, string literal yang terdapat sebelum tanda petik. Tanda kurung kurawal berfungsi untuk menempatkan variabel ke dalam string, dengan memasukkan variabel fungsi yang telah dibuat sebelumnya.

SOAL 5

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

Format input:

Baris pertama merupakan ordo matriks $n \times n$

Baris kedua merupakan isi matriks A

Baris ketiga merupakan isi matriks B

Format output:

Output merupakan hasil kali dari matriks $A \times B$

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

A. Source Code

• C

```
1  #include <stdio.h>
2  int main() {
3      int ordo;
4      scanf("%d", &ordo);
5      int matriksA[ordo][ordo], matriksB[ordo][ordo], i,
6      j;
7      printf("Matriks A \n");
8      for (i = 0; i < ordo; i++) {
9          for (j = 0; j < ordo; j++) {
10             scanf("%d", &matriksA[i][j]); } }
11      printf("Matriks B \n");
12      for (i = 0; i < ordo; i++) {
13          for (j = 0; j < ordo; j++) {
14             scanf("%d", &matriksB[i][j]); } }
15      int hasil[ordo][ordo];
16      printf("\nMatriks AXB\n");
17      for (i = 0; i < ordo; i++) {
18          for (j = 0; j < ordo; j++) {
19             hasil[i][j] = 0;
20             for (int k = 0; k < ordo; k++) {
21                 hasil[i][j] += matriksA[i][k] *
22                 matriksB[k][j]; } } }
23      for (i = 0; i < ordo; i++) {
24          for (j = 0; j < ordo; j++) {
25             printf("%d ", hasil[i][j]); }
26      printf("\n");}
27      return 0;
28 }
```

Table 9 Source Code 1 Soal 5

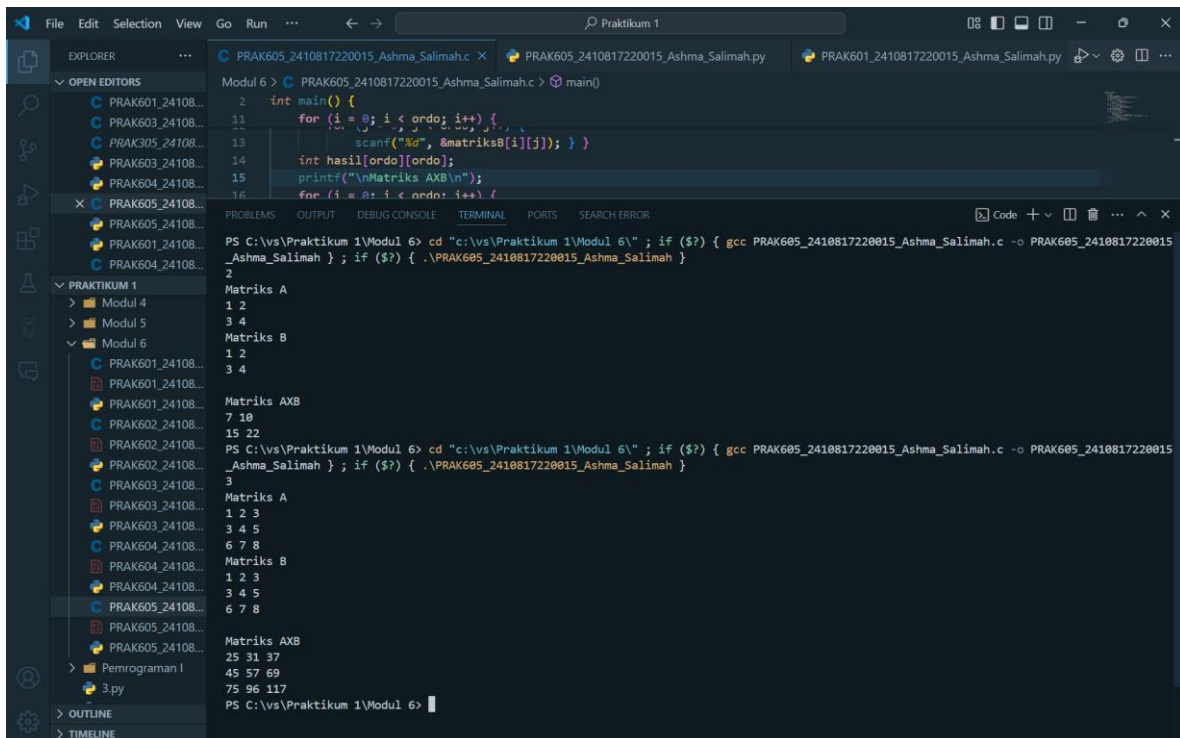
- Python

```
1 ordo = int(input(""))
2 matriksA = [[0] * ordo for _ in range(ordor)]
3 matriksB = [[0] * ordo for _ in range(ordor)]
4 hasil = [[0] * ordo for _ in range(ordor)]
5 print("Matriks A")
6 for i in range (ordo) :
7     matriksA[i] = list(map(int,input(" ").split()))
8 print("Matriks B")
9 for i in range (ordo) :
10     matriksB[i] = list(map(int,input(" ").split()))
11 for i in range (ordo) :
12     for j in range (ordo) :
13         hasil[i][j] = 0
14         for k in range (ordo) :
15             hasil[i][j] += matriksA[i][k] *
matriksB[k][j]
16 print("\nMatriks AXB")
17 for i in range (ordo) :
18     for j in range (ordo) :
19         print(hasil[i][j], end=" ")
20     print("")
```

Table 10 Source Code 2 Soal 5

B. Output Program

- C

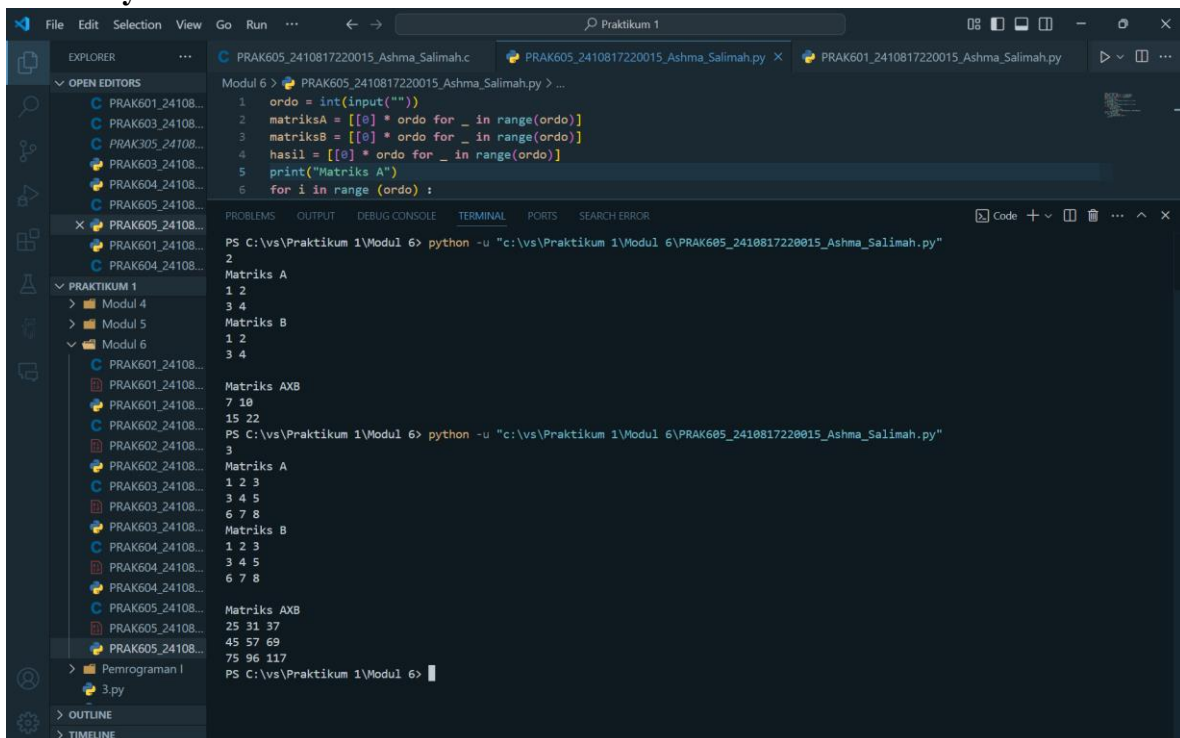


```
Modul 6 > PRAK605_2410817220015_Ashma_Salimah.c > main()
2 int main() {
11 for (i = 0; i < ordo; i++) {
12     scanf("%d", &matriksB[i][j]); } }
13
14 int hasil[ordo][ordo];
15 printf("\nMatriks AXB\n");
16 for (i = 0; i < ordo; i++) {
17     for (j = 0; j < ordo; j++) {
18         hasil[i][j] = 0;
19         for (k = 0; k < ordo; k++) {
20             hasil[i][j] += matriksA[i][k] * matriksB[k][j];
21         }
22     }
23 }
24 printf("\nMatriks AXB\n");
25 for (i = 0; i < ordo; i++) {
26     for (j = 0; j < ordo; j++) {
27         printf("%d\t", hasil[i][j]);
28     }
29     printf("\n");
30 }
31 return 0;
32 }
```

```
PS C:\vs\Praktikum 1\Modul 6> cd "c:\vs\Praktikum 1\Modul 6\" ; if ($?) { gcc PRAK605_2410817220015_Ashma_Salimah.c -o PRAK605_2410817220015_Ashma_Salimah } ; if ($?) { .\PRAK605_2410817220015_Ashma_Salimah }
2
Matriks A
1 2
3 4
Matriks B
1 2
3 4
Matriks AXB
7 10
15 22
PS C:\vs\Praktikum 1\Modul 6> cd "c:\vs\Praktikum 1\Modul 6\" ; if ($?) { gcc PRAK605_2410817220015_Ashma_Salimah.c -o PRAK605_2410817220015_Ashma_Salimah } ; if ($?) { .\PRAK605_2410817220015_Ashma_Salimah }
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
PS C:\vs\Praktikum 1\Modul 6>
```

Gambar 9 Screenshot 1 Hasil Jawaban Soal 5

- Python



```
Modul 6 > PRAK605_2410817220015_Ashma_Salimah.py > ...
1 ordo = int(input(""))
2 matriksA = [[0] * ordo for _ in range(ordo)]
3 matriksB = [[0] * ordo for _ in range(ordo)]
4 hasil = [[0] * ordo for _ in range(ordo)]
5 print("Matriks A")
6 for i in range (ordo) :
7     for j in range (ordo) :
8         matriksA[i][j] = int(input(""))
9     print("\nMatriks B")
10    for k in range (ordo) :
11        for l in range (ordo) :
12            matriksB[k][l] = int(input(""))
13    print("\nMatriks AXB")
14    for i in range (ordo) :
15        for j in range (ordo) :
16            hasil[i][j] = 0
17            for k in range (ordo) :
18                hasil[i][j] += matriksA[i][k] * matriksB[k][j]
19            print(hasil[i][j], end=" ")
20        print("\n")
21    return hasil
22
```

```
PS C:\vs\Praktikum 1\Modul 6> python -u "c:\vs\Praktikum 1\Modul 6\PRAK605_2410817220015_Ashma_Salimah.py"
2
Matriks A
1 2
3 4
Matriks B
1 2
3 4
Matriks AXB
7 10
15 22
PS C:\vs\Praktikum 1\Modul 6> python -u "c:\vs\Praktikum 1\Modul 6\PRAK605_2410817220015_Ashma_Salimah.py"
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
PS C:\vs\Praktikum 1\Modul 6>
```

Gambar 10 Screenshot 2 Hasil Jawaban Soal 5

C. Pembahasan

- C

Pada baris pertama, terdapat `#include <stdio.h>` yang memiliki fungsi menyertakan library standar, juga sebagai kode input maupun output yang akan dilibatkan dalam program.

Pada baris ke-2, terdapat `int main() { .int` berfungsi sebagai penunjuk bahwa `main()` membalikkan nilai integer. Fungsi inilah yang menjadi titik awal dan titik akhir eksekusi program. Tanda kurung kurawal (`{ }`) berfungsi untuk mendefinisikan blok kode, sebagai pendefinisian tubuh fungsi. `{` berfungsi sebagai awal eksekusi program dan `}` berfungsi sebagai akhir tubuh atau akhir eksekusi program.

```
int ordo;
```

```
int matriksA[ordo][ordo], matriksB[ordo][ordo], i, j;
```

```
int hasil[ordo][ordo];
```

`int` memiliki fungsi sama, yaitu membalikkan nilai integer tetapi pada variabel `ordo`, `matriksA`, `matriksB`, `i`, `j`, dan `hasil`.

`[ordo]` adalah indeks baris dan kolom yang ingin digunakan.

```
scanf("%d", &ordo);
```

```
scanf("%d", &matriksA[i][j]); } }
```

```
scanf("%d", &matriksB[i][j]); } }
```

`scanf` berfungsi untuk membaca string pada input. `%d` berfungsi untuk menerima tipe data integer (bilangan bulat). Tanda `&` sebelum variabel adalah operator alamat yang berfungsi untuk memberi alamat memori dari input akan dibaca dan disimpan pada variabel, tanda tersebut dibutuhkan karena fungsi `scanf` membutuhkan alamat memori untuk menempatkan data yang dimasukkan.

```
for (i = 0; i < ordo; i++) {
```

```
    for(j = 0; j < ordo; j++) {
```

`for` adalah loop yang berfungsi untuk melakukan pengulangan blok kode.

Pada `i = 0`, variabel `i` diinisialisasikan dengan 0, berarti pengulangan dimulai dari 0.

Pada `i < ordo;`, loop akan terus berjalan selama `i` kurang dari variabel `ordo`, jika telah mencapai variabel tersebut, loop akan berhenti.

Pada `i++`, loop akan meningkat sebanyak satu unit.

`for(j = 0; j < ordo; j++)` adalah nested loop, yaitu loop yang berjalan di dalam loop. `i` dan `j` mewakili koordinat dalam matriks.

```
hasil[i][j] += matriksA[i][k] * matriksB[k][j]; } } }
```

adalah fungsi untuk menghitung perkalian baris dan kolom matriks, operasi tersebut akan disimpan dalam variabel `hasil[i][j]`

```
printf("\n* = %d \n", hasil[i][j]);
```

berfungsi untuk menampilkan suatu keluaran atau mencetak output pada layar. `%d` berfungsi untuk menerima tipe data integer (bilangan bulat) pada variabel yang terletak setelah tanda koma.

`\n` berfungsi untuk menambahkan baris baru pada output. Jika tidak ada `\n` sebelum tanda petik maka output akan tetap berada pada baris yang sama walaupun dalam inputan berada pada baris baru.

; pada setiap akhir fungsi, berfungsi untuk menandai akhir dari setiap pernyataan atau statement dan bukan sebagai pemisah antara dua pernyataan.

Pada `return 0;`, `return` berfungsi sebagai pembalik nilai dari variabel pada tempat di mana fungsi dipanggil.

- **Python**

```
ordo = int(input(""))
```

```
matriksA[i] = list(map(int,input("").split()))
```

```
matriksB[i] = list(map(int,input("").split()))
```

`list` berfungsi untuk menyimpan kumpulan elemen dalam satu variabel.

`map(int, ...)` mengkonversi elemen dalam string menjadi integer.

`input("")` meminta pengguna untuk menginput data atau string.

`.split()` memisahkan inputan berdasarkan spasi dan menghasilkan daftar string.

```
matriksA = [[0] * ordo for _ in range(ordo)]
```

```
matriksB = [[0] * ordo for _ in range(ordo)]
```

```
hasil = [[0] * ordo for _ in range(ordo)]
```

`[0] * ordo` membuat list baris berisi elemen `ordo` yang semuanya bernilai 0.

Contoh: `ordo = 2` menjadi `[0, 0]`.

```
for i in range (ordo) :
```

```
    for j in range (ordo) :
```

Loop yang membuat iterasi dari 0 sampai `ordo - 1`.

`for j in range (ordo) :` adalah loop yang berjalan dalam loop.

```
hasil[i][j] += matriksA[i][k] * matriksB[k][j]; } } }
```

adalah fungsi untuk menghitung perkalian baris dan kolom matriks, operasi tersebut akan disimpan dalam variabel `hasil[i][j]`.

`print(f"{hasil[i][j]}", end=" ")` berfungsi untuk menampilkan suatu keluaran atau mencetak output pada layar. Tanda `f` adalah fungsi f-string, string literal yang terdapat sebelum tanda petik. Tanda kurung kurawal berfungsi untuk menempatkan variabel ke dalam string, dengan memasukkan variabel fungsi yang telah dibuat sebelumnya.

`end=" ")` memisahkan output berdasarkan spasi.