

LAPORAN AKHIR PRAKTIKUM PEMROGRAMAN I



Oleh:

Nazla Salsabila NIM. 2410817320001

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
2024**

LEMBAR PENGESAHAN

LAPORAN AKHIR PRAKTIKUM PEMROGRAMAN I

Laporan Praktikum Pemrograman I

Modul 0 : How To Program

Modul 1 : Variable, Tipe Data, dan Operator

Modul 2 : Input, dan Output

Modul 3 : Kondisional

Modul 4 : Loop

Modul 5 : Fungsi

Modul 6 : Array

ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman I. Laporan Akhir Praktikum ini dikerjakan oleh:

Nama Praktikan : Nazla Salsabila
NIM : 2410817320001

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Randy Febrian
NIM. 2310817110013

Ir. Eka Setya Wijaya S.T., M.Kom.
NIP. 198205082008011010

DAFTAR ISI

LEMBAR PENGESAHAN	2
DAFTAR ISI.....	3
DAFTAR TABEL.....	9
DAFTAR GAMBAR	13
MODUL 0: HOW TO PROGRAM	17
SOAL 1	17
A. Source Code	17
B. Output Program	17
C. Pembahasan	18
SOAL 2	19
A. Source Code	19
B. Output Program	20
C. Pembahasan	20
SOAL 3.....	22
A. Source Code	22
B. Output Program	22
C. Pembahasan	23
SOAL 4.....	24
A. Source Code	24
B. Output Program	25
C. Pembahasan	25
SOAL 5.....	27
A. Source Code	27
B. Output Program	28
C. Pembahasan	28
MODUL 1: VARIABLE, TIPE DATA, DAN OPERATOR	30
SOAL 1	30
A. Source Code	30

B. Output Program	31
C. Pembahasan	32
SOAL 2	34
A. Source Code	34
B. Output Program	35
C. Pembahasan	36
SOAL 3	38
A. Source Code	38
B. Output Program	39
C. Pembahasan	40
SOAL 4	42
A. Source Code	42
B. Output Program	44
C. Pembahasan	44
SOAL 5	47
A. Source Code	47
B. Output Program	48
C. Pembahasan	49
SOAL 6	51
A. Source Code	51
B. Output Program	53
C. Pembahasan	53
SOAL 7	56
A. Source Code	56
B. Output Program	58
C. Pembahasan	58
SOAL 8	61
A. Source Code	61
B. Output Program	63
C. Pembahasan	63

SOAL 9	66
A. Source Code	66
B. Output Program	67
C. Pembahasan	68
SOAL 10	70
A. Source Code	70
B. Output Program	72
C. Pembahasan	73
MODUL 2: INPUT DAN OUTPUT	75
SOAL 1	75
A. Source Code	76
B. Output Program	78
C. Pembahasan	78
SOAL 2	81
A. Source Code	81
B. Output Program	84
C. Pembahasan	84
SOAL 3	86
A. Source Code	86
B. Output Program	88
C. Pembahasan	89
SOAL 4	91
A. Source Code	91
B. Output Program	93
C. Pembahasan	94
SOAL 5	96
A. Source Code	96
B. Output Program	100
C. Pembahasan	100
MODUL 3: KONDISIONAL	103

SOAL 1	103
A. Source Code	103
B. Output Program	104
C. Pembahasan	105
SOAL 2	108
A. Source Code	108
B. Output Program	110
C. Pembahasan	110
SOAL 3	113
A. Source Code	113
B. Output Program	114
C. Pembahasan	114
SOAL 4	117
A. Source Code	117
B. Output Program	119
C. Pembahasan	119
SOAL 5	122
A. Source Code	122
B. Output Program	124
C. Pembahasan	124
MODUL 4: LOOP	127
SOAL 1	127
A. Source Code	128
B. Output Program	129
C. Pembahasan	129
SOAL 2	133
A. Source Code	133
B. Output Program	135
C. Pembahasan	135
SOAL 3	138

A. Source Code	138
B. Output Program	140
C. Pembahasan	140
SOAL 4	143
A. Source Code	145
B. Output Program	148
C. Pembahasan	148
SOAL 5	152
A. Source Code	152
B. Output Program	154
C. Pembahasan	155
MODUL 5: FUNGSI	158
SOAL 1	158
A. Source Code	159
B. Output Program	160
C. Pembahasan	160
SOAL 2	163
A. Source Code	164
B. Output Program	166
C. Pembahasan	166
SOAL 3	170
A. Source Code	171
B. Output Program	173
C. Pembahasan	173
SOAL 4	176
A. Source Code	177
B. Output Program	178
C. Pembahasan	179
SOAL 5	182
A. Source Code	183

B. Output Program	184
C. Pembahasan	185
MODUL 6: ARRAY	188
SOAL 1	188
A. Source Code	189
B. Output Program	191
C. Pembahasan	191
SOAL 2	195
A. Source Code	195
B. Output Program	197
C. Pembahasan	197
SOAL 3	200
A. Source Code	201
B. Output Program	202
C. Pembahasan	203
SOAL 4	206
A. Source Code	207
B. Output Program	209
C. Pembahasan	209
SOAL 5	213
A. Source Code	214
B. Output Program	216
C. Pembahasan	216
RANGKUMAN	220
TAUTAN GIT	222

DAFTAR TABEL

MODUL 0: HOW TO PROGRAM

Tabel 1. Source Code Jawaban Soal 1 Bahasa C	12
Tabel 2. Source Code Jawaban Soal 1 Bahasa Python	12
Tabel 3. Source Code Jawaban Soal 2 Bahasa C	14
Tabel 4. Source Code Jawaban Soal 2 Bahasa Python	14
Tabel 5. Source Code Jawaban Soal 3 Bahasa C	17
Tabel 6. Source Code Jawaban Soal 3 Bahasa Python	17
Tabel 7. Source Code Jawaban Soal 4 Bahasa C	19
Tabel 8. Source Code Jawaban Soal 4 Bahasa Python	20
Tabel 9. Source Code Jawaban Soal 5 Bahasa C	22
Tabel 10. Source Code Jawaban Soal 5 Bahasa Python	23

MODUL 1: VARIABLE, TIPE DATA, DAN OPERATOR

Tabel 11. Source Code Jawaban Soal 1 Bahasa C	25
Tabel 12. Source Code Jawaban Soal 1 Bahasa Python	26
Tabel 13. Source Code Jawaban Soal 2 Bahasa C	29
Tabel 14. Source Code Jawaban Soal 2 Bahasa Python	30
Tabel 15. Source Code Jawaban Soal 3 Bahasa C	33
Tabel 16. Source Code Jawaban Soal 3 Bahasa Python	34
Tabel 17. Source Code Jawaban Soal 4 Bahasa C	37
Tabel 18. Source Code Jawaban Soal 4 Bahasa Python	38
Tabel 19. Source Code Jawaban Soal 5 Bahasa C	42
Tabel 20. Source Code Jawaban Soal 5 Bahasa Python	43
Tabel 21. Source Code Jawaban Soal 6 Bahasa C	46
Tabel 22. Source Code Jawaban Soal 6 Bahasa Python	47
Tabel 23. Source Code Jawaban Soal 7 Bahasa C	51
Tabel 24. Source Code Jawaban Soal 7 Bahasa Python	52
Tabel 25. Source Code Jawaban Soal 8 Bahasa C	56

Tabel 26. Source Code Jawaban Soal 8 Bahasa Python	57
Tabel 27. Source Code Jawaban Soal 9 Bahasa C	61
Tabel 28. Source Code Jawaban Soal 9 Bahasa Python	62
Tabel 29. Source Code Jawaban Soal 10 Bahasa C	65
Tabel 30. Source Code Jawaban Soal 10 Bahasa Python	66

MODUL 2: INPUT, DAN OUTPUT

Tabel 31. Source Code Jawaban Soal 1 Bahasa C	71
Tabel 32. Source Code Jawaban Soal 1 Bahasa Python	72
Tabel 33. Source Code Jawaban Soal 2 Bahasa C	76
Tabel 34. Source Code Jawaban Soal 2 Bahasa Python	78
Tabel 35. Source Code Jawaban Soal 3 Bahasa C	81
Tabel 36. Source Code Jawaban Soal 3 Bahasa Python	82
Tabel 37. Source Code Jawaban Soal 4 Bahasa C	86
Tabel 38. Source Code Jawaban Soal 4 Bahasa Python	87
Tabel 39. Source Code Jawaban Soal 5 Bahasa C	91
Tabel 40. Source Code Jawaban Soal 5 Bahasa Python	93

MODUL 3: KONDISIONAL

Tabel 41. Source Code Jawaban Soal 1 Bahasa C	98
Tabel 42. Source Code Jawaban Soal 1 Bahasa Python	99
Tabel 43. Source Code Jawaban Soal 2 Bahasa C	103
Tabel 44. Source Code Jawaban Soal 2 Bahasa Python	104
Tabel 45. Source Code Jawaban Soal 3 Bahasa C	108
Tabel 46. Source Code Jawaban Soal 3 Bahasa Python	109
Tabel 47. Source Code Jawaban Soal 4 Bahasa C	112
Tabel 48. Source Code Jawaban Soal 4 Bahasa Python	113
Tabel 49. Source Code Jawaban Soal 5 Bahasa C	117
Tabel 50. Source Code Jawaban Soal 5 Bahasa Python	118

MODUL 4: LOOP

Tabel 51. Source Code Jawaban Soal 1 Bahasa C	123
Tabel 52. Source Code Jawaban Soal 1 Bahasa Python	124
Tabel 53. Source Code Jawaban Soal 2 Bahasa C	128
Tabel 54. Source Code Jawaban Soal 2 Bahasa Python	129
Tabel 55. Source Code Jawaban Soal 3 Bahasa C	133
Tabel 56. Source Code Jawaban Soal 3 Bahasa Python	134
Tabel 57. Source Code Jawaban Soal 4 Bahasa C	140
Tabel 58. Source Code Jawaban Soal 4 Bahasa Python	141
Tabel 59. Source Code Jawaban Soal 5 Bahasa C	147
Tabel 60. Source Code Jawaban Soal 5 Bahasa Python	149

MODUL 5: FUNGSI

Tabel 61. Source Code Jawaban Soal 1 Bahasa C	154
Tabel 62. Source Code Jawaban Soal 1 Bahasa Python	154
Tabel 63. Source Code Jawaban Soal 2 Bahasa C	159
Tabel 64. Source Code Jawaban Soal 2 Bahasa Python	160
Tabel 65. Source Code Jawaban Soal 3 Bahasa C	166
Tabel 66. Source Code Jawaban Soal 3 Bahasa Python	167
Tabel 67. Source Code Jawaban Soal 4 Bahasa C	172
Tabel 68. Source Code Jawaban Soal 4 Bahasa Python	173
Tabel 69. Source Code Jawaban Soal 5 Bahasa C	178
Tabel 70. Source Code Jawaban Soal 5 Bahasa Python	179

MODUL 6: ARRAY

Tabel 71. Source Code Jawaban Soal 1 Bahasa C	189
Tabel 72. Source Code Jawaban Soal 1 Bahasa Python	190
Tabel 73. Source Code Jawaban Soal 2 Bahasa C	195
Tabel 74. Source Code Jawaban Soal 2 Bahasa Python	196

Tabel 75. Source Code Jawaban Soal 3 Bahasa C	201
Tabel 76. Source Code Jawaban Soal 3 Bahasa Python	202
Tabel 77. Source Code Jawaban Soal 4 Bahasa C	207
Tabel 78. Source Code Jawaban Soal 4 Bahasa Python	208
Tabel 79. Source Code Jawaban Soal 5 Bahasa C	214
Tabel 80. Source Code Jawaban Soal 5 Bahasa Python	215

DAFTAR GAMBAR

MODUL 0: HOW TO PROGRAM

Gambar 1. Screenshot Output Soal 1 Bahasa C	12
Gambar 2. Screenshot Output Soal 1 bahasa Python.....	13
Gambar 3. Screenshot Output Soal 2 Bahasa C	15
Gambar 4. Screenshot Output Soal 2 bahasa Python.....	15
Gambar 5. Screenshot Output Soal 3 Bahasa C	17
Gambar 6. Screenshot Output Soal 3 bahasa Python.....	17
Gambar 7. Screenshot Output Soal 4 Bahasa C	20
Gambar 8. Screenshot Output Soal 4 bahasa Python.....	20
Gambar 9. Screenshot Output Soal 5 Bahasa C	23
Gambar 10. Screenshot Output Soal 5 bahasa Python.....	23

MODUL 1: VARIABLE, TIPE DATA, DAN OPERATOR

Gambar 11. Screenshot Output Soal 1 Bahasa C	28
Gambar 12. Screenshot Output Soal 1 bahasa Python.....	28
Gambar 13. Screenshot Output Soal 2 Bahasa C	30
Gambar 14. Screenshot Output Soal 2 bahasa Python.....	30
Gambar 15. Screenshot Output Soal 3 Bahasa C	34
Gambar 16. Screenshot Output Soal 3 bahasa Python.....	35
Gambar 17. Screenshot Output Soal 4 Bahasa C	39
Gambar 18. Screenshot Output Soal 4 bahasa Python.....	39
Gambar 19. Screenshot Output Soal 5 Bahasa C	43
Gambar 20. Screenshot Output Soal 5 bahasa Python.....	43
Gambar 21. Screenshot Output Soal 1 Bahasa C	48
Gambar 22. Screenshot Output Soal 1 bahasa Python.....	48
Gambar 23. Screenshot Output Soal 2 Bahasa C	53
Gambar 24. Screenshot Output Soal 2 bahasa Python.....	53

Gambar 25. Screenshot Output Soal 3 Bahasa C	58
Gambar 26. Screenshot Output Soal 3 bahasa Python.....	58
Gambar 27. Screenshot Output Soal 4 Bahasa C	62
Gambar 28. Screenshot Output Soal 4 bahasa Python.....	63
Gambar 29. Screenshot Output Soal 5 Bahasa C	67
Gambar 30. Screenshot Output Soal 5 bahasa Python.....	68

MODUL 2: INPUT, DAN OUTPUT

Gambar 31. Screenshot Output Soal 1 Bahasa C	73
Gambar 32. Screenshot Output Soal 1 bahasa Python.....	73
Gambar 33. Screenshot Output Soal 2 Bahasa C	79
Gambar 34. Screenshot Output Soal 2 bahasa Python.....	79
Gambar 35. Screenshot Output Soal 3 Bahasa C	83
Gambar 36. Screenshot Output Soal 3 bahasa Python.....	83
Gambar 37. Screenshot Output Soal 4 Bahasa C	88
Gambar 38. Screenshot Output Soal 4 bahasa Python.....	88
Gambar 39. Screenshot Output Soal 5 Bahasa C	95
Gambar 40. Screenshot Output Soal 5 bahasa Python.....	95

MODUL 3: KONDISIONAL

Gambar 41. Screenshot Output Soal 1 Bahasa C	99
Gambar 42. Screenshot Output Soal 1 bahasa Python.....	100
Gambar 43. Screenshot Output Soal 2 Bahasa C	105
Gambar 44. Screenshot Output Soal 2 bahasa Python.....	105
Gambar 45. Screenshot Output Soal 3 Bahasa C	109
Gambar 46. Screenshot Output Soal 3 bahasa Python.....	109
Gambar 47. Screenshot Output Soal 4 Bahasa C	114
Gambar 48. Screenshot Output Soal 4 bahasa Python.....	114
Gambar 49. Screenshot Output Soal 5 Bahasa C	119

Gambar 50. Screenshot Output Soal 5 bahasa Python.....	119
--	-----

MODUL 4: LOOP

Gambar 51. Screenshot Output Soal 1 Bahasa C	124
Gambar 52. Screenshot Output Soal 1 bahasa Python.....	124
Gambar 53. Screenshot Output Soal 2 Bahasa C	130
Gambar 54. Screenshot Output Soal 2 bahasa Python.....	130
Gambar 55. Screenshot Output Soal 3 Bahasa C	135
Gambar 56. Screenshot Output Soal 3 bahasa Python.....	135
Gambar 57. Screenshot Output Soal 4 Bahasa C	143
Gambar 58. Screenshot Output Soal 4 bahasa Python.....	143
Gambar 59. Screenshot Output Soal 5 Bahasa C	149
Gambar 60. Screenshot Output Soal 5 bahasa Python.....	150

MODUL 5: FUNGSI

Gambar 61. Screenshot Output Soal 1 Bahasa C	155
Gambar 62. Screenshot Output Soal 1 bahasa Python.....	155
Gambar 63. Screenshot Output Soal 2 Bahasa C	161
Gambar 64. Screenshot Output Soal 2 bahasa Python.....	161
Gambar 65. Screenshot Output Soal 3 Bahasa C	168
Gambar 66. Screenshot Output Soal 3 bahasa Python.....	168
Gambar 67. Screenshot Output Soal 4 Bahasa C	173
Gambar 68. Screenshot Output Soal 4 bahasa Python.....	173
Gambar 69. Screenshot Output Soal 5 Bahasa C	179
Gambar 70. Screenshot Output Soal 5 bahasa Python.....	179

MODUL 6: ARRAY

Gambar 71. Screenshot Output Soal 1 Bahasa C	191
Gambar 72. Screenshot Output Soal 1 bahasa Python.....	191

Gambar 73. Screenshot Output Soal 2 Bahasa C	197
Gambar 74. Screenshot Output Soal 2 bahasa Python.....	197
Gambar 75. Screenshot Output Soal 3 Bahasa C	202
Gambar 76. Screenshot Output Soal 3 bahasa Python.....	202
Gambar 77. Screenshot Output Soal 4 Bahasa C	209
Gambar 78. Screenshot Output Soal 4 bahasa Python.....	209
Gambar 79. Screenshot Output Soal 5 Bahasa C	216
Gambar 80. Screenshot Output Soal 5 bahasa Python.....	216

MODUL 0: HOW TO PROGRAM

SOAL 1

Buatlah program yang dapat menghasilkan output sebagaimana berikut:

Output
Saya Calon Programmer No. 1

Simpan coding anda dengan nama: **PRAK001-NIM-Nama.py** dan **PRAK001-NIM-Nama.c**

A. Source Code

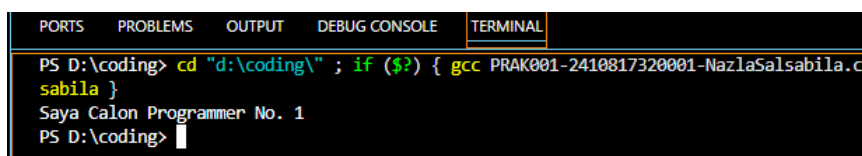
Tabel 1. Source Code Jawaban Soal 1 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	printf("Saya Calon Programmer No.
5	1");
6	return 0;
	}

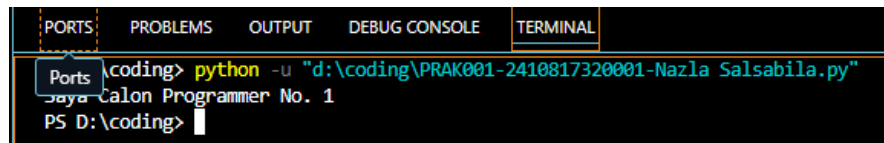
Tabel 2. Source Code Jawaban Soal 1 Bahasa Python

1	print("Saya Calon Programmer No. 1")
---	--------------------------------------

B. Output Program



Gambar 1. Screenshot Output Soal 1 Bahasa C


 A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PORTS', 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is active. The terminal shows a command prompt 'coding>' followed by the command 'python -u "d:\coding\PRAK001-2410817320001-Nazla Salsabila.py"'. Below the command, there is a line of text 'Jaya Calon Programmer No. 1' and another prompt 'PS D:\coding>' with a cursor.

Gambar 2. Screenshot Output Soal 1 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `printf("");`. `Printf` merupakan nama fungsi untuk mencetak output. `""` tanda petik untuk menandai string literal yang akan dicetak, didalam tanda petik dapat memasukkan teks yang ingin ditampilkan. `;` tanda titik koma untuk menandai akhir dari pernyataan.

Pada baris [5], terdapat `return 0;`. `Return` untuk mengembalikan nilai dari fungsi. 0 menunjukkan program telah selesai dengan sukses. `;` titik koma untuk menandai akhir dari pernyataan.

Pada baris [6], terdapat `}`, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `print("")`. `print` berfungsi untuk mencetak output, sedangkan `""` tanda petik untuk menandai string literal yang akan dicetak, di dalam tanda petik dapat memasukkan teks yang ingin ditampilkan.

SOAL 2

Buatlah program yang dapat menghasilkan output sebagaimana berikut:

Output
Selamat Pagi, Nama Anda
Selamat Siang, Nama Anda
Selamat Malam, Nama Anda

Simpan coding anda dengan nama: **PRAK002-NIM-Nama.py** dan **PRAK002-NIM-Nama.c**

A. Source Code

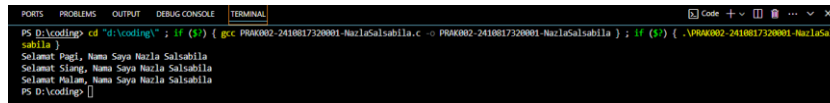
Tabel 3. Source Code Jawaban Soal 2 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	printf("Selamat Pagi, Nama Saya Nazla Salsabila\n");
5	printf("Selamat Siang, Nama Saya Nazla Salsabila\n");
6	printf("Selamat Malam, Nama Saya Nazla Salsabila");
7	return 0;
8	}

Tabel 4. Source Code Jawaban Soal 2 Bahasa Python

1	print("Selamat Pagi, Nama Saya Nazla Salsabila\n")
2	print("Selamat Siang, Nama Saya Nazla Salsabila\n")
3	print("Selamat Malam, Nama Saya Nazla Salsabila")

B. Output Program

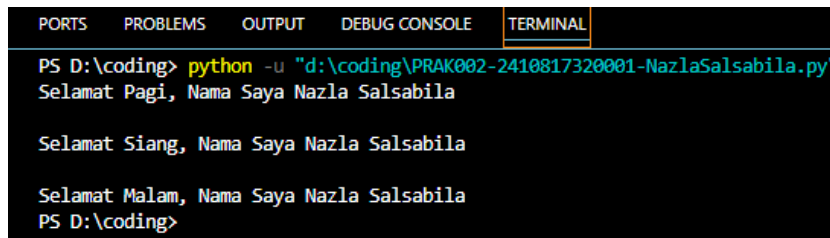


```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK002-2410817320001-NazlaSalsabila.c -o PRAK002-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK002-2410817320001-NazlaSalsabila }
Selamat Pagi, Nama Saya Nazla Salsabila
Selamat Siang, Nama Saya Nazla Salsabila
Selamat Malam, Nama Saya Nazla Salsabila
PS D:\coding>

```

Gambar 3. Screenshot Output Soal 2 Bahasa C



```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> python -u "d:\coding\PRAK002-2410817320001-NazlaSalsabila.py"
Selamat Pagi, Nama Saya Nazla Salsabila

Selamat Siang, Nama Saya Nazla Salsabila

Selamat Malam, Nama Saya Nazla Salsabila
PS D:\coding>

```

Gambar 4. Screenshot Output Soal 2 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [4 – 6], terdapat `printf(“”) ;`. `Printf` merupakan nama fungsi untuk mencetak output. `“”` tanda petik untuk menandai string literal yang akan dicetak, didalam tanda petik dapat memasukkan teks yang ingin ditampilkan. `\n` berfungsi untuk menyertakan karakter baris baru dalam string. `;` tanda titik koma untuk menandai akhir dari pernyataan.

Pada baris [7], terdapat `return 0;`. Return untuk mengembalikan nilai dari fungsi. 0 menunjukkan program telah selesai dengan sukses. `\n` berfungsi untuk menyertakan karakter baris baru dalam string. ; titik koma untuk menandai akhir dari pernyataan.

Pada baris [8], terdapat `}`, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python:

Pada baris [1 – 3], terdapat `Print(“”)`. Print berfungsi untuk mencetak output, sedangkan `“”` tanda petik untuk menandai string literal yang akan dicetak, didalam tanda petik dapat memasukkan teks yang ingin ditampilkan. `\n` berfungsi untuk menyertakan karakter baris baru dalam string.

SOAL 3

Buatlah program yang dapat menghasilkan output sebagaimana berikut:

Output
Andi Berkata "Saya Pasti Bisa"

Simpan coding anda dengan nama: **PRAK003-NIM-Nama.py** dan **PRAK003-NIM-Nama.c**

A. Source Code

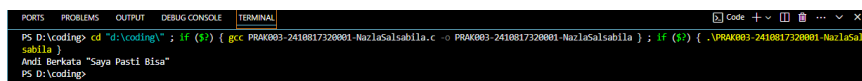
Tabel 5. Source Code Jawaban Soal 3 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	printf("Andi Berkata \"Saya Pasti Bisa\" ");
5	return 0;
6	}

Tabel 6. Source Code Jawaban Soal 3 Bahasa Python

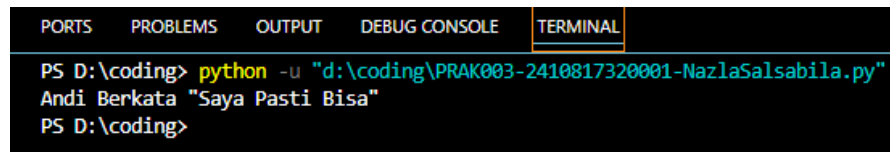
1	print("Andi Berkata \"Saya Pasti Bisa\" ")
---	--

B. Output Program



```
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK003-2410817320001-NazlaSalsabila.c -o PRAK003-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK003-2410817320001-NazlaSalsabila }
Andi Berkata "Saya Pasti Bisa"
PS D:\coding>
```

Gambar 5. Screenshot Output Soal 3 Bahasa C



```
PS D:\coding> python -u "d:\coding\PRAK003-2410817320001-NazlaSalsabila.py"
Andi Berkata "Saya Pasti Bisa"
PS D:\coding>
```

Gambar 6. Screenshot Output Soal 3 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `printf(“”)` ;. `Printf` merupakan nama fungsi untuk mencetak output. `“”` tanda petik untuk menandai string literal yang akan dicetak, didalam tanda petik dapat memasukkan teks yang ingin ditampilkan. `\` berfungsi untuk menambahkan `“` tanda petik pada string. ; tanda titik koma untuk menandai akhir dari pernyataan.

Pada baris [5], terdapat `return 0;`. `Return` untuk mengembalikan nilai dari fungsi. 0 menunjukkan program telah selesai dengan sukses. ; titik koma untuk menandai akhir dari pernyataan.

Pada baris [6], terdapat `}`, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `print(“”)`. `print` berfungsi untuk mencetak output, sedangkan `“”` tanda petik untuk menandai string literal yang akan dicetak, didalam tanda petik dapat memasukkan teks yang ingin ditampilkan. `\` berfungsi untuk menambahkan `“` tanda petik pada string.

SOAL 4

Buatlah program yang dapat menghasilkan output sebagaimana berikut:

Output
#####
#
#
Nama Anda
#
NIM
#
#
#####

Simpan coding anda dengan nama: **PRAK004-NIM-Nama.py** dan **PRAK004-NIM-Nama.c**

A. Source Code

Tabel 7. Source Code Jawaban Soal 4 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	printf("##### \n");
5	printf("# # \n");
6	printf("# Nazla Salsabila # \n");
7	printf("# 2410817320001 # \n");
8	printf("# # \n");
9	printf("##### \n");
10	return 0;
11	}

Tabel 8. Source Code Jawaban Soal 1 Bahasa Python

1	<code>print("##### \n")</code>
2	<code>print("# # \n")</code>
3	<code>print("# Nazla Salsabila # \n")</code>
4	<code>print("# 2410817320001 # \n")</code>
5	<code>print("# # \n")</code>
6	<code>print("##### \n")</code>

B. Output Program

```

PORTS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
[Ports] [coding] cd "d:\coding\" ; if ($?) { gcc PRAK004-2410817320001-NazlaSalsabila.c -o PRAK004-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK004-2410817320001-NazlaSalsabila }
#####
#
# Nazla Salsabila #
# 2410817320001 #
#
#####
PS D:\coding>

```

Gambar 7. Screenshot Output Soal 4 Bahasa C

```

PORTS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\coding> python -u "d:\coding\PRAK004-2410817320001-NazlaSalsabila.py"
#####
#
# Nazla Salsabila #
# 2410817320001 #
#
#####
PS D:\coding>

```

Gambar 8. Screenshot Output Soal 4 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya

printf(). Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. Int menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. Main merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [4 – 9], terdapat `printf(“”) ;`. Printf merupakan nama fungsi untuk mencetak output. `“”` tanda petik untuk menandai string literal yang akan dicetak, didalam tanda petik dapat memasukkan teks yang ingin ditampilkan. `. \n` berfungsi untuk menyertakan karakter baris baru dalam string. `;` tanda titik koma untuk menandai akhir dari pernyataan.

Pada baris [10], terdapat `return 0;`. Return untuk mengembalikan nilai dari fungsi. 0 menunjukkan program telah selesai dengan sukses. `;` titik koma untuk menandai akhir dari pernyataan.

Pada baris [10], terdapat `}`, berfungsi untuk menandai akhir akhir blok kode. `\n` berfungsi untuk menyertakan karakter baris baru dalam string.

Bahasa Python:

Pada baris [1 – 6], terdapat `print(“”)`. print berfungsi untuk mencetak output, sedangkan `“”` tanda petik untuk menandai string literal yang akan dicetak, didalam tanda petik dapat memasukkan teks yang ingin ditampilkan. `\n` berfungsi untuk menyertakan karakter baris baru dalam string.

SOAL 5

Buatlah program yang dapat menampilkan huruf pertama dan terakhir nama anda dengan menggunakan tanda pagar (#). Misalnya, nama saya adalah Nadisheco, maka huruf yang harus dibuat adalah huruf N dan huruf O menggunakan tanda pagar (#). Perhatikan contoh output berikut:

Output

```

# # #####
## # #
### # #
# ## #
# ### #
# ## #
# # #####
-----
Process exited after 0.02385 seconds with return value 125
Press any key to continue . . .

```

Simpan coding anda dengan nama: **PRAK005-NIM-Nama.py** dan **PRAK005-NIM-Nama.c**

A. Source Code

Tabel 9. Source Code Jawaban Soal 5 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	printf("## ## ###\n");
5	printf("### ## ## #\n");
6	printf("#### ## ## #\n");
7	printf("## ### ## #####\n");
8	printf("## #### ## #\n");
9	printf("## ### ## #\n");
10	printf("## ## ## ##");
11	return 0;
12	}

Tabel 10. Source Code Jawaban Soal 5 Bahasa Python

1	<code>print("## ## ###")</code>
2	<code>print("### ## ## ##")</code>
3	<code>print("#### ## ## ##")</code>
4	<code>print("## ### ## #####")</code>
5	<code>print("## #### ## ##")</code>
6	<code>print("## ## ## ##")</code>
7	<code>print("## ## ## ##")</code>

B. Output Program

Gambar 9. Screenshot Output Soal 5 Bahasa C

Gambar 10. Screenshot Output Soal 5 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil

dijalankan. Main merupakan fungsi utama saat program dijalankan. () kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. { kurung kurawal menandai awal blok kode.

Pada baris [4 – 10], terdapat `printf(“”)` ;. `Printf` merupakan nama fungsi untuk mencetak output. “” tanda petik untuk menandai string literal yang akan dicetak, didalam tanda petik dapat memasukkan teks yang ingin ditampilkan. ; tanda titik koma untuk menandai akhir dari pernyataan.

Pada baris [11], terdapat `return 0;`. `Return` untuk mengembalikan nilai dari fungsi. 0 menunjukkan program telah selesai dengan sukses. ; titik koma untuk menandai akhir dari pernyataan.

Pada baris [12], terdapat `}`, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python:

Pada baris [1 – 7], terdapat `print(“”)`. `print` berfungsi untuk mencetak output, sedangkan “” tanda petik untuk menandai string literal yang akan dicetak, didalam tanda petik dapat memasukkan teks yang ingin ditampilkan.

MODUL 1: VARIABLE, TIPE DATA, DAN OPERATOR

SOAL 1

Buatlah program yang dapat menghasilkan output dengan menampilkan variabel yang telah didefinisikan serta menampilkan hasil penjumlahan variabel tersebut dalam bahasa Python dan C.

Output
Variabel x bernilai 5
Variabel y bernilai 7
Variabel z bernilai 9
Jumlah variabel tersebut adalah 21

Simpan coding anda dengan nama: **PRAK101-NIM-Nama.py** dan **PRAK101-NIM-Nama.c**

A. Source Code

Tabel 11. Source Code Jawaban Soal 1 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	
5	int x = 5;
6	int y = 7;
7	int z = 9;
8	
9	printf("Variabel x bernilai %d\n", x);
10	printf("Variabel y bernilai %d\n", y);
11	printf("Variabel z bernilai %d\n", z);
12	
13	int jumlah = x + y + z;

14	<code>printf("Jumlah variabel tersebut adalah %d\n", jumlah);</code>
15	
16	<code>return 0;</code>
17	<code>}</code>

Tabel 12. Source Code Jawaban Soal 1 Bahasa Python

1	<code>x = 5</code>
2	<code>y = 7</code>
3	<code>z = 9</code>
4	
5	<code>print("Variabel x bernilai", x)</code>
6	<code>print("Variabel y bernilai", y)</code>
7	<code>print("Variabel z bernilai", z)</code>
8	
9	<code>jumlah = x + y + z</code>
10	<code>print("Jumlah variabel tersebut adalah", jumlah)</code>

B. Output Program

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK101-2410817320001-NazlaSalsabila.c -o PRAK101-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK101-2410817320001-NazlaSalsabila }
Variabel x bernilai 5
Variabel y bernilai 7
Variabel z bernilai 9
Jumlah variabel tersebut adalah 21
PS D:\coding>

```

Gambar 11. Screenshot Output Soal 1 Bahasa C

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> python -u "d:\coding\PRAK101-2410817320001-NazlaSalsabila.py"
Variabel x bernilai 5
Variabel y bernilai 7
Variabel z bernilai 9
Jumlah variabel tersebut adalah 21
PS D:\coding>

```

Gambar 12. Screenshot Output Soal 1 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [5], `int x = 5;` mendefinisikan variabel `x` dengan tipe data `int` (integer) dan memberikan nilai 5.

Pada baris [6], `int y = 7;` mendefinisikan variabel `y` dengan tipe data `int` (integer) dan memberikan nilai 7.

Pada baris [7], `int z = 9;` mendefinisikan variabel `z` dengan tipe data `int` (integer) dan memberikan nilai 9.

Pada baris [9 – 11], fungsi `printf` digunakan untuk menampilkan teks dan nilai variabel ke layar. String yang ingin ditampilkan ditulis di dalam tanda kutip ganda `" "`. `%d` adalah placeholder untuk menampilkan nilai integer dari variabel. Nilai variabel `x`, `y`, dan `z` ditempatkan setelah tanda koma dan masing-masing akan menggantikan `%d` pada posisi tersebut. `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [13], Variabel `jumlah` didefinisikan dengan hasil penjumlahan dari `x`, `y`, dan `z`. Operasi `x + y + z` menjumlahkan ketiga nilai dan hasilnya disimpan ke dalam variabel `jumlah`.

Pada baris [14], `printf` digunakan untuk menampilkan hasil penjumlahan variabel yang disimpan dalam `jumlah`. `%d` akan digantikan dengan nilai integer dari `jumlah`. `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [16], `return 0`; mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [17], terdapat `}`, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python:

Pada baris [1], `x = 5`, mendefinisikan variabel `x` dengan memberikan nilai 5.

Pada baris [2], `y = 7`, mendefinisikan variabel `y` dengan memberikan nilai 7.

Pada baris [3], `z = 9`, mendefinisikan variabel `z` dengan memberikan nilai 9.

Pada baris [5 – 7], fungsi `print()` digunakan untuk menampilkan teks dan nilai dari variabel. Nilai variabel `x`, `y`, dan `z` ditempatkan setelah tanda koma.

Pada baris [9], `jumlah = x + y + z`, operasi penjumlahan dilakukan dengan menjumlahkan nilai dari variabel `x`, `y`, dan `z`, hasil penjumlahan disimpan dalam variabel baru yang disebut `jumlah`.

Pada baris [10], fungsi `print()` kembali digunakan untuk menampilkan hasil penjumlahan yang telah disimpan dalam variabel `jumlah`.

SOAL 2

Buatlah program yang dapat menghasilkan output berupa hasil kombinasi perkalian dan pembagian dari variabel yang telah didefinisikan menggunakan bahasa Python dan bahasa C.

Output

Variabel x bernilai 4

Variabel y bernilai 8

Variabel z bernilai 3

Hasil dari a dikali b dibagi c adalah 10.666667

Simpan coding anda dengan nama: **PRAK102-NIM-Nama.py** dan **PRAK102-NIM-Nama.c**

A. Source Code

Tabel 13. Source Code Jawaban Soal 2 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	int a = 4;
5	int b = 8;
6	int c = 3;
7	
8	printf("Variabel a bernilai %d\n", a);
9	printf("Variabel b bernilai %d\n", b);
10	printf("Variabel c bernilai %d\n", c);
11	
12	float hasil = (float)(a * b) / c;
13	
14	printf("Hasil dari a dikali b dibagi c adalah %.6f\n", hasil);

15	
16	return 0;
17	}

Tabel 14. Source Code Jawaban Soal 2 Bahasa Python

1	a = 4
2	b = 8
3	c = 3
4	
5	print("Variabel a bernilai", a)
6	print("Variabel b bernilai", b)
7	print("Variabel c bernilai", c)
8	
9	hasil = (a * b) / c
10	
11	print("Hasil dari a dikali b dibagi c adalah", format(hasil, '.6f'))

B. Output Program

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\ASUS\Desktop\coding> cd "d:\coding\" ; if ($?) { gcc PRAK102-2410817320001-NazlaSalsabila.c -o PRAK1
02-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK102-2410817320001-NazlaSalsabila }
Variabel a bernilai 4
Variabel b bernilai 8
Variabel c bernilai 3
Hasil dari a dikali b dibagi c adalah 10.666667
PS D:\coding>

```

Gambar 13. Screenshot Output Soal 2 Bahasa C

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> python -u "d:\coding\tempCodeRunnerFile.py"
Variabel a bernilai 4
Variabel b bernilai 8
Variabel c bernilai 3
Hasil dari a dikali b dibagi c adalah 10.666667
PS D:\coding>

```

Gambar 14. Screenshot Output Soal 2 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [4], `int a = 4;` mendefinisikan variabel `x` dengan tipe data `int` (integer) dan memberikan nilai 4.

Pada baris [5], `int b = 8;` mendefinisikan variabel `y` dengan tipe data `int` (integer) dan memberikan nilai 8.

Pada baris [6], `int c = 3;` mendefinisikan variabel `z` dengan tipe data `int` (integer) dan memberikan nilai 3.

Pada baris [8 – 10], fungsi `printf` digunakan untuk menampilkan teks dan nilai variabel ke layar. String yang ingin ditampilkan ditulis di dalam tanda kutip ganda `" "`. `%d` adalah placeholder untuk menampilkan nilai integer dari variabel. Nilai variabel `a`, `b`, dan `c` ditempatkan setelah tanda koma dan masing-masing akan menggantikan `%d` pada posisi tersebut. `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [12], hasil perkalian antara `a` dan `b` dibagi dengan `c`. Agar hasilnya memiliki presisi yang lebih tinggi (dalam bentuk bilangan desimal), perkalian (`a * b`) terlebih dahulu dikonversi ke tipe `float` menggunakan (`float`) sebelum pembagian. Ini mencegah pembagian integer yang hanya menghasilkan hasil bulat.

Pada baris [14], `printf` digunakan untuk menampilkan hasil perkalian dan pembagian variabel yang disimpan dalam `jumlah`. `%f` adalah format spesifikator untuk

menampilkan angka bertipe float. `.6` menunjukkan bahwa hasil akan ditampilkan dengan 6 angka di belakang koma. `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [16], `return 0`; mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [17], terdapat `}`, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python:

Pada baris [1], `a = 4`, mendefinisikan variabel `x` dengan memberikan nilai 4.

Pada baris [2], `b = 8`, mendefinisikan variabel `y` dengan memberikan nilai 8.

Pada baris [3], `c = 3`, mendefinisikan variabel `z` dengan memberikan nilai 3.

Pada baris [5 – 7], fungsi `print()` digunakan untuk menampilkan teks dan nilai dari variabel. Nilai variabel `a`, `b`, dan `c` ditempatkan setelah tanda koma.

Pada baris [9], hasil perkalian antara `a` dan `b` dibagi dengan `c`, agar hasilnya memiliki presisi yang lebih tinggi (dalam bentuk bilangan desimal), perkalian (`a * b`) terlebih dahulu sebelum pembagian.

Pada baris [11], fungsi `print()` kembali digunakan untuk menampilkan hasil penjumlahan yang telah disimpan dalam variabel `jumlah`. `format(hasil, '.6f')` digunakan untuk menampilkan hasil pembagian dengan 6 angka di belakang koma. Di sini, `.6f` menandakan bahwa angka akan ditampilkan dalam bentuk desimal dengan 6 digit setelah titik.

SOAL 3

Buatlah program yang dapat menghasilkan output berupa hasil kombinasi pertambahan, pengurangan, perkalian dan pembagian dari variabel yang telah didefinisikan menggunakan bahasa Python dan bahasa C.

Output
Variabel a bernilai 9
Variabel b bernilai 6
Variabel x bernilai 10
Variabel y bernilai 7
Hasil dari a ditambah b dikali x dan dibagi y adalah 21.43

Simpan coding anda dengan nama: **PRAK103-NIM-Nama.py** dan **PRAK103-NIM-Nama.c**

A. Source Code

Tabel 15. Source Code Jawaban Soal 3 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	int a = 9;
5	int b = 6;
6	int x = 10;
7	int y = 7;
8	
9	printf("Variabel a bernilai %d\n",
10	a);
11	printf("Variabel b bernilai %d\n",
12	b);
13	printf("Variabel x bernilai %d\n",
14	x);
15	printf("Variabel y bernilai %d\n",
	y);
	float hasil = ((a + b) * (float)x) /
	y;

16	printf("Hasil dari a ditambah b dikali x dan dibagi y adalah %.2f\n", hasil);
17	
18	return 0;
19	}

Tabel 16. Source Code Jawaban Soal 3 Bahasa Python

1	a = 9
2	b = 6
3	x = 10
4	y = 7
5	
6	print("Variabel a bernilai", a)
7	print("Variabel b bernilai", b)
8	print("Variabel x bernilai", x)
9	print("Variabel y bernilai", y)
10	
11	hasil = ((a + b) * x) / y
12	
13	print("Hasil dari a ditambah b dikali x dan dibagi y adalah", format(hasil, '%.2f'))

B. Output Program

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\ASUS\Desktop\coding> cd "d:\coding\" ; if ($?) { gcc PRAK103-2410817320001-NazlaSalsabila.c -o PRAK1
03-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK103-2410817320001-NazlaSalsabila }
Variabel a bernilai 9
Variabel b bernilai 6
Variabel x bernilai 10
Variabel y bernilai 7
Hasil dari a ditambah b dikali x dan dibagi y adalah 21.43
PS D:\coding>

```

Gambar 15. Screenshot Output Soal 3 Bahasa C

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> python -u "d:\coding\tempCodeRunnerFile.py"
Variabel a bernilai 9
Variabel b bernilai 6
Variabel x bernilai 10
Variabel y bernilai 7
Hasil dari a ditambah b dikali x dan dibagi y adalah 21.43
PS D:\coding>

```

Gambar 16. Screenshot Output Soal 3 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [4], `int a = 9;` mendefinisikan variabel `x` dengan tipe data `int` (integer) dan memberikan nilai awal 9.

Pada baris [5], `int b = 6;` mendefinisikan variabel `y` dengan tipe data `int` (integer) dan memberikan nilai awal 6.

Pada baris [6], `int x = 10;` mendefinisikan variabel `z` dengan tipe data `int` (integer) dan memberikan nilai awal 10.

Pada baris [7], `int y = 7;` mendefinisikan variabel `z` dengan tipe data `int` (integer) dan memberikan nilai awal 7.

Pada baris [9 – 12], fungsi `printf` digunakan untuk menampilkan teks dan nilai variabel ke layar. String yang ingin ditampilkan ditulis di dalam tanda kutip ganda `" "`.

%d adalah placeholder untuk menampilkan nilai integer dari variabel. Nilai variabel a, b, x, dan y ditempatkan setelah tanda koma dan masing-masing akan menggantikan %d pada posisi tersebut. \n digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [14], operasi $(a + b)$ dijumlahkan terlebih dahulu, kemudian hasil penjumlahan tersebut dikalikan dengan x. (float)x digunakan untuk mengonversi x menjadi tipe float agar operasi pembagian nantinya menghasilkan bilangan desimal. Hasil kali tersebut dibagi dengan y.

Pada baris [16], printf digunakan untuk menampilkan hasil perkalian dan pembagian variabel yang disimpan dalam jumlah. %f digunakan untuk menampilkan nilai bertipe float. .2 menunjukkan bahwa hasil akan ditampilkan dengan 2 angka di belakang koma. \n digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [18], return 0; mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [19], terdapat }, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python:

Pada baris [1], $a = 9$, mendefinisikan variabel x dengan memberikan nilai 9.

Pada baris [2], $b = 6$, mendefinisikan variabel y dengan memberikan nilai 6.

Pada baris [3], $x = 10$, mendefinisikan variabel z dengan memberikan nilai 10.

Pada baris [4], $y = 7$, mendefinisikan variabel z dengan memberikan nilai 7.

Pada baris [6 – 9], fungsi print() digunakan untuk menampilkan teks dan nilai dari variabel. Nilai variabel a, b, dan c ditempatkan setelah tanda koma.

Pada baris [11], menjumlahkan nilai a dan b, yaitu $9 + 6 = 15$. Hasil penjumlahan tersebut dikalikan dengan x (10), menghasilkan 150. Kemudian, hasil tersebut dibagi dengan nilai y (7), menghasilkan 21.428571.

Pada baris [13], printf digunakan untuk menampilkan hasil perkalian dan pembagian variabel yang disimpan dalam jumlah. Fungsi format(hasil, '.2f') digunakan untuk menampilkan hasil pembagian dengan 2 angka di belakang koma, .2f artinya dua angka desimal).

SOAL 4

Buatlah program yang dapat menghasilkan output berupa hasil diskon dari sebuah harga dengan mengimplementasikan operator menggunakan bahasa Python dan bahasa C.

Output
Harga sepatu A adalah 400000
Harga sepatu B adalah 350000
Sepatu A mendapat diskon 13% sehingga harganya menjadi 348000
Sepatu B mendapat diskon 21% sehingga harganya menjadi 276500

Simpan coding anda dengan nama: **PRAK104-NIM-Nama.py** dan **PRAK104-NIM-Nama.c**

A. Source Code

Tabel 17. Source Code Jawaban Soal 4 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	int harga_sepatu_A = 400000;
5	int harga_sepatu_B = 350000;
6	
7	printf("Harga sepatu A adalah %d\n", harga_sepatu_A);
8	printf("Harga sepatu B adalah %d\n", harga_sepatu_B);
9	
10	float diskon_A_13 = harga_sepatu_A * 13 / 100.0;
11	int harga_setelah_diskon_A_13 = harga_sepatu_A - diskon_A_13;
12	

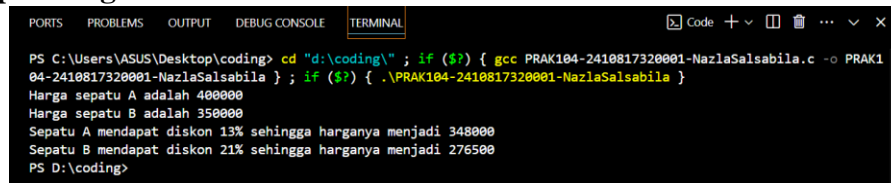
13	float diskon_B_21 = harga_sepatu_B * 21 / 100.0;
14	int harga_setelah_diskon_B_21 = harga_sepatu_B - diskon_B_21;
15	
16	printf("Sepatu A mendapat diskon 13% sehingga harganya menjadi %d\n", harga_setelah_diskon_A_13);
17	printf("Sepatu B mendapat diskon 21% sehingga harganya menjadi %d\n", harga_setelah_diskon_B_21);
18	
19	return 0;
20	}

Tabel 18. Source Code Jawaban Soal 4 Bahasa Python

1	harga_sepatu_A = 400000
2	harga_sepatu_B = 350000
3	
4	print("Harga sepatu A adalah", harga_sepatu_A)
5	print("Harga sepatu B adalah", harga_sepatu_B)
6	
7	diskon_A_13 = harga_sepatu_A * 13 / 100
8	harga_setelah_diskon_A_13 = harga_sepatu_A - diskon_A_13
9	
10	diskon_B_21 = harga_sepatu_B * 21 / 100
11	harga_setelah_diskon_B_21 = harga_sepatu_B - diskon_B_21
12	
13	print("Sepatu A mendapat diskon 13% sehingga harganya menjadi", int(harga_setelah_diskon_A_13))

14	<code>print("Sepatu B mendapat diskon 21% sehingga harganya menjadi", int(harga_setelah_diskon_B_21))</code>
----	--

B. Output Program

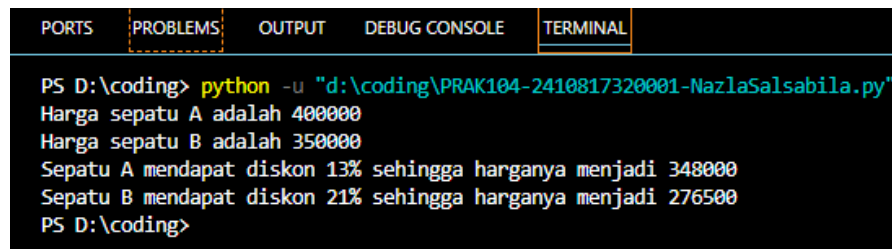


```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\ASUS\Desktop\coding> cd "d:\coding\" ; if ($?) { gcc PRAK104-2410817320001-NazlaSalsabila.c -o PRAK104-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK104-2410817320001-NazlaSalsabila }
Harga sepatu A adalah 400000
Harga sepatu B adalah 350000
Sepatu A mendapat diskon 13% sehingga harganya menjadi 348000
Sepatu B mendapat diskon 21% sehingga harganya menjadi 276500
PS D:\coding>

```

Gambar 17. Screenshot Output Soal 4 Bahasa C



```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> python -u "d:\coding\PRAK104-2410817320001-NazlaSalsabila.py"
Harga sepatu A adalah 400000
Harga sepatu B adalah 350000
Sepatu A mendapat diskon 13% sehingga harganya menjadi 348000
Sepatu B mendapat diskon 21% sehingga harganya menjadi 276500
PS D:\coding>

```

Gambar 18. Screenshot Output Soal 4 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris 1, terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris 3, terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris 5, `int harga sepatu A= 400000;` mendefinisikan variabel `x` dengan tipe data `int` (integer) dan memberikan nilai harga 400000.

Pada baris 6, `int harga sepatu B = 350000;` mendefinisikan variabel `y` dengan tipe data `int` (integer) dan memberikan nilai harga 350000.

Pada baris 9 dan 10, `printf` digunakan untuk menampilkan nilai harga masing-masing sepatu. `%d` akan digantikan dengan nilai integer dari jumlah. `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris 18, diskon sepatu A dihitung dengan mengalikan `harga_sepatu_A` dengan 13% (`13 / 100.0`). Penggunaan `100.0` bertujuan agar hasil pembagian berupa angka desimal (float). Setelah diskon dihitung, harga setelah diskon untuk sepatu A disimpan dalam variabel `harga_setelah_diskon_A_13`.

Pada baris 16, Diskon sepatu B dihitung dengan cara yang sama, yakni mengalikan harga sepatu B dengan 21%. Hasil akhirnya disimpan dalam `harga_setelah_diskon_B_21`.

Pada baris 19 dan 20, `printf()` digunakan untuk menampilkan harga setelah diskon. Untuk menampilkan simbol `%` dalam `printf()`, harus menggunakan `%%`. `%d` akan digantikan dengan nilai integer. `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris 22, `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris 23, terdapat `}`, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python:

Pada baris 2, `harga_sepatu_A = 400000`, mendefinisikan variabel `x` dengan memberikan nilai harga 400000.

Pada baris 3, `harga_sepatu_B = 350000`, mendefinisikan variabel `y` dengan memberikan nilai harga 350000.

Pada baris 6 dan 7, fungsi `print()` digunakan untuk menampilkan harga sepatu sebelum diskon. Harga sepatu sebelum diskon ditempatkan setelah tanda koma.

Pada baris 10, diskon dihitung dengan mengalikan harga sepatu A dengan 13%, lalu dikurangi dari harga awal untuk mendapatkan harga setelah diskon.

Pada baris 14, diskon dihitung dengan mengalikan harga sepatu `B` dengan 21%, lalu dikurangi dari harga awal untuk mendapatkan harga setelah diskon.

Pada baris 18 dan 19, fungsi `print()` digunakan untuk menampilkan harga sepatu setelah diskon. `int()` digunakan untuk membulatkan harga setelah diskon menjadi bilangan bulat.

SOAL 5

Buatlah program yang dapat menghasilkan output berupa total sisa bagi dari pembagian dengan mengimplementasikan operator menggunakan bahasa Python dan bahasa C.

Output

Variabel a bernilai 9

Variabel b bernilai 5

Variabel x bernilai 8

Variabel y bernilai 8

Total sisa bagi dari a dibagi b dan x dibagi y adalah 4

Simpan coding anda dengan nama: **PRAK105-NIM-Nama.py** dan **PRAK105-NIM-Nama.c**

A. Source Code

Tabel 19. Source Code Jawaban Soal 5 Bahasa C

```
1  #include <stdio.h>
2
3  int main() {
4      int a = 9;
5      int b = 5;
6      int x = 8;
7      int y = 8;
8
9      int sisa_a_b = a % b;
10     int sisa_x_y = x % y;
11
12     int total_sisa_bagi = sisa_a_b +
13     sisa_x_y;
14
15     printf("Variabel a bernilai %d \n", a);
16     printf("Variabel b bernilai %d \n", b);
17     printf("Variabel x bernilai %d \n", x);
18     printf("Variabel y bernilai %d \n", y);
```

18	printf("Total sisa bagi dari a dibagi b dan x dibagi y adalah %d\n",
19	total_sisa_bagi);
20	return 0;
21	}

Tabel 20. Source Code Jawaban Soal 5 Bahasa Python

1	a = 9
2	b = 5
3	x = 8
4	y = 8
5	
6	print("Variabel a bernilai", a)
7	print("Variabel b bernilai", b)
8	print("Variabel x bernilai", x)
9	print("Variabel y bernilai", y)
10	
11	sisa_a_b = a % b
12	sisa_x_y = x % y
13	
14	total_sisa_bagi = sisa_a_b + sisa_x_y
15	
16	print("Total sisa bagi dari a dibagi b dan x dibagi y adalah", total_sisa_bagi)

B. Output Program

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\ASUS\Desktop\coding> cd "d:\coding\" ; if ($?) { gcc PRAK105-2410817320001-NazlaSalsabila.c -o PRAK105-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK105-2410817320001-NazlaSalsabila }
Variabel a bernilai 9
Variabel b bernilai 5
Variabel x bernilai 8
Variabel y bernilai 8
Total sisa bagi dari a dibagi b dan x dibagi y adalah 4
PS D:\coding>

```

Gambar 19. Screenshot Output Soal 5 Bahasa C

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> python -u "d:\coding\PRAK105-2410817320001-NazlaSalsabila.py"
Variabel a bernilai 9
Variabel b bernilai 5
Variabel x bernilai 8
Variabel y bernilai 8
Total sisa bagi dari a dibagi b dan x dibagi y adalah 4
PS D:\coding>

```

Gambar 20. Screenshot Output Soal 5 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [4], `int a = 9;` mendefinisikan variabel `x` dengan tipe data `int` (integer) dan memberikan nilai 9.

Pada baris [5], `int b = 5;` mendefinisikan variabel `y` dengan tipe data `int` (integer) dan memberikan nilai 5.

Pada baris [6], `int x = 8;` mendefinisikan variabel `z` dengan tipe data `int` (integer) dan memberikan nilai 8.

Pada baris [7], `int y = 8;` mendefinisikan variabel `z` dengan tipe data `int` (integer) dan memberikan nilai 8.

Pada baris Pada [9], `sis_a_b`: Menghitung sisa bagi dari `a` dibagi `b`. Hasilnya adalah $9 \% 5 = 4$ (karena 9 dibagi 5 menghasilkan 1 dengan sisa 4).

Pada baris [10], `sis_x_y`: Menghitung sisa bagi dari `x` dibagi `y`, yaitu $8 \% 8 = 0$ (karena 8 dibagi 8 tepat 1 kali tanpa sisa).

Pada baris [12], `total_sisa_bagi` dihitung dengan menjumlahkan `sis_a_b` dan `sis_x_y`. Hasilnya adalah $4 + 0 = 4$.

Pada baris [14 – 18], fungsi `printf()` digunakan untuk menampilkan nilai variabel `a`, `b`, `x`, `y`, dan `total_sisa_bagi` ke layar. String yang ingin ditampilkan ditulis di dalam tanda kutip ganda `" "`. `%d` adalah placeholder untuk menampilkan nilai integer dari

variabel. Nilai variable a, b, x, dan y ditempatkan setelah tanda koma dan masing-masing akan menggantikan %d pada posisi tersebut. \n digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [20], return 0; mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [21], terdapat }, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python:

Pada baris [1], a = 9; mendefinisikan variabel x dengan tipe data int (integer) dan memberikan nilai 9.

Pada baris [2], b = 5; mendefinisikan variabel y dengan tipe data int (integer) dan memberikan nilai 5.

Pada baris [3], x = 8; mendefinisikan variabel z dengan tipe data int (integer) dan memberikan nilai 8.

Pada baris [4], y = 8; mendefinisikan variabel z dengan tipe data int (integer) dan memberikan nilai 8.

Pada baris [6 – 9], fungsi print() digunakan untuk menampilkan teks dan nilai dari variabel. Nilai variabel a, b, x, dan y ditempatkan setelah tanda koma.

Pada baris [11], sisa_a_b: Menghitung sisa bagi dari a dibagi b menggunakan operator modulo (%). Hasilnya adalah $9 \% 5 = 4$ (karena 9 dibagi 5 menghasilkan 1 dengan sisa 4).

Pada baris [12], sisa_x_y: Menghitung sisa bagi dari x dibagi y, yaitu $8 \% 8 = 0$ (karena 8 dibagi 8 tepat 1 kali tanpa sisa).

Pada baris [14], total_sisa_bagi dihitung dengan menjumlahkan sisa_a_b dan sisa_x_y. Hasilnya adalah $4 + 0 = 4$.

Pada baris [16], menampilkan hasil total sisa bagi dengan fungsi print("Total sisa bagi dari a dibagi b dan x dibagi y adalah", total_sisa_bagi). Ini mencetak pesan di layar yang menyatakan total sisa bagi tersebut, yaitu 4.

SOAL 6

Buatlah program yang dapat menghasilkan output berupa hasil pengecekan antara 2 nilai

Output
Variabel a bernilai 4
Variabel b bernilai 8
Variabel c bernilai 3
Apakah a sama dengan b ? jawabannya adalah 0
Apakah b lebih besar dari c ? jawabannya adalah 1
Apakah a tidak sama dengan c ? jawabannya adalah 1

Simpan coding anda dengan nama: **PRAK106-NIM-Nama.py** dan **PRAK106-NIM-Nama.c**

A. Source Code

Tabel 21. Source Code Jawaban Soal 6 Bahasa C

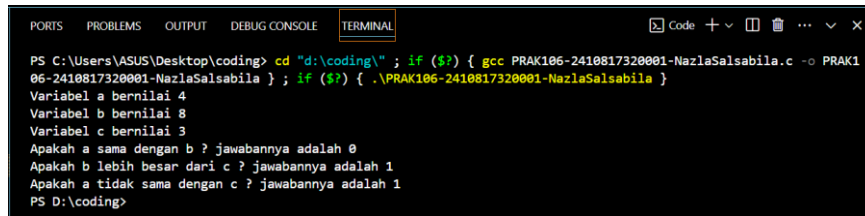
1	#include <stdio.h>
2	
3	int main() {
4	int a = 4;
5	int b = 8;
6	int c = 3;
7	
8	printf("Variabel a bernilai %d\n", a);
9	printf("Variabel b bernilai %d\n", b);
10	printf("Variabel c bernilai %d\n", c);
11	
12	int hasil_sama = (a == b);
13	
14	int hasil_lebih_besar = (b > c);

15	
16	<code>int hasil_tidak_sama = (a != c);</code>
17	
18	<code>printf("Apakah a sama dengan b ? jawabannya adalah %d\n", hasil_sama);</code>
19	<code>printf("Apakah b lebih besar dari c ? jawabannya adalah %d\n", hasil_lebih_besar);</code>
20	<code>printf("Apakah a tidak sama dengan c ? jawabannya adalah %d\n", hasil_tidak_sama);</code>
21	
22	<code>return 0;</code>
23	<code>}</code>

Tabel 22. Source Code Jawaban Soal 6 Bahasa Python

1	<code>a = 4</code>
2	<code>b = 8</code>
3	<code>c = 3</code>
4	
5	<code>print("Variabel a bernilai", a)</code>
6	<code>print("Variabel b bernilai", b)</code>
7	<code>print("Variabel c bernilai", c)</code>
8	
9	<code>hasil_sama = a == b</code>
10	
11	<code>hasil_lebih_besar = b > c</code>
12	
13	<code>hasil_tidak_sama = a != c</code>
14	
15	<code>print("Apakah a sama dengan b ? jawabannya adalah", int(hasil_sama))</code>
16	<code>print("Apakah b lebih besar dari c ? jawabannya adalah", int(hasil_lebih_besar))</code>
17	<code>print("Apakah a tidak sama dengan c ? jawabannya adalah", int(hasil_tidak_sama))</code>

B. Output Program

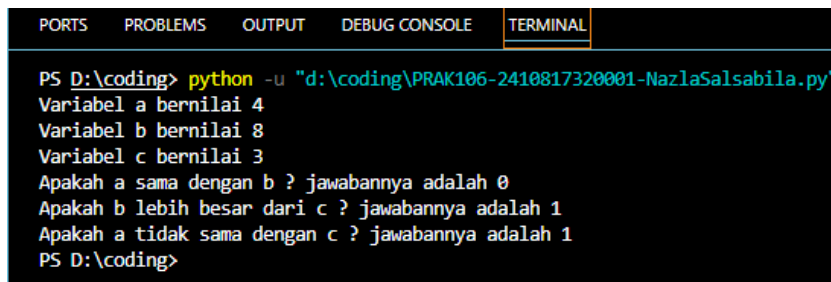


```

PORTS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\ASUS\Desktop\coding> cd "d:\coding\" ; if ($?) { gcc PRAK106-2410817320001-NazlaSalsabila.c -o PRAK106-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK106-2410817320001-NazlaSalsabila }
Variabel a bernilai 4
Variabel b bernilai 8
Variabel c bernilai 3
Apakah a sama dengan b ? jawabannya adalah 0
Apakah b lebih besar dari c ? jawabannya adalah 1
Apakah a tidak sama dengan c ? jawabannya adalah 1
PS D:\coding>

```

Gambar 21. Screenshot Output Soal 6 Bahasa C



```

PORTS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\coding> python -u "d:\coding\PRAK106-2410817320001-NazlaSalsabila.py"
Variabel a bernilai 4
Variabel b bernilai 8
Variabel c bernilai 3
Apakah a sama dengan b ? jawabannya adalah 0
Apakah b lebih besar dari c ? jawabannya adalah 1
Apakah a tidak sama dengan c ? jawabannya adalah 1
PS D:\coding>

```

Gambar 22. Screenshot Output Soal 6 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [4], `int a = 4;` mendefinisikan variabel `x` dengan tipe data `int` (integer) dan memberikan nilai 4.

Pada baris [5], `int b = 8;` mendefinisikan variabel `y` dengan tipe data `int` (integer) dan memberikan nilai 8.

Pada baris [6], `int c = 3;` mendefinisikan variabel `z` dengan tipe data `int` (integer) dan memberikan nilai 3.

Pada baris [8 – 10], fungsi `printf` digunakan untuk menampilkan teks dan nilai variabel ke layar. String yang ingin ditampilkan ditulis di dalam tanda kutip ganda `" "`. `%d` adalah placeholder untuk menampilkan nilai integer dari variabel. Nilai variabel `a`, `b`, dan `c` ditempatkan setelah tanda koma dan masing-masing akan menggantikan `%d` pada posisi tersebut. `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [12], `==` digunakan untuk memeriksa apakah nilai `a` sama dengan nilai `b`. Jika `a` sama dengan `b`, maka hasilnya 1 (benar), jika tidak, hasilnya 0 (salah). Dalam contoh ini, `a` tidak sama dengan `b` (karena $4 \neq 8$), jadi `hasil_sama` akan bernilai 0.

Pada baris [14], `>` digunakan untuk memeriksa apakah `b` lebih besar dari `c`. Jika benar, maka hasilnya 1, jika salah, hasilnya 0. Karena `b` (8) lebih besar dari `c` (3), maka `hasil_lebih_besar` akan bernilai 1.

Pada baris [16], `!=` digunakan untuk memeriksa apakah `a` tidak sama dengan `c`. Jika `a` tidak sama dengan `c`, hasilnya 1, jika sama, hasilnya 0. Karena `a` (4) tidak sama dengan `c` (3), `hasil_tidak_sama` akan bernilai 1.

Pada baris [18 – 20], akan dilakukan hasil pengecekan logika (apakah benar atau salah) ditampilkan menggunakan `printf`. `%d` digunakan untuk menampilkan hasil dalam bentuk angka (0 untuk salah, 1 untuk benar). `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [22], `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [23], terdapat `}`, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python:

Pada baris [1], `a = 4`, mendefinisikan variabel `a` dengan memberikan nilai 4.

Pada baris [2], `b = 8`, mendefinisikan variabel `b` dengan memberikan nilai 8.

Pada baris [3], `c = 3`, mendefinisikan variabel `c` dengan memberikan nilai 3.

Pada baris [5 – 7], fungsi `print()` digunakan untuk menampilkan teks dan nilai dari variabel. Nilai variabel `a`, `b`, `x`, dan `y` ditempatkan setelah tanda koma.

Pada baris [9], `==` digunakan untuk memeriksa apakah nilai `a` sama dengan nilai `b`. Jika `a` sama dengan `b`, maka hasilnya 1 (benar), jika tidak, hasilnya 0 (salah). Dalam contoh ini, `a` tidak sama dengan `b` (karena $4 \neq 8$), jadi `hasil_sama` akan bernilai 0.

Pada baris [11], `>` digunakan untuk memeriksa apakah `b` lebih besar dari `c`. Jika benar, maka hasilnya 1, jika salah, hasilnya 0. Karena `b` (8) lebih besar dari `c` (3), maka `hasil_lebih_besar` akan bernilai 1.

Pada baris [13], `!=` digunakan untuk memeriksa apakah `a` tidak sama dengan `c`. Jika `a` tidak sama dengan `c`, hasilnya 1, jika sama, hasilnya 0. Karena `a` (4) tidak sama dengan `c` (3), `hasil_tidak_sama` akan bernilai 1.

Pada baris [15 – 17], akan dilakukan hasil pengecekan logika (apakah benar atau salah) ditampilkan menggunakan `print`.

SOAL 7

Pak Dengklek baru saja membeli sebidang tanah berbentuk segitiga sebarang dengan panjang tiap sisi tanah berturut-turut 4 m, 5 m, dan 7 m. Tanah tersebut hendak dipasang pagar agar bebek yang di pelihara Pak Dengklek tidak berkeliaran sembarangan. Biaya pemasangan pagar per meter-nya adalah Rp 85.000,00.

Buatkan sebuah program untuk membantu Pak Dengklek menghitung biaya yang diperlukan agar seluruh tanahnya dikelilingi pagar menggunakan bahasa Python dan bahasa C.

Output

Diketahui :

Panjang sisi segitiga berturut-turut adalah 4, 5, dan 7

Keliling Tanah Pak Dengklek adalah 16

Harga tanah Per Meter adalah 85000

Jawaban :

Biaya yang diperlukan Pak Dengklek adalah : Rp 1360000

Simpan coding anda dengan nama: **PRAK107-NIM-Nama.py** dan **PRAK107-NIM-Nama.c**

Note: gunakan format specifier untuk menampilkan angka yang ada pada contoh output

A. Source Code

Tabel 23. Source Code Jawaban Soal 7 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	int sisi_a = 4;
5	int sisi_b = 5;
6	int sisi_c = 7;
7	
8	int harga_per_meter = 85000;
9	

10	int keliling = sisi_a + sisi_b + sisi_c;
11	
12	int total_biaya = keliling *
13	harga_per_meter;
14	printf("Diketahui :\n");
15	printf("Panjang sisi segitiga berturut-
16	turut adalah %d, %d, dan %d\n", sisi_a,
17	sisi_b, sisi_c);
18	printf("Keliling Tanah Pak Dengklek
19	adalah %d\n", keliling);
20	printf("Harga tanah Per Meter adalah
21	%d\n", harga_per_meter);
22	printf("Jawaban :\n");
	printf("Biaya yang diperlukan Pak
	Dengklek adalah : Rp %d\n", total_biaya);
	return 0;
	}

Tabel 24. Source Code Jawaban Soal 7 Bahasa Python

1	sisi_a = 4
2	sisi_b = 5
3	sisi_c = 7
4	
5	harga_per_meter = 85000
6	
7	keliling = sisi_a + sisi_b + sisi_c
8	
9	total_biaya = keliling * harga_per_meter
10	
11	print("Diketahui :")
12	print("Panjang sisi segitiga berturut-turut
13	adalah %d, %d, dan %d" % (sisi_a, sisi_b,
14	sisi_c))
15	print("Keliling Tanah Pak Dengklek adalah %d"
	% keliling)
	print("Harga tanah Per Meter adalah %d" %
	harga_per_meter)
	print("Jawaban :")

16	<code>print("Biaya yang diperlukan Pak Dengklek adalah : Rp %d" % total_biaya)</code>
----	---

B. Output Program

```

PORTS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\ASUS\Desktop\coding> cd "d:\coding\" ; if ($?) { gcc PRAK107-2410817320001-NazlaSalsabila.c -o PRAK107-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK107-2410817320001-NazlaSalsabila }
Diketahui :
Panjang sisi segitiga berturut-turut adalah 4, 5, dan 7
Keliling Tanah Pak Dengklek adalah 16
Harga tanah Per Meter adalah 85000
Jawaban :
Biaya yang diperlukan Pak Dengklek adalah : Rp 1360000
PS D:\coding>

```

Gambar 23. Screenshot Output Soal 7 Bahasa C

```

PORTS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\coding> python -u "d:\coding\PRAK107-2410817320001-NazlaSalsabila.py"
Diketahui :
Panjang sisi segitiga berturut-turut adalah 4, 5, dan 7
Keliling Tanah Pak Dengklek adalah 16
Harga tanah Per Meter adalah 85000
Jawaban :
Biaya yang diperlukan Pak Dengklek adalah : Rp 1360000
PS D:\coding>

```

Gambar 24. Screenshot Output Soal 7 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [4], `int sisi a = 4;` mendefinisikan `int` sisi A dengan tipe data `int` (integer) dan memberikan nilai panjang sisi segitiga 4.

Pada baris [5], `int sisi b = 5;` mendefinisikan `int` sisi A dengan tipe data `int` (integer) dan memberikan nilai panjang sisi segitiga 5.

Pada baris [6], `int sisi c = 7;` mendefinisikan `int` sisi A dengan tipe data `int` (integer) dan memberikan nilai panjang sisi segitiga 7.

Pada baris [8], `int harga per meter = 85000;` mendefinisikan biaya pemasangan pagar per meter, yang nilainya 85000.

Pada baris [10], menghitung keliling tanah segitiga dengan menjumlahkan panjang ketiga sisinya: $\text{keliling} = 4 + 5 + 7$, sehingga keliling bernilai 16.

Pada baris [12], menghitung total biaya pemasangan pagar dengan mengalikan keliling tanah (16 meter) dengan harga per meter (85000), sehingga `total_biaya` bernilai 1360000.

Pada baris [14], menampilkan teks "Diketahui :" di layar. `\n` menunjukkan karakter baris baru.

Pada baris [15], menampilkan panjang sisi segitiga menggunakan format specifier `%d`, yang digunakan untuk menampilkan nilai bertipe `int`. Variabel `sisi_a`, `sisi_b`, dan `sisi_c` akan digantikan pada posisi `%d`.

Pada baris [16], menampilkan hasil keliling yang telah dihitung sebelumnya.

Pada baris [17], menampilkan harga pemasangan pagar per meter.

Pada baris [18], menampilkan teks "Jawaban :" di layar.

Pada baris [19], menampilkan total biaya pemasangan pagar dengan format specifier `%d`, menggantikan dengan nilai dari `total_biaya`.

Pada baris [21], `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [22], terdapat `}`, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python:

Pada baris [1], `int sisi a = 4;` mendefinisikan `int` sisi A dengan memberikan nilai panjang sisi segitiga 4.

Pada baris [2], `int sisi b = 5;` mendefinisikan `int sisi A` dengan memberikan nilai panjang sisi segitiga 5.

Pada baris [3], `int sisi c = 7;` mendefinisikan `int sisi A` dengan memberikan nilai panjang sisi segitiga 7.

Pada baris [5], `int harga per meter = 85000;` mendefinisikan biaya pemasangan pagar per meter, yang nilainya 85000.

Pada baris [7], menghitung keliling tanah segitiga dengan menjumlahkan panjang ketiga sisinya: $\text{keliling} = 4 + 5 + 7$, sehingga keliling bernilai 16.

Pada baris [9], menghitung total biaya pemasangan pagar dengan mengalikan keliling tanah (16 meter) dengan harga per meter (85000), sehingga `total_biaya` bernilai 1360000.

Pada baris [11 – 16], fungsi `print()` digunakan untuk menampilkan informasi ke layar. Format specifier `%d` digunakan untuk menampilkan nilai numerik (`int`). Nilai variabel seperti `sisi_a`, `keliling`, `harga_per_meter`, dan `total_biaya` ditampilkan sesuai dengan posisi `%d` pada string.

SOAL 8

Hari ini Pak Dengklek jogging mengelilingi taman berbentuk lingkaran sebanyak 5 putaran. Berdasarkan aplikasi Runkeeper pada smartphone yang digunakan, Pak Dengklek telah berlari sejauh 14 kilometer. Berapakah jari-jari taman yang dikelilingi Pak Dengklek?

Output

Diketahui :

Pak Dengklek mengelilingi taman = 5 Putaran

Jarak tempuh Pak Dengklek = 14 Kilometer

Jawaban :

Jari-jari taman yang dikelilingi Pak Dengklek adalah 0.45 Kilometer

Simpan coding anda dengan nama: **PRAK108-NIM-Nama.py** dan **PRAK108-NIM-Nama.c**

Note: gunakan tipe data float untuk jari-jari dengan ketelitian 2 angka dibelakang koma

A. Source Code

Tabel 25. Source Code Jawaban Soal 8 Bahasa C

1	#include <stdio.h>
2	#include <math.h>
3	
4	int main() {
5	float putaran = 5.0;
6	float jarak_tempuh = 14.0;
7	
8	float keliling_per_putaran = jarak_tempuh
9	/ putaran;
10	float jari_jari = keliling_per_putaran /
	(2 * M_PI);

11	
12	printf("Diketahui :\n");
	printf("Pak Dengklek mengelilingi taman =
13	%.0f Putaran\n", putaran);
	printf("Jarak tempuh Pak Dengklek = %.0f
14	Kilometer\n", jarak_tempuh);
15	printf("\n");
16	printf("Jawaban :\n");
	printf("Jari-jari taman yang dikelilingi
	Pak Dengklek adalah %.2f Kilometer\n",
17	jari_jari);
18	
19	return 0;
20	}

Tabel 26. Source Code Jawaban Soal 8 Bahasa Python

1	import math
2	
3	putaran = 5.0
4	jarak_tempuh = 14.0
5	
6	keliling_per_putaran = jarak_tempuh / putaran
7	
	jari_jari = keliling_per_putaran / (2 *
8	math.pi)
9	
10	print("Diketahui :")
	print("Pak Dengklek mengelilingi taman = %.0f
11	Putaran" % putaran)
	print("Jarak tempuh Pak Dengklek = %.0f
12	Kilometer" % jarak_tempuh)
13	print()
14	print("Jawaban :")

15	<code>print("Jari-jari taman yang dikelilingi Pak Dengklek adalah %.2f Kilometer" % jari_jari)</code>
----	---

B. Output Program

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Jari-jari taman yang dikelilingi Pak Dengklek adalah 0.45 Kilometer
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK108-2410817320001-NazlaSalsabila.c -o PRAK108-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK108-2410817320001-NazlaSalsabila }
Diketahui :
Pak Dengklek mengelilingi taman = 5 Putaran
Jarak tempuh Pak Dengklek = 14 Kilometer

Jawaban :
Jari-jari taman yang dikelilingi Pak Dengklek adalah 0.45 Kilometer
PS D:\coding>

```

Gambar 25. Screenshot Output Soal 8 Bahasa C

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> python -u "d:\coding\PRAK108-2410817320001-NazlaSalsabila.py"
Diketahui :
Pak Dengklek mengelilingi taman = 5 Putaran
Jarak tempuh Pak Dengklek = 14 Kilometer

Jawaban :
Jari-jari taman yang dikelilingi Pak Dengklek adalah 0.45 Kilometer
PS D:\coding>

```

Gambar 26. Screenshot Output Soal 8 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [2], terdapat `#include <math.h>`. Fungsinya untuk menggunakan fungsi matematika, seperti `M_PI` yang mewakili nilai konstanta π (pi).

Pada baris [4], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [5], variabel putaran menyimpan jumlah putaran yang dilakukan Pak Dengklek (5 putaran).

Pada baris [6], variabel jarak_tempuh menyimpan total jarak yang ditempuh Pak Dengklek (14 kilometer). Tipe data float digunakan untuk nilai desimal agar hasil perhitungan lebih akurat.

Pada baris [8], menghitung keliling satu putaran taman dengan membagi total jarak dengan jumlah putaran. Dalam hal ini, keliling per putaran adalah $14 \text{ km} / 5 = 2.8 \text{ km}$.

Pada baris [10], Menghitung jari-jari taman. Rumus keliling lingkaran adalah $K=2\pi r$, sehingga jari-jari bisa dihitung dengan rumus $r=K/2\pi$, di mana K adalah keliling satu putaran.

Pada baris [12], menampilkan teks "Diketahui:" di layar.

Pada baris [13], menampilkan jumlah putaran yang ditempuh Pak Dengklek. Format specifier %.0f digunakan untuk menampilkan angka float tanpa angka desimal (karena angka desimalnya nol).

Pada baris [14], menampilkan total jarak yang ditempuh Pak Dengklek dalam kilometer. Sama seperti sebelumnya, %.0f digunakan untuk menampilkan nilai desimal yang dibulatkan.

Pada baris [16], menampilkan teks "Jawaban :" di layar.

Pada baris [17], menampilkan jari-jari taman dengan dua angka desimal (%.2f).

Pada baris [19], return 0; mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [20], terdapat }, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python:

Pada baris [3], variabel putaran menyimpan jumlah putaran yang dilakukan Pak Dengklek (5 putaran).

Pada baris [4], variabel jarak_tempuh menyimpan total jarak yang ditempuh Pak Dengklek (14 kilometer). Tipe data float digunakan untuk nilai desimal agar hasil perhitungan lebih akurat.

Pada baris [6], menghitung keliling satu putaran taman dengan membagi total jarak dengan jumlah putaran. Dalam hal ini, keliling per putaran adalah $14 \text{ km} / 5 = 2.8 \text{ km}$.

Pada baris [8], Menghitung jari-jari taman. Rumus keliling lingkaran adalah $K=2\pi r$, sehingga jari-jari bisa dihitung dengan rumus $r=K/2\pi$, di mana K adalah keliling satu putaran.

Pada baris [11], menampilkan jumlah putaran yang ditempuh Pak Dengklek. Format specifier `%.0f` digunakan untuk menampilkan angka float tanpa angka desimal (karena angka desimalnya nol).

Pada baris [12], menampilkan total jarak yang ditempuh Pak Dengklek dalam kilometer. Sama seperti sebelumnya, `%.0f` digunakan untuk menampilkan nilai desimal yang dibulatkan.

Pada baris [15], menampilkan jari-jari taman dengan dua angka desimal (`%.2f`)

SOAL 9

Cardia Riverlands merupakan tanah yang damai sampai Yu Zhong jelmaan dari Black Dragon datang dengan membawa 958.730 pasukan dan memporak-porandakan tempat tersebut. Mendengar berita atas kacaunya Cardia Riverlands, padepokan Dragon Altar mengirim pahlawan yang terdiri dari Zilong, Ling, Baxia, Wanwan, dan Chang'e dengan misi untuk mengalahkan semua pasukan Yu Zhong. Jika para pahlawan Dragon Altar bersepakat untuk membagi musuh sama rata, berapa pasukan yang harus dikalahkan oleh setiap pahlawan Dragon Altar?

Buatlah program untuk menghitung jumlah pasukan yang harus dikalahkan oleh setiap pahlawan Dragon Altar!

Output	Note = tanda tanya “?” disesuaikan dengan soal
Jumlah pasukan yang dibawa Yu Zhong = ?	
Jumlah pahlawan = ?	
Jumlah pasukan yang harus dikalahkan setiap pahlawan adalah 191746 pasukan	

Simpan coding anda dengan nama: **PRAK109-NIM-Nama.py** dan **PRAK109-NIM-Nama.c**

A. Source Code

Tabel 27. Source Code Jawaban Soal 9 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	int jumlah_pasukan = 958730;
5	int jumlah_pahlawan = 5;
6	
7	int pasukan_per_pahlawan = jumlah_pasukan
7	/ jumlah_pahlawan;
8	
9	printf("Jumlah pasukan yang dibawa Yu
9	Zhong = %d\n", jumlah_pasukan);

10	<code>printf("Jumlah pahlawan = %d\n", jumlah_pahlawan);</code>
11	<code>printf("Jumlah pasukan yang harus dikalahkan setiap pahlawan adalah %d pasukan\n", pasukan_per_pahlawan);</code>
12	
13	<code>return 0;</code>
14	<code>}</code>

Tabel 28. Source Code Jawaban Soal 9 Bahasa Python

1	<code>jumlah_pasukan = 958730</code>
2	<code>jumlah_pahlawan = 5</code>
3	
4	<code>pasukan_per_pahlawan = jumlah_pasukan // jumlah_pahlawan</code>
5	
6	<code>print("Jumlah pasukan yang dibawa Yu Zhong = %d" % jumlah_pasukan)</code>
7	<code>print("Jumlah pahlawan = %d" % jumlah_pahlawan)</code>
8	<code>print("Jumlah pasukan yang harus dikalahkan setiap pahlawan adalah %d pasukan" % pasukan_per_pahlawan)</code>

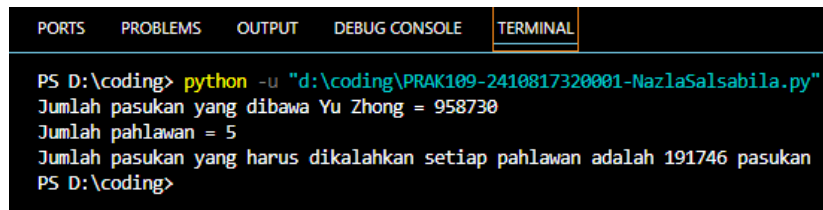
B. Output Program

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\ASUS\Desktop\coding> cd "d:\coding\"; if ($?) { gcc PRAK109-2410817320001-NazlaSalsabila.c -o PRAK109-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK109-2410817320001-NazlaSalsabila }
Jumlah pasukan yang dibawa Yu Zhong = 958730
Jumlah pahlawan = 5
Jumlah pasukan yang harus dikalahkan setiap pahlawan adalah 191746 pasukan
PS D:\coding>

```

Gambar 27. Screenshot Output Soal 9 Bahasa C



```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> python -U "d:\coding\PRAK109-2410817320001-NazlaSalsabila.py"
Jumlah pasukan yang dibawa Yu Zhong = 958730
Jumlah pahlawan = 5
Jumlah pasukan yang harus dikalahkan setiap pahlawan adalah 191746 pasukan
PS D:\coding>

```

Gambar 28. Screenshot Output Soal 9 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [4], variabel `jumlah_pasukan` dengan nilai 958730 yang menunjukkan jumlah pasukan yang dibawa oleh Yu Zhong.

Pada baris [5], variabel `jumlah_pahlawan` dengan nilai 5, yang menunjukkan jumlah pahlawan yang akan melawan pasukan tersebut.

Pada baris [7], Menghitung berapa banyak pasukan yang harus dikalahkan oleh setiap pahlawan dengan membagi total jumlah pasukan dengan jumlah pahlawan. Hasilnya adalah setiap pahlawan harus mengalahkan 191746 pasukan.

Pada baris [9], menampilkan jumlah total pasukan yang dibawa oleh Yu Zhong.

Pada baris [10], menampilkan jumlah pahlawan.

Pada baris [11], menampilkan jumlah pasukan yang harus dikalahkan oleh setiap pahlawan.

Pada baris [13], `return 0`; mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [14], terdapat `}`, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python:

Pada baris [1], variabel `jumlah_pasukan` dengan nilai 958730 yang menunjukkan jumlah pasukan yang dibawa oleh Yu Zhong.

Pada baris [2], variabel `jumlah_pahlawan` dengan nilai 5, yang menunjukkan jumlah pahlawan yang akan melawan pasukan tersebut.

Pada baris [4], Menghitung berapa banyak pasukan yang harus dikalahkan oleh setiap pahlawan dengan membagi total jumlah pasukan dengan jumlah pahlawan. Hasilnya adalah setiap pahlawan harus mengalahkan 191746 pasukan.

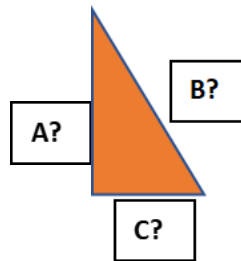
Pada baris [6], menampilkan jumlah total pasukan yang dibawa oleh Yu Zhong.

Pada baris [7], menampilkan jumlah pahlawan.

Pada baris [8], menampilkan jumlah pasukan yang harus dikalahkan oleh setiap pahlawan.

SOAL 10

Buatlah sebuah program yang digunakan untuk menghitung keliling dan luas segitiga siku- siku jika diketahui alas = 5cm dan tinggi = 12cm.



Output	Note = tanda tanya “?” disesuaikan dengan soal
<p>Diketahui :</p> <p>Alas = 5 cm</p> <p>Tinggi = 12 cm</p> <p>Jawab :</p> <p>Sisi A = ? cm</p> <p>Sisi B = ? cm</p> <p>Sisi C = ? cm</p> <p>Keliling = 30 cm</p> <p>Luas = 30 cm</p>	

Simpan coding anda dengan nama: **PRAK110-NIM-Nama.py** dan **PRAK110-NIM-Nama.c**

A. Source Code

Tabel 29. Source Code Jawaban Soal 10 Bahasa C

1	#include <stdio.h>
2	#include <math.h>

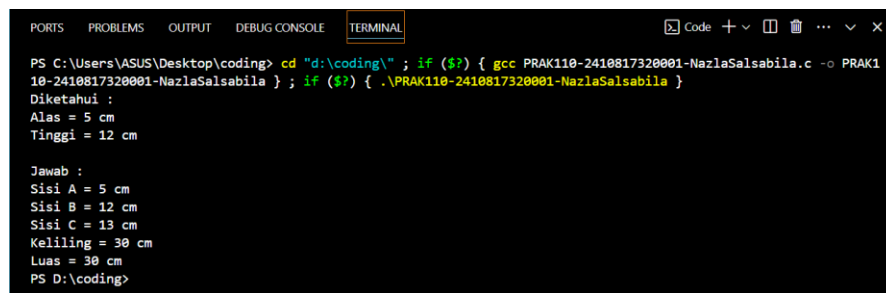
3	
4	int main() {
5	int alas = 5;
6	int tinggi = 12;
7	
8	double sisi_miring = sqrt(pow(alas,
9	2) + pow(tinggi, 2));
10	
11	double keliling = alas + tinggi +
12	sisi_miring;
13	
14	double luas = 0.5 * alas * tinggi;
15	
16	printf("Diketahui : \n");
17	printf("Alas = %d cm \n", alas);
18	printf("Tinggi = %d cm \n", tinggi);
19	printf(" \n");
20	printf("Jawab : \n");
21	printf("Sisi A = %d cm \n", alas);
22	printf("Sisi B = %d cm \n", tinggi);
23	printf("Sisi C = %.0f cm \n",
24	sisi_miring);
25	printf("Keliling = %.0f cm \n",
26	keliling);
	printf("Luas = %.0f cm \n", luas);
	return 0;
	}

Tabel 30. Source Code Jawaban Soal 10 Bahasa Python

1	import math
2	
3	alas = 5

4	tinggi = 12
5	
6	sisi_miring = math.sqrt(alas**2 + tinggi**2)
7	
8	keliling = alas + tinggi + sisi_miring
9	
10	luas = 0.5 * alas * tinggi
11	
12	print("Diketahui :")
13	print("Alas = %d cm" % alas)
14	print("Tinggi = %d cm" % tinggi)
15	print()
16	print("Jawab :")
17	print("Sisi A = %d cm" % alas)
18	print("Sisi B = %d cm" % tinggi)
19	print("Sisi C = %.0f cm" % sisi_miring)
20	print("Keliling = %.0f cm" % keliling)
21	print("Luas = %.0f cm" % luas)

B. Output Program



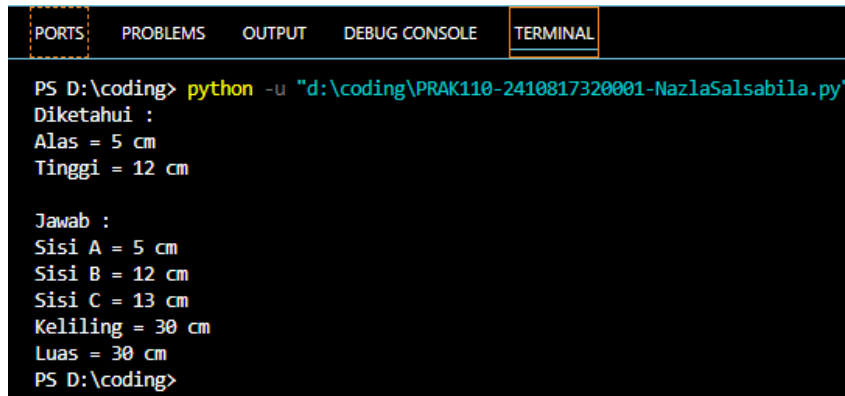
```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\ASUS\Desktop\coding> cd "d:\coding\" ; if ($?) { gcc PRAK110-2410817320001-NazlaSalsabila.c -o PRAK1
10-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK110-2410817320001-NazlaSalsabila }
Diketahui :
Alas = 5 cm
Tinggi = 12 cm

Jawab :
Sisi A = 5 cm
Sisi B = 12 cm
Sisi C = 13 cm
Keliling = 30 cm
Luas = 30 cm
PS D:\coding>

```

Gambar 29. Screenshot Output Soal 10 Bahasa C



```

PORTS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\coding> python -u "d:\coding\PRAK110-2410817320001-NazlaSalsabila.py"
Diketahui :
Alas = 5 cm
Tinggi = 12 cm

Jawab :
Sisi A = 5 cm
Sisi B = 12 cm
Sisi C = 13 cm
Keliling = 30 cm
Luas = 30 cm
PS D:\coding>

```

Gambar 30. Screenshot Output Soal 10 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [2], terdapat `#include <math.h>`. Fungsinya untuk menggunakan fungsi matematika, seperti `M_PI` yang mewakili nilai konstanta π (pi).

Pada baris [4], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [5], `int alas = 5;` mendefinisikan variabel `alas` dengan tipe data `int` (integer) dan memberikan nilai `alas` 5.

Pada baris [6], `int tinggi = 12;` mendefinisikan variabel `tinggi` dengan tipe data `int` (integer) dan memberikan nilai `tinggi` 12.

Pada baris [8], Teorema Pythagoras menyatakan bahwa untuk segitiga siku-siku dengan sisi miring dihitung: $\text{akar } a^2 + b^2 = \text{akar } 5^2 + 12^2 = \text{akar } 25 + 144 = \text{akar } 169 = 13 \text{ cm}$.

Pada baris [10], keliling segitiga adalah jumlah panjang ketiga sisinya, yaitu alas + tinggi + sisi_miring, sehingga kelilingnya adalah $5 + 12 + 13 = 30$ cm.

Pada baris [12], Luas segitiga dihitung menggunakan rumus $luas = \frac{1}{2} \times \text{alas} \times \text{tinggi}$. Dalam hal ini, luas segitiga adalah $0.5 \times 5 \times 12 = 30$ cm².

Pada baris [14 – 23], fungsi printf() digunakan untuk menampilkan nilai variabel dengan format %.0f untuk menampilkan bilangan desimal yang dibulatkan.. \n digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [25], return 0; mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [26], terdapat }, berfungsi untuk menandai akhir akhir blok kode.

Bahasa Python

Pada baris [1], import math fungsinya untuk menggunakan fungsi matematika, seperti sqrt (akar kuadrat) dan konstanta matematika lainnya.

Pada baris [3], alas = 5; mendefinisikan variabel alas dengan memberikan nilai alas 5.

Pada baris [4], tinggi = 12; mendefinisikan variabel tinggi dengan memberikan nilai tinggi 12.

Pada baris [6], Teorema Pythagoras menyatakan bahwa untuk segitiga siku-siku dengan sisi miring dihitung: $\text{akar } a^2 + b^2 = \text{akar } 5^2 + 12^2 = \text{akar } 25 + 144 = \text{akar } 169 = 13$ cm.

Pada baris [8], keliling segitiga adalah jumlah panjang ketiga sisinya, yaitu alas + tinggi + sisi_miring, sehingga kelilingnya adalah $5 + 12 + 13 = 30$ cm.

Pada baris [10], Luas segitiga dihitung menggunakan rumus $luas = \frac{1}{2} \times \text{alas} \times \text{tinggi}$. Dalam hal ini, luas segitiga adalah $0.5 \times 5 \times 12 = 30$ cm².

Pada baris [12 – 21], fungsi print() digunakan untuk menampilkan hasil perhitungan ke layar. Format specifier %d digunakan untuk menampilkan bilangan bulat, dan %.0f digunakan untuk menampilkan bilangan float dengan 0 angka desimal.

MODUL 2: INPUT DAN OUTPUT

SOAL 1

Buatlah program yang dapat menginput biodata dan menghasilkan output dengan menampilkan biodata tersebut dalam bahasa Python dan C.

Note : Sesuaikan dengan biodata kalian, Output hasilnya sama dengan inputan.

Input	
Nama	:
NIM	:
Kelas Paralel	:
Tempat/Tanggal Lahir	:
Alamat	:
Hobby	:
No. HP	:
Output	
Nama	: Nama Lengkap
NIM	:
Kelas Paralel	:
Tempat/Tanggal Lahir	: Kota/dd-mm-yyyy
Alamat	:
Hobby	:
No. HP	:

Simpan coding anda dengan nama: **PRAK201-NIM-Nama.py** dan **PRAK201-NIM-Nama.c**

A. Source Code*Tabel 31. Source Code Jawaban Soal 1 Bahasa C*

1	#include <stdio.h>
2	
3	int main() {
	char nama[50], nim[20], kelas[10],
	tempat_tanggal_lahir[100], alamat[100],
4	hobby[50], no_hp[15];
5	
6	printf("\n==== INPUT =====\n");
7	printf("Nama : ");
8	scanf(" %[^\\n]", nama);
9	
10	printf("NIM : ");
11	scanf("%s", nim);
12	
13	printf("Kelas Paralel : ");
14	scanf("%s", kelas);
15	
16	printf("Tempat/Tanggal Lahir: ");
	scanf(" %[^\\n]",
17	tempat_tanggal_lahir);
18	
19	printf("Alamat : ");
20	scanf(" %[^\\n]", alamat);
21	
22	printf("Hobby : ");
23	scanf(" %[^\\n]", hobby);
24	
25	printf("No. HP : ");
26	scanf("%s", no_hp);
27	

28	printf("\n===== OUTPUT =====\n");
29	printf("Nama : %s\n", nama);
30	printf("NIM : %s\n", nim);
31	printf("Kelas Paralel : %s\n", kelas);
32	printf("Tempat/Tanggal Lahir: %s\n", tempat_tanggal_lahir);
33	printf("Alamat : %s\n", alamat);
34	printf("Hobby : %s\n", hobby);
35	printf("No. HP : %s\n", no_hp);
36	
37	return 0;
38	}

Tabel 32. Source Code Jawaban Soal 1 Bahasa Python

1	print("\n===== INPUT =====")
2	nama = input("Nama : ")
3	nim = input("NIM : ")
4	kelas = input("Kelas Paralel : ")
5	tempat_tanggal_lahir = input("Tempat/Tanggal Lahir: ")
6	alamat = input("Alamat : ")
7	hobby = input("Hobby : ")
8	no_hp = input("No. HP : ")
9	
10	print("\n===== OUTPUT =====")
11	print(f>Nama : {nama}")
12	print(f"NIM : {nim}")
13	print(f"Kelas Paralel : {kelas}")

14	<code>print(f"Tempat/Tanggal Lahir: {tempat_tanggal_lahir}")</code>
15	<code>print(f"Alamat : {alamat}")</code>
16	<code>print(f"Hobby : {hobby}")</code>
17	<code>print(f"No. HP : {no_hp}")</code>

B. Output Program

```

PORTS  PROBLEMS  2  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAC201-2410817320001-NazlaSalsabila.c -o PRAC201-2410817320001-NazlaSalsabila ; if ($?) { .\PRAC201-2410817320001-NazlaSalsabila }

===== INPUT =====
Nama      : Nazla Salsabila
NIM       : 2410817320001
Kelas Paralel : Paralel-2
Tempat/Tanggal Lahir: Banjarmasin/10-05-2006
Alamat    : Jalan AMD Perumahan Abdi Persada Blok Limau 3
Hobby     : Bermain Game Online
No. HP    : 082153097691

===== OUTPUT =====
Nama      : Nazla Salsabila
NIM       : 2410817320001
Kelas Paralel : Paralel-2
Tempat/Tanggal Lahir: Banjarmasin/10-05-2006
Alamat    : Jalan AMD Perumahan Abdi Persada Blok Limau 3
Hobby     : Bermain Game Online
No. HP    : 082153097691
PS D:\coding>

```

Gambar 31. Screenshot Output Soal 1 Bahasa C

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> python -u "d:\coding\tempCodeRunnerFile.py"

===== INPUT =====
Nama      : Nazla Salsabila
NIM       : 2410817320001
Kelas Paralel : Paralel-2
Tempat/Tanggal Lahir: Banjarmasin/10-05-2006
Alamat    : Jalan AMD Perumahan Abdi Persada Blok Limau 3
Hobby     : Bermain Game Online
No. HP    : 082153097691

===== OUTPUT =====
Nama      : Nazla Salsabila
NIM       : 2410817320001
Kelas Paralel : Paralel-2
Tempat/Tanggal Lahir: Banjarmasin/10-05-2006
Alamat    : Jalan AMD Perumahan Abdi Persada Blok Limau 3
Hobby     : Bermain Game Online
No. HP    : 082153097691
PS D:\coding>

```

Gambar 32. Screenshot Output Soal 1 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `char` yang merupakan tipe data untuk menyimpan karakter dan juga dapat digunakan untuk menyimpan string. `nama[50]` berfungsi menyediakan ruang untuk menyimpan nama hingga 50 karakter. Variabel lainnya seperti `NIM`, `kelas`, `tempat/tanggal_lahir`, `alamat`, `hobby`, dan `no_hp` memiliki fungsi yang sama, tetapi dengan ukuran berbeda untuk menyimpan ruang karakternya.

Pada baris [6, 8, 17, 20, 23, dan 28 – 35], terdapat tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [6, 7, 10, 13, 16, 19, 22, 25, dan 28 – 35], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [8, 11, 14, 17, 20, 23, dan 26], terdapat `scanf` yang merupakan perintah untuk membaca input berupa string (teks) dan menyimpannya dalam variable `nama`, `NIM`, `kelas`, `tempat/tanggal_lahir`, `Alamat`, `hobby`, dan `no. hp`. Tanda `%s` digunakan untuk membaca dan menampilkan string tanpa spasi. Format `%[^\n]` digunakan untuk membaca string dengan menggunakan spasi. `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [37], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [38], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1 dan 10], terdapat tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [1 dan 10 – 17], terdapat `print` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [2 – 8], terdapat input () untuk mengambil input. Pada saat program dijalankan, dan program meminta data, maka harus memasukkan data yang diinginkan dan menekan Enter.

Pada baris [11 – 17], terdapat f-strings untuk menampilkan nilai variabel nama, NIM, kelas paralel, tempat/tanggal lahir, Alamat, hobby, dan no. hp.

SOAL 2

Buatlah program yang dapat menginputkan dan menghasilkan output berupa hasil dari nilai pertama ditambah nilai kedua adalah 34.50 dengan bahasa Python dan C (**ketelitian dua angka di belakang koma**)

Test case ke 1 :

Input
Masukkan Nilai Pertama : 14
Masukkan Nilai Kedua : 20.5
Output
Hasil dari penjumlahan nilai pertama “14” dan nilai kedua “20.5” adalah “34.50”

Test case ke 2 :

Input
Masukkan Nilai Pertama : 0.45
Masukkan Nilai Kedua : 99.5
Output
Hasil dari penjumlahan nilai pertama “0.45” dan nilai kedua “99.5” adalah “99.95”

Simpan coding anda dengan nama: **PRAK202-NIM-Nama.py** dan **PRAK202-NIM-Nama.c**

A. Source Code

Tabel 33. Source Code Jawaban Soal 2 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	float nilai_pertama1, nilai_kedua1,
	hasil1;

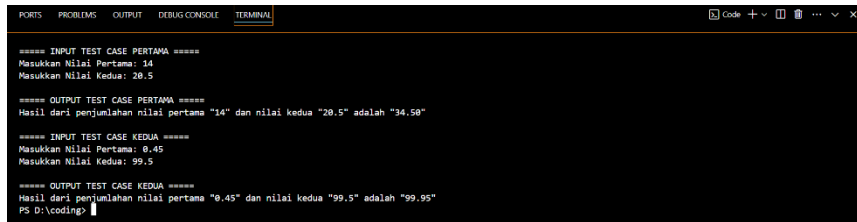
5	float nilai_pertama2, nilai_kedua2, hasil2;
6	
7	printf("\n==== INPUT TEST CASE PERTAMA ====\n");
8	printf("Masukkan Nilai Pertama: ");
9	scanf("%f", &nilai_pertama1);
10	printf("Masukkan Nilai Kedua: ");
11	scanf("%f", &nilai_kedua1);
12	
13	hasil1 = nilai_pertama1 + nilai_kedua1;
14	
15	printf("\n==== OUTPUT TEST CASE PERTAMA ====\n");
16	printf("Hasil dari penjumlahan nilai pertama \"%d\" dan nilai kedua \"%.1f\" adalah \"%.2f\"\\n", (int)nilai_pertama1, nilai_kedua1, hasil1);
17	
18	
19	printf("\n==== INPUT TEST CASE KEDUA ====\n");
20	printf("Masukkan Nilai Pertama: ");
21	scanf("%f", &nilai_pertama2);
22	printf("Masukkan Nilai Kedua: ");
23	scanf("%f", &nilai_kedua2);
24	
25	hasil2 = nilai_pertama2 + nilai_kedua2;
26	
27	printf("\n==== OUTPUT TEST CASE KEDUA ====\n");
28	printf("Hasil dari penjumlahan nilai pertama \"%.2f\" dan nilai kedua \"%.1f\" adalah \"%.2f\"\\n",

29	nilai_pertama2, nilai_kedua2,
30	hasil2);
31	return 0;
32	}

Tabel 34. Source Code Jawaban Soal 2 Bahasa Python

1	print("\n==== INPUT TEST CASE PERTAMA ====")
2	nilai_pertama1 = float(input("Masukkan Nilai Pertama: "))
3	nilai_kedua1 = float(input("Masukkan Nilai Kedua: "))
4	
5	hasil1 = nilai_pertama1 + nilai_kedua1
6	
7	print("\n==== OUTPUT TEST CASE PERTAMA ====")
8	print(f'Hasil dari penjumlahan nilai pertama {int(nilai_pertama1)}" dan nilai kedua {nilai_kedua1:.1f}" adalah "{hasil1:.2f}"')
9	
10	print("\n==== INPUT TEST CASE KEDUA ====")
11	nilai_pertama2 = float(input("Masukkan Nilai Pertama: "))
12	nilai_kedua2 = float(input("Masukkan Nilai Kedua: "))
13	
14	hasil2 = nilai_pertama2 + nilai_kedua2
15	
16	print("\n==== OUTPUT TEST CASE KEDUA ====")
17	print(f'Hasil dari penjumlahan nilai pertama {nilai_pertama2:.2f}" dan nilai kedua {nilai_kedua2:.1f}" adalah "{hasil2:.2f}"')

B. Output Program



```

===== INPUT TEST CASE PERTAMA =====
Masukkan Nilai Pertama: 14
Masukkan Nilai Kedua: 20.5

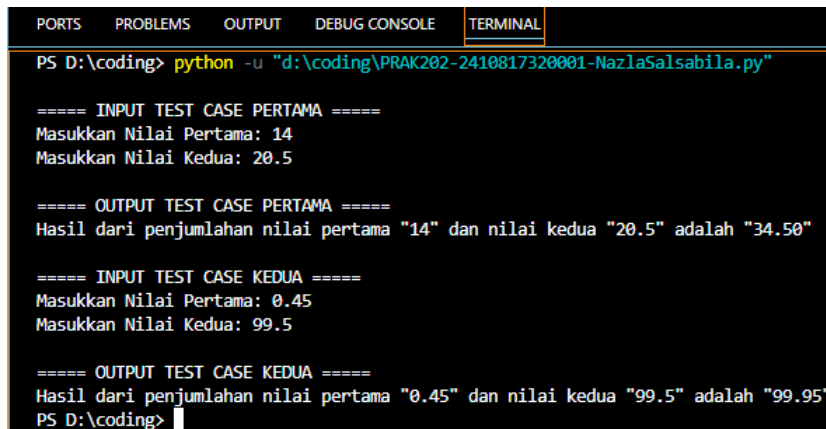
===== OUTPUT TEST CASE PERTAMA =====
Hasil dari penjumlahan nilai pertama "14" dan nilai kedua "20.5" adalah "34.50"

===== INPUT TEST CASE KEDUA =====
Masukkan Nilai Pertama: 0.45
Masukkan Nilai Kedua: 99.5

===== OUTPUT TEST CASE KEDUA =====
Hasil dari penjumlahan nilai pertama "0.45" dan nilai kedua "99.5" adalah "99.95"
PS D:\coding>

```

Gambar 33. Screenshot Output Soal 2 Bahasa C



```

PS D:\coding> python -u "d:\coding\PRAK202-2410817320001-NazlaSalsabila.py"

===== INPUT TEST CASE PERTAMA =====
Masukkan Nilai Pertama: 14
Masukkan Nilai Kedua: 20.5

===== OUTPUT TEST CASE PERTAMA =====
Hasil dari penjumlahan nilai pertama "14" dan nilai kedua "20.5" adalah "34.50"

===== INPUT TEST CASE KEDUA =====
Masukkan Nilai Pertama: 0.45
Masukkan Nilai Kedua: 99.5

===== OUTPUT TEST CASE KEDUA =====
Hasil dari penjumlahan nilai pertama "0.45" dan nilai kedua "99.5" adalah "99.95"
PS D:\coding>

```

Gambar 34. Screenshot Output Soal 2 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [4 dan 5], terdapat `float` untuk mendefinisikan tiga variabel dengan bentuk bilangan desimal.

Pada baris [7, 8, 10, 15, 16, 19, 20, 22, 27, dan 28], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [7, 15, 16, 19, 27, dan 28], terdapat tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [9, 11, 21, dan 23], terdapat `scanf` yang merupakan perintah untuk membaca input berupa string (teks) dan menyimpan ke variabel yang bersangkutan. Tanda `%f` adalah format untuk menampilkan nilai bertipe float.

Pada baris [13 dan 25], program akan menjumlahkan kedua nilai dan menyimpan hasilnya di variabel `hasil1` dan `hasil2`.

Pada baris [16 dan 28], program akan mencetak output yang menampilkan nilai pertama, nilai kedua, dan hasil penjumlahan dengan format `%.1f` yang menampilkan angka desimal dengan satu angka di belakang koma dan `%.2f` yang menampilkan angka desimal dengan dua angka di belakang koma.

Pada baris [31], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [32], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1, 7, 8, 10, 16, dan 17], terdapat `print` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [1, 7, 10, dan 16], terdapat tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [2, 3, 11, dan 12], terdapat `input()` untuk mengambil input. Pada saat program dijalankan, dan program meminta data, maka harus memasukkan data yang diinginkan dan menekan Enter. Kemudian dikonversi ke tipe float menggunakan `float()` untuk menghasilkan angka desimal.

Pada baris [5 dan 14], nilai pertama dan nilai kedua dijumlahkan, dan hasilnya disimpan dalam variabel `hasil1` dan `hasil2`.

Pada baris [8 dan 17], terdapat f-strings untuk menampilkan hasil dari penjumlahan. Tanda `.1f` untuk menampilkan satu angka di belakang koma dan tanda `.2f` untuk menampilkan dua angka di belakang koma.

SOAL 3

Buatlah program yang dapat menghitung dan menghasilkan output langsung menampilkan berupa **hasil** dari a dikurang b dikali dengan i dibagi j dikurang dengan x ditambah y adalah dengan bahasa Python dan C.

Note : Input yang pertama a, kedua b, ketiga i, ke-empat j, kelima x, dan yang ke-enam y. (secara berurutan) (Ketelitian 3 angka di belakang koma)

Input	Output
20 3 4 12 5 9	-8.333
12 2 10 4 3 14	8.000

Simpan coding anda dengan nama: **PRAK203-NIM-Nama.py** dan **PRAK203-NIM-Nama.c**

A. Source Code

Tabel 35. Source Code Jawaban Soal 3 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	float a, b, i, j, x, y, hasil,
5	perkalian, jumlahXY;
6	printf("\n===== INPUT =====\n");
7	printf("Masukkan nilai a: ");
8	scanf("%f", &a);
9	
10	printf("Masukkan nilai b: ");
11	scanf("%f", &b);
12	
13	printf("Masukkan nilai i: ");

14	<code>scanf("%f", &i);</code>
15	
16	<code>printf("Masukkan nilai j: ");</code>
17	<code>scanf("%f", &j);</code>
18	
19	<code>printf("Masukkan nilai x: ");</code>
20	<code>scanf("%f", &x);</code>
21	
22	<code>printf("Masukkan nilai y: ");</code>
23	<code>scanf("%f", &y);</code>
24	
25	<code>float selisih = a - b;</code>
26	<code>float bagi = i / j;</code>
27	<code>perkalian = selisih * bagi;</code>
28	<code>jumlahXY = x + y;</code>
29	<code>hasil = perkalian - jumlahXY;</code>
30	
31	<code>printf("\n==== OUTPUT =====\n");</code>
32	<code>printf("Hasil dari (a - b) * (i / j) -</code>
33	<code>(x + y) adalah: %.3f\n", hasil);</code>
34	<code>return 0;</code>
35	<code>}</code>

Tabel 36. Source Code Jawaban Soal 3 Bahasa Python

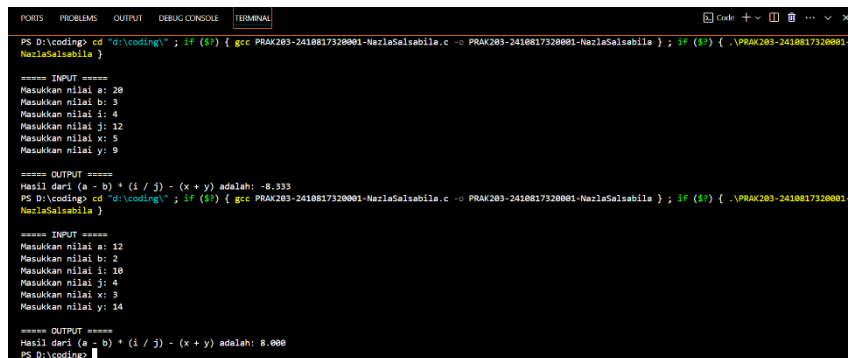
1	<code>print("\n==== INPUT =====");</code>
2	<code>a = float(input("Masukkan nilai a: "))</code>
3	<code>b = float(input("Masukkan nilai b: "))</code>
4	<code>i = float(input("Masukkan nilai i: "))</code>
5	<code>j = float(input("Masukkan nilai j: "))</code>
6	<code>x = float(input("Masukkan nilai x: "))</code>
7	<code>y = float(input("Masukkan nilai y: "))</code>

```

8
9     selisih = a - b
10    bagi = i / j
11    perkalian = selisih * bagi
12    jumlahXY = x + y
13    hasil = perkalian - jumlahXY
14
15    print("\n===== OUTPUT =====");
16    print(f"Hasil dari (a - b) * (i / j) - (x + y) adalah: {hasil:.3f}")

```

B. Output Program



```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK203-2410817320001-NazlaSalsabila.c -o PRAK203-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK203-2410817320001-NazlaSalsabila }

===== INPUT =====
Masukkan nilai a: 20
Masukkan nilai b: 3
Masukkan nilai i: 4
Masukkan nilai j: 12
Masukkan nilai x: 5
Masukkan nilai y: 9

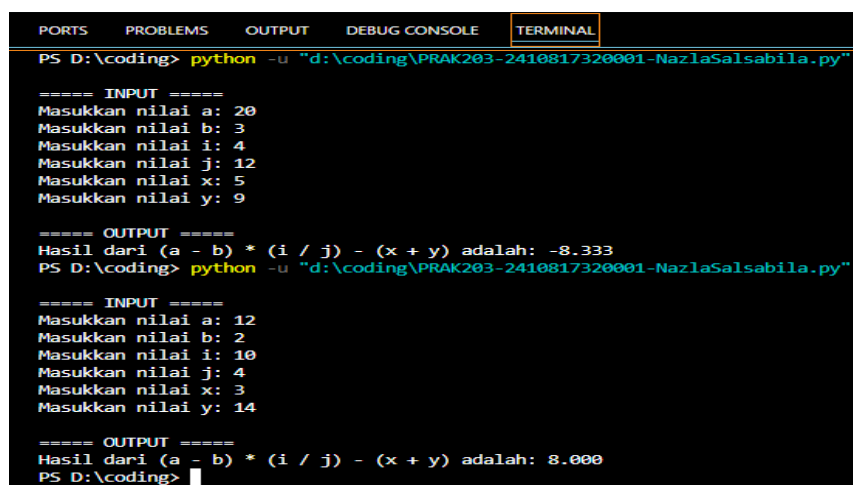
===== OUTPUT =====
Hasil dari (a - b) * (i / j) - (x + y) adalah: -8.333
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK203-2410817320001-NazlaSalsabila.c -o PRAK203-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK203-2410817320001-NazlaSalsabila }

===== INPUT =====
Masukkan nilai a: 12
Masukkan nilai b: 2
Masukkan nilai i: 10
Masukkan nilai j: 4
Masukkan nilai x: 3
Masukkan nilai y: 14

===== OUTPUT =====
Hasil dari (a - b) * (i / j) - (x + y) adalah: 8.000
PS D:\coding>

```

Gambar 35. Screenshot Output Soal 3 Bahasa C



```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> python -u "d:\coding\PRAK203-2410817320001-NazlaSalsabila.py"

===== INPUT =====
Masukkan nilai a: 20
Masukkan nilai b: 3
Masukkan nilai i: 4
Masukkan nilai j: 12
Masukkan nilai x: 5
Masukkan nilai y: 9

===== OUTPUT =====
Hasil dari (a - b) * (i / j) - (x + y) adalah: -8.333
PS D:\coding> python -u "d:\coding\PRAK203-2410817320001-NazlaSalsabila.py"

===== INPUT =====
Masukkan nilai a: 12
Masukkan nilai b: 2
Masukkan nilai i: 10
Masukkan nilai j: 4
Masukkan nilai x: 3
Masukkan nilai y: 14

===== OUTPUT =====
Hasil dari (a - b) * (i / j) - (x + y) adalah: 8.000
PS D:\coding>

```

Gambar 36. Screenshot Output Soal 3 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `float` untuk menyimpan input serta hasil dari operasi matematika. `a`, `b`, `i`, `j`, `x`, dan `y` adalah variabel yang akan diisi dengan input. `hasil`, `perkalian`, dan `jumlahXY` adalah variabel yang digunakan untuk menyimpan hasil sementara dari perhitungan.

Pada baris [6 dan 31], terdapat tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [6, 7, 10, 13, 16, 19, 22, 31, dan 32], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [8, 11, 14, 17, 20, dan 23], terdapat `scanf` yang merupakan perintah untuk membaca input berupa string (teks) dan menyimpannya dalam variabel `a`, `b`, `i`, `j`, `x`, dan `y`. Tanda `%f` untuk membaca input berupa angka desimal (float).

Pada baris [25], terdapat `selisih = a - b` untuk Menghitung selisih antara nilai `a` dan `b`.

Pada baris [26], terdapat `bagi = i / j` untuk menghitung hasil pembagian antara `i` dan `j`.

Pada baris [27], terdapat `perkalian = selisih * bagi;` untuk mengalikan hasil selisih dengan hasil pembagian.

Pada baris [28], terdapat $\text{jumlahXY} = x + y$ untuk menjumlahkan nilai x dan y .

Pada baris [29], terdapat $\text{hasil} = \text{perkalian} - \text{jumlahXY}$ untuk menghitung hasil akhir dengan mengurangi hasil perkalian dengan penjumlahan x dan y .

Pada baris [32], terdapat tanda `%,3f` untuk memastikan hasil ditampilkan dengan tiga angka desimal.

Pada baris [34], terdapat `return 0`; mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [35], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1 dan 15], terdapat tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [1, 15, 16], terdapat `print` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [2 – 7], terdapat `float ()` untuk mengonversi input dari string menjadi tipe data float (angka desimal). `input ()` untuk mengambil input. Pada saat program dijalankan, dan program meminta data, maka harus memasukkan data yang diinginkan dan menekan Enter.

Pada baris [9], terdapat $\text{selisih} = a - b$ untuk Menghitung selisih antara nilai a dan b .

Pada baris [10], terdapat $\text{bagi} = i / j$ untuk menghitung hasil pembagian antara i dan j .

Pada baris [11], terdapat $\text{perkalian} = \text{selisih} * \text{bagi}$; untuk mengalikan hasil selisih dengan hasil pembagian.

Pada baris [12], terdapat $\text{jumlahXY} = x + y$ untuk menjumlahkan nilai x dan y .

Pada baris [13], terdapat $\text{hasil} = \text{perkalian} - \text{jumlahXY}$ untuk menghitung hasil akhir dengan mengurangi hasil perkalian dengan penjumlahan x dan y .

Pada baris [16], terdapat f-strings untuk menampilkan nilai variabel. Tanda `%,3f` untuk memastikan hasil ditampilkan dengan tiga angka desimal.

SOAL 4

Hari ini pak dengklek berencana membeli sebuah bejana memakai tutup dan berbentuk tabung di pasar sukagadai. Di pasar, pak Dengklek menemukan banyak bejana yang membuatnya bingung bejana mana yang yang harus dibeli oleh Pak Dengklek. Buatlah program untuk mengetahui volume, luas, dan keliling bejana jika yang diketahui hanya jari- jari dan tinggi bejana tersebut pada bahasa Python dan C,

Note : input pertama adalah jari-jari, dan kedua adalah tinggi bejana. (ketelitian 2 angka dibelakang koma)

Input	Output
7 10	Volume = 1540.00 Luas = 748.00 Keliling = 44.00
10 10	Volume = 3142.86 Luas = 1257.14 Keliling = 62.86

Simpan coding anda dengan nama: **PRAK204-NIM-Nama.py** dan **PRAK204-NIM-Nama.c**

Note: gunakan format specifier untuk menampilkan angka yang ada pada contoh output.

A. Source Code

Tabel 37. Source Code Jawaban Soal 4 Bahasa C

1	#include <stdio.h>
2	#include <math.h>
3	
4	int main() {
5	
6	float r, h;
7	printf("\n===== INPUT =====\n");

8	<code>printf("Masukkan jari-jari bejana: ");</code>
9	<code>scanf("%f", &r);</code>
10	<code>printf("Masukkan tinggi bejana: ");</code>
11	<code>scanf("%f", &h);</code>
12	
13	<code>printf("\n===== OUTPUT =====\n");</code>
14	
15	<code>float pi = 22.0 / 7.0;</code>
16	<code>float volume = pi * r * r * h;</code>
17	
18	<code>float luas = (2 * pi * r * h) + (2 * pi</code>
19	<code>* r * r);</code>
20	<code>float keliling = 2 * pi * r;</code>
21	
22	<code>printf("Volume = %.2f\n", volume);</code>
23	<code>printf("Luas = %.2f\n", luas);</code>
24	<code>printf("Keliling = %.2f\n", keliling);</code>
25	
26	<code>return 0;</code>
27	<code>}</code>

Tabel 38. Source Code Jawaban Soal 4 Bahasa Python

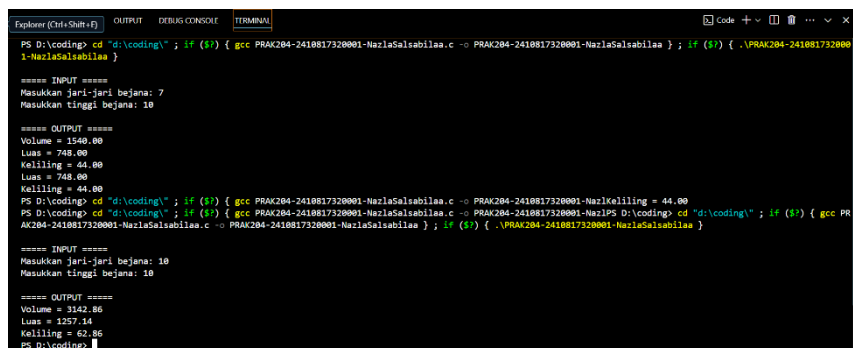
1	<code>pi = 22 / 7</code>
2	
3	<code>print("\n===== INPUT =====")</code>
4	<code>r = float(input("Masukkan jari-jari bejana:"))</code>
5	<code>h = float(input("Masukkan tinggi bejana:"))</code>
6	
7	<code>print("\n===== OUTPUT =====")</code>

```

8
9 volume = pi * r**2 * h
10
11 luas = (2 * pi * r * h) + (2 * pi * r**2)
12
13 keliling = 2 * pi * r
14
15 print(f"Volume = {volume:.2f}")
16 print(f"Luas = {luas:.2f}")
17 print(f"Keliling = {keliling:.2f}")

```

B. Output Program



```

PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK204-2410817320001-NazlaSalsabila.c -o PRAK204-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK204-2410817320001-NazlaSalsabila }

===== INPUT =====
Masukkan jari-jari bejana: 7
Masukkan tinggi bejana: 10

===== OUTPUT =====
Volume = 1540.00
Luas = 748.00
Keliling = 44.00

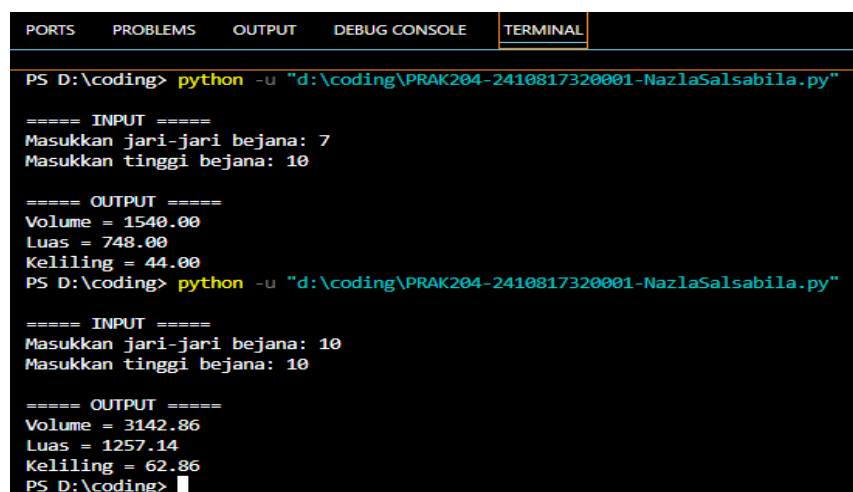
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK204-2410817320001-NazlaSalsabila.c -o PRAK204-2410817320001-NazlaSalsabila }; if ($?) { gcc PR
AK204-2410817320001-NazlaSalsabila.c -o PRAK204-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK204-2410817320001-NazlaSalsabila }

===== INPUT =====
Masukkan jari-jari bejana: 10
Masukkan tinggi bejana: 10

===== OUTPUT =====
Volume = 3142.86
Luas = 1257.14
Keliling = 62.86
PS D:\coding>

```

Gambar 37. Screenshot Output Soal 4 Bahasa C



```

PORTS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\coding> python -u "d:\coding\PRAK204-2410817320001-NazlaSalsabila.py"

===== INPUT =====
Masukkan jari-jari bejana: 7
Masukkan tinggi bejana: 10

===== OUTPUT =====
Volume = 1540.00
Luas = 748.00
Keliling = 44.00
PS D:\coding> python -u "d:\coding\PRAK204-2410817320001-NazlaSalsabila.py"

===== INPUT =====
Masukkan jari-jari bejana: 10
Masukkan tinggi bejana: 10

===== OUTPUT =====
Volume = 3142.86
Luas = 1257.14
Keliling = 62.86
PS D:\coding>

```

Gambar 38. Screenshot Output Soal 4 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [2], terdapat `#include <math.h>`. Fungsinya untuk menggunakan fungsi matematika, seperti `M_PI` yang mewakili nilai konstanta π (pi).

Pada baris [4], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [6, 15, 16, 18, dan 20], terdapat `float` untuk mendeklarasikan variabel dengan tipe data floating point. Tipe data `float` di C digunakan untuk menyimpan angka desimal atau bilangan dengan bagian pecahan.

Pada baris [7, 13, 22, 23, dan 25], terdapat tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [7, 8, 10, 13, 22, 23, dan 24], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [9 dan 11], terdapat `scanf` yang merupakan perintah untuk membaca input berupa string (teks) dan menyimpannya dalam variabel. Tanda `%f` memberi tahu `scanf` bahwa input yang dimasukkan adalah bilangan desimal dan harus disimpan ke dalam variabel tipe `float`.

Pada baris [22 – 24], terdapat tanda `%.2f` untuk mencetak angka desimal dengan dua angka dibelakang koma.

Pada baris [26], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [27], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [3 dan 7], terdapat tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [3, 7, 15, 16, dan 17], terdapat `print` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

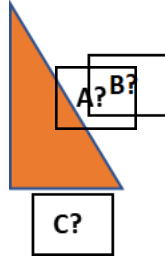
Pada baris [4 dan 5], terdapat `float ()` untuk mengonversi input dari string menjadi tipe data float (angka desimal). `input ()` untuk mengambil input. Pada saat program dijalankan, dan program meminta data, maka harus memasukkan data yang diinginkan dan menekan Enter.

Pada baris [15 – 17], terdapat f-strings untuk menampilkan nilai variabel. Tanda `%.2f` untuk memastikan hasil ditampilkan dengan tiga angka desimal.

SOAL 5

Buatlah sebuah program yang digunakan untuk menghitung alas, tinggi keliling dan luas segitiga pythagoras, Jika yang diketahui hanya A dan B dengan bahasa Python dan C.

Note : Input pertama dan kedua adalah A dan B.



Input	Output
40 41	Alas = 9 cm Tinggi = 40 cm Keliling = 90 cm Luas = 180 cm ²
16 65	Alas = 63 cm Tinggi = 16 cm Keliling = 144 cm Luas = 504 cm ²

Simpan coding anda dengan nama: **PRAK205-NIM-Nama.py** dan **PRAK205-NIM-Nama.c**

A. Source Code

Tabel 39. Source Code Jawaban Soal 5 Bahasa C

1	#include <stdio.h>
2	#include <math.h>
3	
4	double hitung_alas(int A, int B) {


```

5     return sqrt(B * B - A * A);
6 }
7
8 double hitung_keliling(int A, int B, double
9 C) {
10     return A + B + C;
11 }
12
13 double hitung_luas(double C, int A) {
14     return 0.5 * C * A;
15 }
16
17 int main() {
18     int A, B;
19     double C, keliling, luas;
20
21     printf("\n===== INPUT =====\n");
22     printf("Masukkan nilai A (tinggi): ");
23     scanf("%d", &A);
24     printf("Masukkan nilai B (sisi miring): ");
25     scanf("%d", &B);
26
27     if (A <= 0 || B <= 0) {
28         printf("Nilai A dan B harus positif.\n");
29         return 1;
30     }
31
32     if (A >= B) {
33         printf("Nilai B (sisi miring) harus lebih besar dari A (tinggi).\n");

```

33	return 1;
34	}
35	
36	C = hitung_alas(A, B);
37	
38	keliling = hitung_keliling(A, B, C);
39	
40	luas = hitung_luas(C, A);
41	
42	printf("\n==== OUTPUT =====\n");
43	printf("Alas = %.0f cm\n", round(C));
44	printf("Tinggi = %d cm\n", A);
45	printf("Keliling = %.0f cm\n", round(keliling));
46	printf("Luas = %.0f cm^2\n", round(luas));
47	
48	return 0;
49	}

Tabel 40. Source Code Jawaban Soal 5 Bahasa Python

1	import math
2	
3	def hitung_alas(A, B):
4	return math.sqrt(B**2 - A**2)
5	
6	def hitung_keliling(A, B, C):
7	return A + B + C
8	
9	def hitung_luas(C, A):
10	return 0.5 * C * A
11	

```

12 def main():
13     print("\n===== INPUT =====")
14     A = int(input("Masukkan nilai A (tinggi): "))
15     B = int(input("Masukkan nilai B (sisi miring): "))
16
17     if A <= 0 or B <= 0:
18         print("Nilai A dan B harus positif.")
19         return
20
21     if A >= B:
22         print("Nilai B (sisi miring) harus lebih besar dari A (tinggi).")
23         return
24
25     C = hitung_alas(A, B)
26
27     keliling = hitung_keliling(A, B, C)
28
29     luas = hitung_luas(C, A)
30
31     print("\n===== OUTPUT =====")
32     print(f"Alas = {round(C)} cm")
33     print(f"Tinggi = {A} cm")
34     print(f"Keliling = {round(keliling)} cm")
35     print(f"Luas = {round(luas)} cm^2")
36
37     main()

```

B. Output Program

```

PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK205-2410817320001-NazlaSalsabila.c -o PRAK205-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK205-2410817320001-NazlaSalsabila }

===== INPUT =====
Masukkan nilai A (tinggi): 40
Masukkan nilai B (sisi miring): 41

===== OUTPUT =====
Alas = 9 cm
Tinggi = 40 cm
Keliling = 90 cm
Luas = 180 cm^2
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK205-2410817320001-NazlaSalsabila.c -o PRAK205-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK205-2410817320001-NazlaSalsabila }

===== INPUT =====
Masukkan nilai A (tinggi): 16
Masukkan nilai B (sisi miring): 65

===== OUTPUT =====
Alas = 63 cm
Tinggi = 16 cm
Keliling = 144 cm
Luas = 504 cm^2
PS D:\coding>

```

Gambar 39. Screenshot Output Soal 5 Bahasa C

```

===== INPUT =====
Masukkan nilai A (tinggi): 40
Masukkan nilai B (sisi miring): 41

===== OUTPUT =====
Alas = 9 cm
Tinggi = 40 cm
Keliling = 90 cm
Luas = 180 cm^2
PS D:\coding> python -u "d:\coding\PRAK205-2410817320001-NazlaSalsabila.py"

===== INPUT =====
Masukkan nilai A (tinggi): 16
Masukkan nilai B (sisi miring): 65

===== OUTPUT =====
Alas = 63 cm
Tinggi = 16 cm
Keliling = 144 cm
Luas = 504 cm^2
PS D:\coding>

```

Gambar 40. Screenshot Output Soal 5 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [2], terdapat `#include <math.h>`. Fungsinya untuk menggunakan fungsi matematika, seperti `sqrt()` untuk menghitung akar kuadrat dan `round()` untuk membulatkan angka.

Pada baris [4, 8, 12, dan 18], untuk mendefinisikan variabel.

Pada baris [5, 9, 13, 28, dan 33], terdapat `return` untuk mengembalikan nilai sebagai angka desimal (tipe `double`).

Pada baris [16], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi. `{` kurung kurawal menandai awal blok kode.

Pada baris [20, 21, 23, 27, 32, dan 42 – 46], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [20, 42, 43, 44, 45, dan 46], terdapat tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [26 dan 31], melakukan validasi input untuk memastikan input A dan B adalah nilai positif dan nilai B harus lebih besar dari nilai A.

Pada baris [36, 38, dan 40], terdapat fungsi untuk menghitung alas, keliling, dan luas.

Pada baris [43, 45, dan 46], terdapat tanda `%.0f` untuk mencetak angka desimal tanpa menampilkan angka di belakang koma.

Pada baris [48], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [49], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `import math` yaitu fungsi matematika yang dipakai untuk menghitung akar kuadrat.

Pada baris [3, 6, dan 9], untuk mendefinisikan variabel.

Pada baris [4, 7, 10, 19, dan 23], terdapat `return` untuk mengembalikan nilai sebagai angka desimal (tipe `double`).

Pada baris [13, 18, 22, dan 31 – 35], terdapat `print` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [13 dan 31], terdapat tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [14 dan 15], terdapat `input()` untuk mengambil input tinggi (A) dan sisi miring (B) sebagai bilangan bulat menggunakan `int()`.

Pada baris [17 dan 21], melakukan validasi input untuk memastikan input A dan B adalah nilai positif dan nilai B harus lebih besar dari nilai A.

Pada baris [25, 27, dan 29], terdapat fungsi untuk menghitung alas, keliling, dan luas.

Pada baris [32 – 35], terdapat f-strings untuk menampilkan nilai variabel.

Pada baris [37], terdapat `main()` berfungsi memanggil fungsi utama `main()` untuk menjalankan program.

MODUL 3: KONDISIONAL

SOAL 1

Buatlah sebuah program untuk mengurutkan 3 angka inputan dari nilai terkecil ke terbesar menggunakan metode kondisional.

Input merupakan angka secara acak dengan output adalah hasil dari pengurutan.

Input	Output
36 12	12 36
5 6	5 6
94 65	65 94

Simpan coding anda dengan nama: **PRAK301-NIM-Nama.py** dan **PRAK301-NIM-Nama.c**

A. Source Code

Tabel 41. Source Code Jawaban Soal 1 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	int a, b, c, temp;
5	
6	printf("Masukkan tiga angka: ");
7	scanf("%d %d %d", &a, &b, &c);
8	
9	if (a > b) {
10	temp = a; a = b; b = temp;
11	}
12	if (a > c) {
13	temp = a; a = c; c = temp;
14	}
15	if (b > c) {

16	temp = b; b = c; c = temp;
17	}
18	
19	printf("Hasil dari pengurutan dari nilai terkecil ke terbesar: %d %d %d\n", a, b, c);
20	
21	return 0;
22	}

Tabel 42. Source Code Jawaban Soal 1 Bahasa Python

1	a, b, c = map(int, input("Masukkan tiga angka: ").split())
2	
3	if a > b:
4	a, b = b, a
5	if a > c:
6	a, c = c, a
7	if b > c:
8	b, c = c, b
9	
10	print(f"Hasil dari pengurutan dari nilai terkecil ke terbesar: {a} {b} {c}")

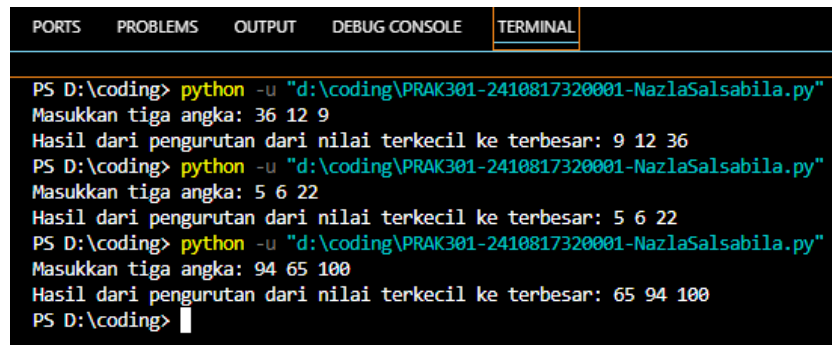
B. Output Program

```

PORTS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK301-2410817320001-NazlaSalsabila.c -o PRAK301-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK301-2410817320001-NazlaSalsabila }
Masukkan tiga angka: 36 12 9
Hasil dari pengurutan dari nilai terkecil ke terbesar: 9 12 36
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK301-2410817320001-NazlaSalsabila.c -o PRAK301-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK301-2410817320001-NazlaSalsabila }
Masukkan tiga angka: 5 6 22
Hasil dari pengurutan dari nilai terkecil ke terbesar: 5 6 22
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK301-2410817320001-NazlaSalsabila.c -o PRAK301-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK301-2410817320001-NazlaSalsabila }
Masukkan tiga angka: 94 65 100
Hasil dari pengurutan dari nilai terkecil ke terbesar: 65 94 100
PS D:\coding>

```

Gambar 41. Screenshot Output Soal 1 Bahasa C



```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> python -u "d:\coding\PRAK301-2410817320001-NazlaSalsabila.py"
Masukkan tiga angka: 36 12 9
Hasil dari pengurutan dari nilai terkecil ke terbesar: 9 12 36
PS D:\coding> python -u "d:\coding\PRAK301-2410817320001-NazlaSalsabila.py"
Masukkan tiga angka: 5 6 22
Hasil dari pengurutan dari nilai terkecil ke terbesar: 5 6 22
PS D:\coding> python -u "d:\coding\PRAK301-2410817320001-NazlaSalsabila.py"
Masukkan tiga angka: 94 65 100
Hasil dari pengurutan dari nilai terkecil ke terbesar: 65 94 100
PS D:\coding>

```

Gambar 42. Screenshot Output Soal 1 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [3, 9, 12, dan 15], terdapat `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `int a, b, c` yaitu tiga variabel untuk menyimpan tiga angka yang diinput. `temp` merupakan variabel sementara untuk membantu proses penukaran nilai.

Pada baris [6 dan 19], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [7], terdapat `scanf` yang merupakan perintah untuk membaca input dan menyimpan ke variabel `a`, `b`, dan `c`.

Pada baris [7 dan 19], terdapat format `%d` untuk menunjukkan inputnya bilangan bulat.

Pada baris [9], terdapat tanda `if (a > b)` yang artinya `a` lebih besar dari `b`.

Pada baris [10], terdapat tanda `temp = a; a = b; b = temp;` yang artinya tukar a dan b, karena angka yang lebih kecil seharusnya berada di a.

Pada baris [12], terdapat tanda `if (a > c)` yang artinya a lebih besar dari c.

Pada baris [13], terdapat tanda `temp = a; a = c; c = temp;` yang artinya tukar a dan c, karena angka yang lebih kecil seharusnya berada di a.

Pada baris [15], terdapat tanda `if (b > c)` yang artinya b lebih besar dari c.

Pada baris [16], terdapat tanda `temp = b; b = c; c = temp;` yang artinya tukar b dan c, karena angka yang lebih kecil seharusnya berada di b.

Pada baris [19], terdapat tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [21], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [11, 14, 17, dan 22], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat a, b, c yaitu tiga variabel untuk menyimpan tiga angka yang diinput. `map(int` untuk mengubah string dan akan diubah menjadi tipe integer (bilangan bulat). `input` fungsinya untuk memasukkan teks (angka). `String()` fungsinya membagi string misal "5 6 22" menjadi tiga bagian string: ["5", "6", "22"].

Pada baris [3], terdapat tanda `if a > b:` yang artinya a lebih besar dari b.

Pada baris [4], terdapat tanda `a, b = b, a` yang artinya tukar a dan b, karena angka yang lebih kecil seharusnya berada di a.

Pada baris [5], terdapat tanda `if a > c:` yang artinya a lebih besar dari c.

Pada baris [6], terdapat tanda `a, c = c, a` yang artinya tukar a dan c, karena angka yang lebih kecil seharusnya berada di a.

Pada baris [7], terdapat tanda `if b > c:` yang artinya b lebih besar dari c.

Pada baris [8], terdapat tanda `b, c = c, b` yang artinya tukar b dan c, karena angka yang lebih kecil seharusnya berada di b.

Pada baris [10], terdapat print untuk menampilkan teks atau format yang ingin ditampilkan ke layar. Fungsi f-string adalah untuk mencetak nilai dari variabel a, b, dan c

SOAL 2

Buatlah sebuah program untuk melakukan konversi nilai seperti pada tabel nilai berikut!

HURUF	NILAI
A	≥ 80
B	70 - 79
C	60 - 69
D	50 - 59
E	< 50

Tanda “-” merepresentasikan kata “sampai”

Input	Output
50	D
75	B
68	C
98	A
49	E

Simpan coding anda dengan nama: **PRAK302-NIM-Nama.py** dan **PRAK302-NIM-Nama.c**

A. Source Code

Tabel 43. Source Code Jawaban Soal 2 Bahasa C

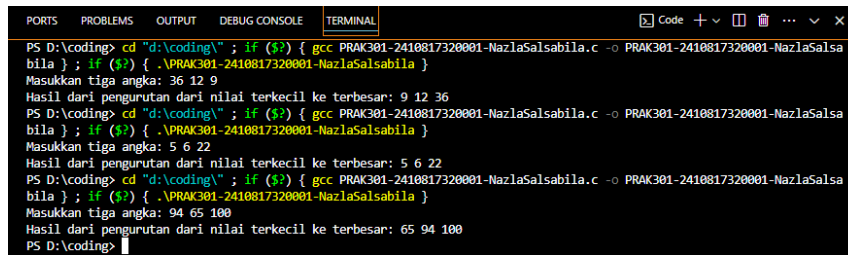
1	#include <stdio.h>
2	
3	int main() {
4	
5	int bil;
6	printf("Masukan angka: ");
7	scanf("%d", &bil);
8	
9	if (bil \geq 80) {

10	printf("nilai huruf adalah A\n");
11	}
12	else if (bil >= 70 && bil < 80) {
13	printf("nilai huruf adalah B\n");
14	}
15	else if (bil >= 60 && bil < 70) {
16	printf("nilai huruf adalah C\n");
17	}
18	else if (bil >= 50 && bil < 60) {
19	printf("nilai huruf adalah D\n");
20	}
21	else {
22	printf("nilai huruf adalah E\n");
23	}
24	
25	return 0;
26	}

Tabel 44. Source Code Jawaban Soal 2 Bahasa Python

1	bil = int(input("Masukan angka: "))
2	
3	if bil >= 80:
4	print("nilai huruf adalah A")
5	elif bil >= 70 and bil < 80:
6	print("nilai huruf adalah B")
7	elif bil >= 60 and bil < 70:
8	print("nilai huruf adalah C")
9	elif bil >= 50 and bil < 60:
10	print("nilai huruf adalah D")
11	else:
12	print("nilai huruf adalah E")

B. Output Program

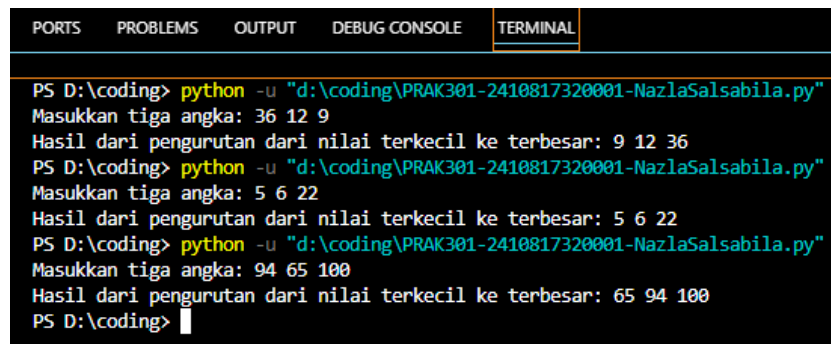


```

PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK301-2410817320001-NazlaSalsabila.c -o PRAK301-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK301-2410817320001-NazlaSalsabila }
Masukkan tiga angka: 36 12 9
Hasil dari pengurutan dari nilai terkecil ke terbesar: 9 12 36
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK301-2410817320001-NazlaSalsabila.c -o PRAK301-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK301-2410817320001-NazlaSalsabila }
Masukkan tiga angka: 5 6 22
Hasil dari pengurutan dari nilai terkecil ke terbesar: 5 6 22
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK301-2410817320001-NazlaSalsabila.c -o PRAK301-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK301-2410817320001-NazlaSalsabila }
Masukkan tiga angka: 94 65 100
Hasil dari pengurutan dari nilai terkecil ke terbesar: 65 94 100
PS D:\coding>

```

Gambar 43. Screenshot Output Soal 2 Bahasa C



```

PS D:\coding> python -u "d:\coding\PRAK301-2410817320001-NazlaSalsabila.py"
Masukkan tiga angka: 36 12 9
Hasil dari pengurutan dari nilai terkecil ke terbesar: 9 12 36
PS D:\coding> python -u "d:\coding\PRAK301-2410817320001-NazlaSalsabila.py"
Masukkan tiga angka: 5 6 22
Hasil dari pengurutan dari nilai terkecil ke terbesar: 5 6 22
PS D:\coding> python -u "d:\coding\PRAK301-2410817320001-NazlaSalsabila.py"
Masukkan tiga angka: 94 65 100
Hasil dari pengurutan dari nilai terkecil ke terbesar: 65 94 100
PS D:\coding>

```

Gambar 44. Screenshot Output Soal 2 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [3, 9, 12, 15, 18, dan 21], terdapat tanda `{` kurung kurawal menandai awal blok kode.

Pada baris [5], terdapat `int bil;` untuk mendeklarasikan variabel `bil` dengan tipe integer dan akan menyimpan angka yang diinputkan.

Pada baris [6, 10, 13, 16, 19, dan 22], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [7], terdapat `scanf` untuk membaca input angka dan menyimpannya ke variabel `bil`. Tanda `%d` untuk membaca bilangan bulat (integer). `bil` adalah tempat angka yang diinputkan akan disimpan

Pada baris [10, 13, 16, 19, dan 22], terdapat tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [9], terdapat `if (bil >= 80)` yang artinya apakah nilai `bil` lebih besar atau sama dengan 80. Jika benar, program mencetak "nilai huruf adalah A"

Pada baris [12], terdapat `else if (bil >= 70 && bil < 80)` yang artinya apakah nilai `bil` berada dalam rentang 70 hingga 79. Jika benar, program mencetak "nilai huruf adalah B". `&&` adalah operator "dan" yang memastikan kedua kondisi harus benar.

Pada baris [15], terdapat `else if (bil >= 60 && bil < 70)` yang artinya apakah nilai `bil` berada dalam rentang 60 hingga 69. Jika benar, program mencetak "nilai huruf adalah C". `&&` adalah operator "dan" yang memastikan kedua kondisi harus benar.

Pada baris [18], terdapat `else if (bil >= 50 && bil < 60)` yang artinya apakah nilai `bil` berada dalam rentang 50 hingga 59. Jika benar, program mencetak "nilai huruf adalah D". `&&` adalah operator "dan" yang memastikan kedua kondisi harus benar.

Pada baris [21], terdapat `else` yang artinya jika semua kondisi sebelumnya salah, maka program akan mencetak "nilai huruf adalah E".

Pada baris [25], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [11, 14, 17, 20, 23, dan 26], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `bil = int(input)`. `int` untuk mengubah string dan akan diubah menjadi tipe integer (bilangan bulat). `input` fungsinya untuk memasukkan teks (angka).

Pada baris [3], terdapat `if bil >= 80`: yang artinya apakah nilai `bil` lebih besar atau sama dengan 80. Jika benar, program mencetak "nilai huruf adalah A".

Pada baris [4, 6, 8, 10, dan 12], terdapat `print` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [5], terdapat `elif bil >= 70 and bil < 80`: yang artinya apakah nilai `bil` berada dalam rentang 70 hingga 79. Jika benar, program mencetak "nilai huruf adalah B". `and` adalah operator "dan" yang memastikan kedua kondisi harus benar.

Pada baris [7], terdapat `elif bil >= 60 and bil < 70`: yang artinya apakah nilai `bil` berada dalam rentang 60 hingga 69. Jika benar, program mencetak "nilai huruf adalah C". `and` adalah operator "dan" yang memastikan kedua kondisi harus benar.

Pada baris [9], terdapat `elif bil >= 50 and bil < 60`: yang artinya apakah nilai `bil` berada dalam rentang 50 hingga 59. Jika benar, program mencetak "nilai huruf adalah D". `and` adalah operator "dan" yang memastikan kedua kondisi harus benar.

Pada baris [11], terdapat `else`: yang artinya jika semua kondisi sebelumnya salah, maka program akan mencetak "nilai huruf adalah E".

SOAL 3

Pak Soni meminta kepada Anda untuk membuat sebuah program sebagai berikut. Program akan menerima sebuah bilangan bulat N. Jika N adalah bilangan bulat positif, cetak positif. Jika N adalah bilangan bulat negatif, cetak negatif. Selain itu (yakni jika N adalah nol), cetak nol.

Input	Output
50	positif
-3000	negatif
0	nol

Simpan coding anda dengan nama: **PRAK303-NIM-Nama.py** dan **PRAK303-NIM-Nama.c**

A. Source Code

Tabel 45. Source Code Jawaban Soal 3 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	int N;
5	
6	printf("Masukkan bilangan bulat: ");
7	scanf("%d", &N);
8	
9	if (N > 0) {
10	printf("positif\n");
11	}
12	else if (N < 0) {
13	printf("negatif\n");
14	}
15	else {
16	printf("nol\n");
17	}

18	
19	return 0;
20	}

Tabel 46. Source Code Jawaban Soal 3 Bahasa Python

1	N = int(input("Masukkan bilangan bulat: "))
2	
3	if N > 0:
4	print("positif")
5	elif N < 0:
6	print("negatif")
7	else:
8	print("nol")

B. Output Program

```

PS D:\coding> cd "d:\coding" ; if ($?) { gcc PRAK303-2410817320001-NazlaSalsabila.c -o PRAK303-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK303-2410817320001-NazlaSalsabila }
Masukkan bilangan bulat: 50
positif
PS D:\coding> cd "d:\coding" ; if ($?) { gcc PRAK303-2410817320001-NazlaSalsabila.c -o PRAK303-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK303-2410817320001-NazlaSalsabila }
Masukkan bilangan bulat: -3000
negatif
PS D:\coding> cd "d:\coding" ; if ($?) { gcc PRAK303-2410817320001-NazlaSalsabila.c -o PRAK303-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK303-2410817320001-NazlaSalsabila }
Masukkan bilangan bulat: 0
nol
PS D:\coding>

```

Gambar 45. Screenshot Output Soal 3 Bahasa C

```

PS D:\coding> python -u "d:\coding\PRAK303-2410817320001-NazlaSalsabila.py"
Masukkan bilangan bulat: 50
positif
PS D:\coding> python -u "d:\coding\PRAK303-2410817320001-NazlaSalsabila.py"
Masukkan bilangan bulat: -3000
negatif
PS D:\coding> python -u "d:\coding\PRAK303-2410817320001-NazlaSalsabila.py"
Masukkan bilangan bulat: 0
nol
PS D:\coding>

```

Gambar 46. Screenshot Output Soal 3 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan

stdio.h adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya printf(). Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. Int menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. Main merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [3, 9, 12, dan 15], terdapat tanda `{` kurung kurawal menandai awal blok kode

Pada baris [4], terdapat `int N;` untuk mendeklarasikan variabel N dengan tipe integer dan akan menyimpan angka yang diinputkan.

Pada baris [6, 10, 13, dan 16], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [7], terdapat `scanf` untuk membaca input angka dan menyimpannya ke variabel `bil`. Tanda `%d` untuk membaca bilangan bulat (integer). `bil` adalah tempat angka yang diinputkan akan disimpan

Pada baris [10, 13, dan 16], terdapat tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [9], terdapat `if (N > 0)` yang artinya apakah nilai N lebih besar dari 0. Jika benar, program mencetak “positif”.

Pada baris [12], terdapat `else if (N < 0)` yang artinya apakah nilai N kurang dari 0. Jika benar, program mencetak "negatif”.

Pada baris [15], terdapat `else` yang artinya jika semua kondisi sebelumnya salah, maka program akan mencetak "nol”.

Pada baris [19], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [11, 14, 17, dan 20], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `N = int(input)`. `int` untuk mengubah string dan akan diubah menjadi tipe integer (bilangan bulat). `input` fungsinya untuk memasukkan teks (angka).

Pada baris [3], terdapat `if N > 0`: yang artinya apakah nilai `N` lebih besar dari 0. Jika benar, program mencetak “positif”.

Pada baris [4, 6, dan 8], terdapat `print` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [5], terdapat `elif N < 0`: yang artinya apakah nilai `N` kurang dari 0. Jika benar, program mencetak "negatif”.

Pada baris [7], terdapat `else` yang artinya jika semua kondisi sebelumnya salah, maka program akan mencetak "nol”.

SOAL 4

Pak Ranzi ingin meminta anda untuk membuatkan program untuk membaca ejaan dari bilangan cacah agar mempermudah pekerjaannya.

Format Masukan :

Sebuah baris berisi satu buah bilangan cacah dan batas maksimal 99. ($a \geq 0 < 100$)

Format Keluaran :

Sebuah baris berisi sebuah bilangan cacah yang merupakan Ejaan dari bilangan tersebut atau batas limit bilangan.

Input	Output
3	Satuan
0	Nol
100	Anda Menginput Melebihi Limit Bilangan
62	Puluhan
13	Belasan

Simpan coding anda dengan nama: **PRAK304-NIM-Nama.py** dan **PRAK304-NIM-Nama.c**

A. Source Code

Tabel 47. Source Code Jawaban Soal 4 Bahasa C

1	#include <stdio.h>
2	
3	int main () {
4	int a;
5	
6	printf("Masukkan bilangan: ");
7	scanf("%d", &a);
8	
9	if (a == 0) {
10	printf("Nol");
11	}

12	else if (a >= 1 && a <= 9) {
13	printf("Satuan");
14	}
	else if (a == 10 a >= 20 && a <= 99)
15	{
16	printf("Puluhan");
17	}
18	else if (a >= 11 && a <= 19) {
19	printf("Belasan");
20	}
21	else {
	printf("Anda Menginput Melebihi
22	Limit Bilangan");
23	}
24	
25	return 0;
26	}

Tabel 48. Source Code Jawaban Soal 4 Bahasa Python

1	a = int(input("Masukkan bilangan: "))
2	
3	if a == 0:
4	print("Nol")
5	elif a >= 1 and a <= 9:
6	print("Satuan")
7	elif a == 10 or a >= 20 and a <= 99:
8	print("Puluhan")
9	elif a >= 11 and a <= 19:
10	print("Belasan")
11	else:
	print("Anda Menginput Melebihi Limit
12	Bilangan")

B. Output Program

```

PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK304-2410817320001-NazlaSalsabila.c -o PRAK304-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK304-2410817320001-NazlaSalsabila }
Masukkan bilangan: 3
Satuan
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK304-2410817320001-NazlaSalsabila.c -o PRAK304-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK304-2410817320001-NazlaSalsabila }
Masukkan bilangan: 0
Nol
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK304-2410817320001-NazlaSalsabila.c -o PRAK304-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK304-2410817320001-NazlaSalsabila }
Masukkan bilangan: 100
Anda Menginput Melebihi Limit Bilangan
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK304-2410817320001-NazlaSalsabila.c -o PRAK304-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK304-2410817320001-NazlaSalsabila }
Masukkan bilangan: 62
Puluhan
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK304-2410817320001-NazlaSalsabila.c -o PRAK304-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK304-2410817320001-NazlaSalsabila }
Masukkan bilangan: 13
Belasan
PS D:\coding>

```

Gambar 47. Screenshot Output Soal 4 Bahasa C

```

PS D:\coding> python -u "d:\coding\PRAK304-2410817320001-NazlaSalsabila.py"
Masukkan bilangan: 3
Satuan
PS D:\coding> python -u "d:\coding\PRAK304-2410817320001-NazlaSalsabila.py"
Masukkan bilangan: 0
Nol
PS D:\coding> python -u "d:\coding\PRAK304-2410817320001-NazlaSalsabila.py"
Masukkan bilangan: 100
Anda Menginput Melebihi Limit Bilangan
PS D:\coding> python -u "d:\coding\PRAK304-2410817320001-NazlaSalsabila.py"
Masukkan bilangan: 62
Puluhan
PS D:\coding> python -u "d:\coding\PRAK304-2410817320001-NazlaSalsabila.py"
Masukkan bilangan: 13
Belasan
PS D:\coding>

```

Gambar 48. Screenshot Output Soal 4 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [3, 9, 12, 15, 18, dan 21], terdapat `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `int a`; untuk mendeklarasikan variabel `a` dengan tipe integer dan akan menyimpan angka yang diinputkan.

Pada baris [6, 10, 13, 16, 19, dan 22], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [7], terdapat `scanf` untuk membaca input angka dan menyimpannya ke variabel `bil`. Tanda `%d` untuk membaca bilangan bulat (integer). `a` adalah tempat angka yang diinputkan akan disimpan

Pada baris [9], terdapat `if (a == 0)` yang artinya apakah nilai `a` sama dengan 0. Jika benar, program mencetak “Nol”.

Pada baris [12], terdapat `else if (a >= 1 && a <= 9)` yang artinya apakah nilai `a` berada dalam rentang 1 hingga 9. Jika benar, program mencetak "Satuan”.

Pada baris [15], terdapat `else if (a == 10 || a >= 20 && a <= 99)` yang artinya apakah nilai `a` adalah 10 atau berada dalam rentang 20 hingga 99. Jika benar, program mencetak “Puluhan”.

Pada baris [18], terdapat `else if (a >= 11 && a <= 19)` yang artinya apakah nilai `a` berada dalam rentang 11 hingga 19. Jika benar, program mencetak "Belasan”.

Pada baris [21], terdapat `else` yang artinya jika semua kondisi sebelumnya salah, maka program akan mencetak "Anda Menginput Melebihi Limit Bilangan”.

Pada baris [25], terdapat `return 0`; mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [11, 14, 17, 20, 23, dan 26], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `a = int(input)`. `int` untuk mengubah string dan akan diubah menjadi tipe integer (bilangan bulat). `input` fungsinya untuk memasukkan teks (angka).

Pada baris [3], terdapat `if a == 0`: yang artinya apakah nilai `a` sama dengan 0. Jika benar, program mencetak “Nol”.

Pada baris [4, 6, 8, 10, dan 12], terdapat print untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [5], terdapat elif $a \geq 1$ and $a \leq 9$ yang artinya apakah nilai a berada dalam rentang 1 hingga 9. Jika benar, program mencetak "Satuan".

Pada baris [7], terdapat elif $a == 10$ or $a \geq 20$ and $a \leq 99$ yang artinya apakah nilai a adalah 10 atau berada dalam rentang 20 hingga 99. Jika benar, program mencetak "Puluhan".

Pada baris [9], terdapat elif $a \geq 11$ and $a \leq 19$ yang artinya apakah nilai a berada dalam rentang 11 hingga 19. Jika benar, program mencetak "Belasan".

Pada baris [11], terdapat else yang artinya jika semua kondisi sebelumnya salah, maka program akan mencetak "Anda Menginput Melebihi Limit Bilangan".

SOAL 5

Buat program untuk mengkonversikan jumlah detik ke dalam jam, menit, dan detik.

Format Masukan :

Sebuah bilangan yang merepresentasikan detik

Format Keluaran :

Sebuah baris berisi angka hasil konfersi jam, menit, dan detik. (dengan format **jam:menit:detik**)

Input	Output
3600	01:00:00
1432	00:23:52
8453	02:20:53
21542	05:59:02
125478	1 hari 10:51:18

Simpan coding anda dengan nama: **PRAK305-NIM-Nama.py** dan **PRAK305-NIM-Nama.c**

A. Source Code

Tabel 49. Source Code Jawaban Soal 5 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	int total_detik, hari, jam, menit,
5	detik;
6	printf("Masukkan jumlah detik: ");
7	scanf("%d", &total_detik);
8	
9	hari = total_detik / 86400;
10	total_detik %= 86400;
11	

12	jam = total_detik / 3600;
13	total_detik %= 3600;
14	
15	menit = total_detik / 60;
16	detik = total_detik % 60;
17	
18	if (hari > 0) {
	printf("%d hari
19	%02d:%02d:%02d\n", hari, jam, menit,
	detik);
20	}
21	else {
	printf("%02d:%02d:%02d\n", jam,
22	menit, detik);
23	}
24	
25	return 0;
26	}

Tabel 50. Source Code Jawaban Soal 5 Bahasa Python

1	total_detik = int(input("Masukkan jumlah detik: "))
2	
3	hari = total_detik // 86400
4	total_detik %= 86400
5	
6	jam = total_detik // 3600
7	total_detik %= 3600
8	
9	menit = total_detik // 60
10	detik = total_detik % 60
11	
12	if hari > 0:

13	<code>print(f"{hari} hari</code>
14	<code>{jam:02}:{menit:02}:{detik:02}")</code>
15	<code>else:</code>
	<code>print(f"{jam:02}:{menit:02}:{detik:02}")</code>

B. Output Program

```

PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK305-2410817320001-NazlaSalsabila.c -o PRAK305-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK305-2410817320001-NazlaSalsabila }
Masukkan jumlah detik: 3600
01:00:00
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK305-2410817320001-NazlaSalsabila.c -o PRAK305-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK305-2410817320001-NazlaSalsabila }
Masukkan jumlah detik: 1432
00:23:52
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK305-2410817320001-NazlaSalsabila.c -o PRAK305-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK305-2410817320001-NazlaSalsabila }
Masukkan jumlah detik: 8453
02:20:53
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK305-2410817320001-NazlaSalsabila.c -o PRAK305-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK305-2410817320001-NazlaSalsabila }
Masukkan jumlah detik: 21542
05:59:02
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK305-2410817320001-NazlaSalsabila.c -o PRAK305-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK305-2410817320001-NazlaSalsabila }
Masukkan jumlah detik: 125478
1 hari 10:51:18
PS D:\coding>

```

Gambar 49. Screenshot Output Soal 5 Bahasa C

```

PS D:\coding> python -u "d:\coding\PRAK305-2410817320001-NazlaSalsabila.py"
Masukkan jumlah detik: 3600
01:00:00
PS D:\coding> python -u "d:\coding\PRAK305-2410817320001-NazlaSalsabila.py"
Masukkan jumlah detik: 1432
00:23:52
PS D:\coding> python -u "d:\coding\PRAK305-2410817320001-NazlaSalsabila.py"
Masukkan jumlah detik: 8453
02:20:53
PS D:\coding> python -u "d:\coding\PRAK305-2410817320001-NazlaSalsabila.py"
Masukkan jumlah detik: 21542
05:59:02
PS D:\coding> python -u "d:\coding\PRAK305-2410817320001-NazlaSalsabila.py"
Masukkan jumlah detik: 125478
1 hari 10:51:18
PS D:\coding>

```

Gambar 50. Screenshot Output Soal 5 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil

dijalankan. Main merupakan fungsi utama saat program dijalankan. () kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [3, 18, dan 21], terdapat { kurung kurawal menandai awal blok kode.

Pada baris [4,] terdapat `int total_detik, hari, jam, menit, detik;` untuk mendeklarasikan variabel dengan tipe integer dan akan menyimpan angka yang diinputkan.

Pada baris [6, 19, dan 22], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [7], terdapat `scanf` untuk membaca input angka dan menyimpannya ke variabel `bil`. Tanda `%d` untuk membaca bilangan bulat (integer). `total_detik` adalah tempat angka yang diinputkan akan disimpan.

Pada baris [9], terdapat variabel `hari` untuk menyimpan jumlah hari.

Pada baris [10], terdapat variabel `total_detik` untuk menyimpan sisa detik setelah hari dihitung.

Pada baris [12], terdapat variabel `jam` untuk menyimpan sisa detik menjadi jam.

Pada baris [13], terdapat variabel `total_detik` untuk menyimpan sisa_detik setelah jam dihitung.

Pada baris [15], terdapat variabel `menit` untuk menyimpan sisa detik yang tersisa setelah jam dihitung menjadi menit.

Pada baris [16], terdapat variabel `detik` untuk menyimpan sisa detik setelah menghitung menit.

Pada baris [18], terdapat `if (hari > 0)` yang artinya output akan mencetak jumlah hari dan waktu dalam format `hari jam:menit:detik`.

Pada baris [19 dan 22], terdapat tanda `%02d` yang artinya angka yang dicetak memiliki 2 digit. Tanda `\n` digunakan untuk membuat baris baru setelah mencetak teks.

Pada baris [21], terdapat `else` yang artinya jika tidak ada hari, maka output hanya mencetak jam, menit, dan detik tanpa menampilkan "hari".

Pada baris [25], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [20, 23, dan 26], terdapat tanda } yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `total_detik = int(input)`. `int` untuk mengubah string dan akan diubah menjadi tipe integer (bilangan bulat). `input` fungsinya untuk memasukkan teks (angka).

Pada baris [3], terdapat variabel `hari` untuk menyimpan jumlah hari.

Pada baris [4], terdapat variabel `total_detik` untuk menyimpan sisa detik setelah hari dihitung.

Pada baris [6], terdapat variabel `jam` untuk menyimpan sisa detik menjadi jam.

Pada baris [7], terdapat variabel `total_detik` untuk menyimpan `sisa_detik` setelah jam dihitung.

Pada baris [9], terdapat variabel `menit` untuk menyimpan sisa detik yang tersisa setelah jam dihitung menjadi menit.

Pada baris [10], terdapat variabel `detik` untuk menyimpan sisa detik setelah menghitung menit.

Pada baris [12], terdapat `if hari > 0:` yang artinya output akan mencetak jumlah hari dan waktu dalam format `hari jam:menit:detik`.

Pada baris [13 dan 15], terdapat `print` untuk menampilkan teks atau format yang ingin ditampilkan ke layar. Fungsi `f-string` adalah untuk mencetak nilai dari variabel.

Pada baris [15], terdapat `else` yang artinya jika tidak ada hari, maka output hanya mencetak jam, menit, dan detik tanpa menampilkan "hari".

MODUL 4: LOOP

SOAL 1

Pak Dengklek merupakan seorang guru TK Suka Berhitung. Hari ini Pak Dengklek mengajarkan murid-muridnya bilangan kelipatan dengan cara bermain. Setiap murid diminta untuk berjejer dan menyebutkan angka sesuai urutannya kecuali bilangan kelipatan yang harus disebut dengan suatu simbol. Misalnya simbol yang harus disebutkan adalah bintang (*) pada kelipatan 3 maka urutan yang disebut pada tiap anak menjadi: 1 2 * 4 5 * dan seterusnya. Buatlah program untuk membantu Pak Dengklek agar permainan dapat berjalan dengan baik jika jumlah anak didiknya adalah 50 anak.

Info:

Input pertama merupakan bilangan kelipatan yang dirubah menjadi simbol

Input kedua merupakan simbol yang akan menggantikan bilangan tersebut

Output merupakan bilangan 1-50 dengan bilangan kelipatan dirubah menjadi simbol

Input	Output
6 *	1 2 3 4 5 * 7 8 9 10 11 * 13 14 15 16 17 * 19 20 21 22 23 * 25 26 27 28 29 * 31 32 33 34 35 * 37 38 39 40 41 * 43 44 45 46 47 * 49 50
3 #	1 2 # 4 5 # 7 8 # 10 11 # 13 14 # 16 17 # 19 20 # 22 23 # 25 26 # 28 29 # 31 32 # 34 35 # 37 38 # 40 41 # 43 44 # 46 47 # 49 50

11 &	1 2 3 4 5 6 7 8 9 10 & 12 13 14 15 16 17 18 19 20 21 & 23 24 25 26 27 28 29 30 31 32 & 34 35 36 37 38 39 40 41 42 43 & 45 46 47 48 49 50
------	--

Simpan coding anda dengan nama: **PRAK401-NIM-Nama.py** dan **PRAK401-NIM-Nama.c**

A. Source Code

Tabel 51. Source Code Jawaban Soal 1 Bahasa C

```

1  #include <stdio.h>
2
3  int main() {
4      int kelipatan, i;
5      char simbol[10];
6      printf("Masukkan bilangan kelipatan
yang dirubah menjadi simbol: ");
7      scanf("%d", &kelipatan);
8      printf("Masukkan simbol yang akan
menggantikan bilangan kelipatan: ");
9      scanf("%s", simbol);
10
11     for (i = 1; i <= 50; i++) {
12         if (i % kelipatan == 0) {
13             printf("%s ", simbol);
14         } else {
15             printf("%d ", i);
16         }
17     }
18     return 0;
19 }
```


Tabel 52. Source Code Jawaban Soal 1 Bahasa Python

1	def main():
	kelipatan = int(input("Masukkan bilangan kelipatan yang dirubah menjadi simbol: "))
2	
	simbol = input("Masukkan simbol yang akan menggantikan bilangan kelipatan: ")
3	
4	
5	for i in range(1, 51):
6	if i % kelipatan == 0:
7	print(simbol, end=" ")
8	else:
9	print(i, end=" ")
10	
11	main()

B. Output Program

Gambar 51. Screenshot Output Soal 1 Bahasa C

Gambar 52. Screenshot Output Soal 1 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan

stdio.h adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya printf(). Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. Int menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. Main merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [3, 11, 12, dan 14], terdapat `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `int kelipatan, i` yaitu dua variabel bertipe integer. `kelipatan` untuk menyimpan bilangan yang akan dijadikan kelipatan dan `i` variabel untuk perulangan.

Pada baris [5], terdapat `char simbol [10]` untuk menyimpan simbol yang akan menggantikan bilangan kelipatan dengan Panjang hingga 9 karakter.

Pada baris [6, 8, 13, dan 15], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [7 dan 9], terdapat `scanf kelipatan` dan `scanf simbol` yang merupakan perintah untuk membaca input dan menyimpannya ke variabel `kelipatan` dan ke array `simbol`.

Pada baris [7 dan 15], terdapat format `%d` untuk menunjukkan inputnya bilangan bulat.

Pada baris [9 dan 13], terdapat format `%s` untuk menunjukkan inputnya berupa string (karakter).

Pada baris [11], terdapat `for (i = 1; i <= 50; i++) {` yang artinya Perulangan `for`: dimulai dari `i = 1`, kemudian akan terus berjalan selama `i` kurang dari atau sama dengan 50. Jika kondisi menjadi salah (contoh: `i > 50`), maka perulangan akan berhenti. Operasi `i++` berarti menambahkan nilai `i` dengan 1.

Pada baris [12], terdapat `if (i % kelipatan == 0) {` yang artinya mengecek apakah `i` habis dibagi dengan `kelipatan`. Jika ya, maka `i` adalah kelipatan dari bilangan yang dimasukkan.

Pada baris [14], terdapat `else` yang artinya bagian ini dijalankan jika kondisi `if` tidak terpenuhi (`i` bukan kelipatan dari bilangan yang dimasukkan).

Pada baris [18], terdapat `return 0`; mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [14, 16, dan 19], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `def main()`: yang artinya mendefinisikan fungsi utama `main`.

Pada baris [2], terdapat `kelipatan = int(input(` yang artinya variabel `kelipatan` menyimpan bilangan yang akan dijadikan kelipatan. `int` digunakan untuk mengubah string yang dimasukkan menjadi bilangan bulat. `input` digunakan untuk menerima input berupa string.

Pada baris [3], terdapat `simbol = input` yang artinya variabel `simbol` untuk menyimpan string yang akan menggantikan angka kelipatan. `input` digunakan untuk menerima input berupa string.

Pada baris [5], terdapat `for i in range(1, 51)`: yang artinya Perulangan `for`: dimulai dari `i = 1`, kemudian `range(1, 51)` untuk membuat daftar angka dari 1 hingga 50. Setiap iterasi, nilai `i` akan berubah mulai dari 1 hingga 50.

Pada baris [6], terdapat `if i % kelipatan == 0`: yang artinya mengecek apakah `i` habis dibagi dengan kelipatan. Jika ya, maka `i` adalah kelipatan dari bilangan yang dimasukkan.

Pada baris [7 dan 9], terdapat `print(simbol, end=" ")` yang artinya jika kondisi `if` terpenuhi, maka program mencetak `simbol` yang dimasukkan, `end=" "` memastikan hasil cetak berada dalam satu baris, dipisahkan dengan spasi dan `print(i, end=" ")` yang artinya jika `i` bukan kelipatan dari bilangan kelipatan, maka program akan mencetak angka `i`, `end=" "` memastikan hasil cetak berada dalam satu baris, dipisahkan dengan spasi.

Pada baris [8], terdapat `else:` yang artinya bagian ini dijalankan jika kondisi `if` tidak terpenuhi (i bukan kelipatan dari bilangan yang dimasukkan).

Pada baris [11], terdapat `main()` yang artinya mengeksekusi seluruh logika program yang telah didefinisikan.

SOAL 2

Hari ini Pak Dengklek tidak bisa mengajar anak didiknya yang ada di TK Suka Berhitung karena sedang sakit sehingga harus digantikan oleh Bu Dengklek. Pak Dengklek berpesan kepada Bu Dengklek agar mengajarkan materi bilangan genap dan ganjil kepada muridnya. Bu Dengklek mempunyai metode agar materi bilangan ganjil genap dapat disampaikan dengan cara yang menyenangkan. Metodenya adalah dengan cara mengurutkan bilangan ganjil dari 1 sampai batas tertentu dan menyebutkan bilangan genapnya secara terbalik.

Buatlah program untuk mempermudah Bu Dengklek menghitung bilangan tersebut!

Info:

Input merupakan batas maksimal dari bilangan yang akan dihitung

Output baris pertama merupakan urutan bilangan ganjil dari 1 sampai batas maksimal

Output baris kedua merupakan urutan bilangan genap dari batas maksimal sampai 2

Input	Output
10	1 3 5 7 9 10 8 6 4 2
25	1 3 5 7 9 11 13 15 17 19 21 23 25 24 22 20 18 16 14 12 10 8 6 4 2
6	1 3 5 6 4 2

Simpan coding anda dengan nama: **PRAK402-NIM-Nama.py** dan **PRAK402-NIM-Nama.c**

A. Source Code

Tabel 53. Source Code Jawaban Soal 2 Bahasa C

1	#include <stdio.h>
2	
3	int main() {

4	int batas, i;
5	printf("Masukkan batas maksimal dari bilangan yang akan dihitung: ");
6	scanf("%d", &batas);
7	
8	for (i = 1; i <= batas; i += 2) {
9	printf("%d ", i);
10	}
11	printf("\n");
12	
13	for (i = (batas % 2 == 0 ? batas : batas - 1); i >= 2; i -= 2) {
14	printf("%d ", i);
15	}
16	printf("\n");
17	
18	return 0;
19	}

Tabel 54. Source Code Jawaban Soal 2 Bahasa Python

1	batas = int(input("Masukkan batas maksimal dari bilangan yang akan dihitung: "))
2	
3	for i in range(1, batas + 1, 2):
4	print(i, end=" ")
5	
6	print()
7	
8	i = batas if batas % 2 == 0 else batas - 1
9	while i >= 2:
10	print(i, end=" ")
11	i -= 2

12	
13	<code>print()</code>

B. Output Program

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK402-2410817320001-NazlaSalsabila.c -o PRAK402-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK402-2410817320001-NazlaSalsabila }
Masukkan batas maksimal dari bilangan yang akan dihitung: 10
1 3 5 7 9
10 8 6 4 2
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK402-2410817320001-NazlaSalsabila.c -o PRAK402-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK402-2410817320001-NazlaSalsabila }
Masukkan batas maksimal dari bilangan yang akan dihitung: 25
1 3 5 7 9 11 13 15 17 19 21 23 25
24 22 20 18 16 14 12 10 8 6 4 2
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK402-2410817320001-NazlaSalsabila.c -o PRAK402-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK402-2410817320001-NazlaSalsabila }
Masukkan batas maksimal dari bilangan yang akan dihitung: 6
1 3 5
6 4 2
PS D:\coding>

```

Gambar 53, Screenshot Output Soal 2 Bahasa C

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> python -u "d:\coding\PRAK402-2410817320001-NazlaSalsabila.py"
Masukkan batas maksimal dari bilangan yang akan dihitung: 10
1 3 5 7 9
10 8 6 4 2
PS D:\coding> python -u "d:\coding\PRAK402-2410817320001-NazlaSalsabila.py"
Masukkan batas maksimal dari bilangan yang akan dihitung: 25
1 3 5 7 9 11 13 15 17 19 21 23 25
24 22 20 18 16 14 12 10 8 6 4 2
PS D:\coding> python -u "d:\coding\PRAK402-2410817320001-NazlaSalsabila.py"
Masukkan batas maksimal dari bilangan yang akan dihitung: 6
1 3 5
6 4 2
PS D:\coding>

```

Gambar 54. Screenshot Output Soal 2 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [3, 8, dan 13], terdapat tanda `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `int batas, i;` untuk mendeklarasikan dua variabel bertipe integer. variabel `batas` akan menyimpan batas maksimal bilangan. Variabel `i`: sebagai variabel loop.

Pada baris [5, 9, 11, 14, dan 16], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar. `"%d "`, `i` artinya mencetak nilai `i` di baris yang sama, dipisahkan dengan spasi. `\n` artinya mencetak baris baru.

Pada baris [6], terdapat `scanf` untuk membaca input angka dan menyimpannya ke variabel `batas`. Tanda `%d` untuk membaca bilangan bulat (integer). Tanda `&` menunjukkan alamat memori dari variabel `batas`.

Pada baris [8], terdapat `for (i = 1; i <= batas; i += 2)` artinya loop pertama untuk mencetak bilangan ganjil. `i = 1`: Inisialisasi variabel `i` dengan nilai 1. `i <= batas`: loop akan terus berjalan selama `i` kurang dari atau sama dengan `batas`. `i += 2`: pada setiap iterasi, `i` bertambah 2, sehingga hanya bilangan ganjil yang tercetak.

Pada baris [13], terdapat `for (i = (batas % 2 == 0 ? batas : batas - 1); i >= 2; i -= 2)` artinya loop kedua untuk mencetak bilangan genap secara terbalik. `i = (batas % 2 == 0 ? batas : batas - 1)`: Jika `batas` genap (`batas % 2 == 0`), mulai dari `batas`. Jika `batas` ganjil, mulai dari `batas - 1` (bilangan genap terbesar di bawah `batas`). `i >= 2`: loop berjalan hingga `i` lebih besar atau sama dengan 2. `i -= 2`: ada setiap iterasi, `i` berkurang 2, sehingga hanya bilangan genap yang tercetak secara terbalik.

Pada baris [18], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [10, 15, dan 19], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `batas = int(input("Masukkan batas maksimal dari bilangan yang akan dihitung: "))` artinya meminta input untuk menentukan batas maksimal bilangan yang akan dihitung. Input diubah menjadi tipe data integer menggunakan `int()`.

Pada baris [3], terdapat `for i in range(1, batas + 1, 2)`: artinya `for` untuk mencetak bilangan ganjil dari 1 hingga batas. `range(1, batas + 1, 2)`: artinya mulai dari 1, kemudian berhenti di batas + 1 (tidak termasuk batas + 1), dan melangkah dengan nilai 2 (menambah setiap iterasi dengan 2 untuk mencetak bilangan ganjil).

Pada baris [4, 6, 10, dan 13], terdapat `print` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [8], terdapat `i = batas if batas % 2 == 0 else batas - 1` artinya jika batas adalah bilangan genap (`batas % 2 == 0`), maka nilai awal `i` adalah batas dan jika batas adalah bilangan ganjil, maka nilai awal `i` adalah batas - 1 (bilangan genap terbesar di bawah atau sama dengan batas).

Pada baris [9], terdapat `while i >= 2`: artinya `while` untuk mencetak bilangan genap dari `i` hingga 2 secara menurun dan kondisi loop akan terus berjalan selama `i >= 2`.

Pada baris [11], terdapat `i -= 2` artinya mengurangi nilai `i` sebesar 2 pada setiap iterasi untuk menghasilkan bilangan genap sebelumnya.

SOAL 3

Pak Samson adalah seorang pengusaha sekaligus programmer ternama di Kota Xam. Pak Samson kesulitan membagi waktu antara membuat program dan mengurus usahanya, lalu Pak Samson menyuruh anda membuat program yang diminta oleh clientnya, permintaannya sebagai berikut : Buatlah program angka yang bersilangan, dan input terdiri dari angka pertama dan angka kedua, setiap angka yang bersilangan dibatasi dengan symbol -. jika angka pertama lebih besar dari angka kedua maka program akan menampilkan dari terkecil ke terbesar dari batasan angka kedua. jika angka kedua lebih besar dari angka pertama maka program akan menampilkan dari terbesar ke terkecil dari batasan angka kedua.

Untuk lebih jelasnya lihat tabel input output berikut :

Input	Output
3 7	3 7 - 4 6 - 5 5 - 6 4 - 7 3
7 3	7 3 - 6 4 - 5 5 - 4 6 - 3 7
95 100	95 100 - 96 99 - 97 98 - 98 97 - 99 96 - 100 95
23 17	23 17 - 22 18 - 21 19 - 20 20 - 19 21 - 18 22 - 17 23

Simpan coding anda dengan nama: **PRAK401-NIM-Nama.py** dan **PRAK401-NIM-Nama.c**

A. Source Code

Tabel 55. Source Code Jawaban Soal 3 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	int angka1, angka2;
5	printf("Masukkan angka pertama dan kedua: ");
6	scanf("%d %d", &angka1, &angka2);
7	

8	if (angka1 < angka2) {
	for (int i = angka1; i <=
9	angka2; i++) {
	printf("%d %d", i, angka2 -
10	(i - angka1));
11	if (i != angka2) {
12	printf(" - ");
13	}
14	}
15	}
16	else {
	for (int i = angka1; i >=
17	angka2; i--) {
	printf("%d %d", i, angka2 +
18	(angka1 - i));
19	if (i != angka2) {
20	printf(" - ");
21	}
22	}
23	}
24	printf("\n");
25	
26	return 0;
27	}

Tabel 56. Source Code Jawaban Soal 3 Bahasa Python

	angka1, angka2 = map(int,
	input("Masukkan angka pertama dan kedua:
1	").split())
2	
3	if angka1 < angka2:
4	for i in range(angka1, angka2 + 1):
	print(i, angka2 - (i - angka1),
5	end=" - " if i != angka2 else "\n")

6	else:
7	for i in range(angka1, angka2 - 1, -
8	1):
	print(i, angka2 + (angka1 - i),
	end=" - " if i != angka2 else "\n")

B. Output Program

```

PS D:\coding> cd "d:\coding"; if ($?) { gcc PRAK403-2410817320001-NazlaSalsabila.c -o PRAK403-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK403-2410817320001-NazlaSalsabila }
Masukkan angka pertama dan kedua: 3 7
3 7 - 4 6 - 5 5 - 6 4 - 7 3
PS D:\coding> cd "d:\coding"; if ($?) { gcc PRAK403-2410817320001-NazlaSalsabila.c -o PRAK403-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK403-2410817320001-NazlaSalsabila }
Masukkan angka pertama dan kedua: 7 3
7 3 - 6 4 - 5 5 - 4 6 - 3 7
PS D:\coding> cd "d:\coding"; if ($?) { gcc PRAK403-2410817320001-NazlaSalsabila.c -o PRAK403-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK403-2410817320001-NazlaSalsabila }
Masukkan angka pertama dan kedua: 95 100
95 100 - 96 99 - 97 98 - 98 97 - 99 96 - 100 95
PS D:\coding> cd "d:\coding"; if ($?) { gcc PRAK403-2410817320001-NazlaSalsabila.c -o PRAK403-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK403-2410817320001-NazlaSalsabila }
Masukkan angka pertama dan kedua: 23 17
23 17 - 22 18 - 21 19 - 20 20 - 19 21 - 18 22 - 17 23
PS D:\coding>

```

Gambar 55. Screenshot Output Soal 3 Bahasa C

```

PS D:\coding> python -u "d:\coding\PRAK403-2410817320001-NazlaSalsabila.py"
Masukkan angka pertama dan kedua: 3 7
3 7 - 4 6 - 5 5 - 6 4 - 7 3
PS D:\coding> python -u "d:\coding\PRAK403-2410817320001-NazlaSalsabila.py"
Masukkan angka pertama dan kedua: 7 3
7 3 - 6 4 - 5 5 - 4 6 - 3 7
PS D:\coding> python -u "d:\coding\PRAK403-2410817320001-NazlaSalsabila.py"
Masukkan angka pertama dan kedua: 95 100
95 100 - 96 99 - 97 98 - 98 97 - 99 96 - 100 95
PS D:\coding> python -u "d:\coding\PRAK403-2410817320001-NazlaSalsabila.py"
Masukkan angka pertama dan kedua: 23 17
23 17 - 22 18 - 21 19 - 20 20 - 19 21 - 18 22 - 17 23
PS D:\coding>

```

Gambar 56. Screenshot Output Soal 3 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil

dijalankan. Main merupakan fungsi utama saat program dijalankan. () kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [3, 8, 9, 11, 17, 19, dan 27], terdapat tanda { kurung kurawal menandai awal blok kode

Pada baris [4], terdapat `int angka1, angka2;` artinya dua variabel integer (angka1 dan angka2) untuk menyimpan angka yang dimasukkan.

Pada baris [5, 10, 12, 18, 20, dan 24], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [6], terdapat `scanf("%d %d", &angka1, &angka2);` artinya membaca dua angka integer dari input pengguna dan menyimpannya ke variabel angka1 dan angka2.

Pada baris [8], terdapat `if (angka1 < angka2) {` artinya mengevaluasi apakah angka pertama lebih kecil daripada angka kedua (`angka1 < angka2`): Jika iya, program menggunakan loop untuk mencetak angka secara naik. Jika tidak program menggunakan loop untuk mencetak angka secara turun.

Pada baris [9], terdapat `for (int i = angka1; i <= angka2; i++) {` artinya iterasi angka mulai dari angka1 hingga angka2.

Pada baris [10], terdapat `angka2 - (i - angka1)` artinya menyesuaikan angka yang ditampilkan di samping i.

Pada baris [11 dan 19], terdapat `if (i != angka2` artinya menambahkan tanda - di antara angka.

Pada baris [16], terdapat `else` yang artinya menjalankan logika jika angka pertama lebih besar dari atau sama dengan angka kedua.

Pada baris [17], terdapat `for (int i = angka1; i >= angka2; i--)` artinya Iterasi angka mulai dari angka1 hingga angka2, dengan langkah mundur.

Pada baris [18], terdapat `angka2 + (angka1 - i)` artinya menyesuaikan angka yang ditampilkan di samping i.

Pada baris [24], terdapat tanda `/n` artinya untuk mencetak baris baru.

Pada baris [26], terdapat `return 0`; mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [13, 14, 15, 21, 22, 23, dan 27], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python

Pada baris [1], terdapat `angka1, angka2 = map(int, input("Masukkan angka pertama dan kedua: ").split())` artinya `angka1` dan `angka2` adalah variabel yang menyimpan dua angka. `map(int, ...)` artinya mengonversi elemen yang dipecah menjadi tipe `int`. `input()` artinya meminta input. `split()` artinya membagi input berdasarkan spasi menjadi dua elemen.

Pada baris [3], terdapat `if angka1 < angka2:` artinya mengecek apakah angka pertama lebih kecil dari angka kedua. Jika iya, maka program akan menjalankan blok `if`.

Pada baris [4], terdapat `for i in range(angka1, angka2 + 1):` artinya menghasilkan deretan angka dari `angka1` hingga `angka2`.

Pada baris [5], terdapat `print(i, angka2 - (i - angka1), end=" - " if i != angka2 else "\n")` artinya menampilkan `i` dan hasil dari `angka2 - (i - angka1)`. `angka2 - (i - angka1):` artinya menghitung angka kedua dikurangi dengan selisih antara `i` dan `angka1`. `end=" - "` jika belum mencapai angka terakhir (`angka2`). `\n` untuk mencetak baris baru.

Pada baris [6], terdapat `else:` artinya jika kondisi `if angka1 < angka2` tidak terpenuhi, maka program akan menjalankan blok di bawah `else`, yaitu iterasi mundur dari `angka1` ke `angka2`.

Pada baris [7], terdapat `for i in range(angka1, angka2 - 1, -1):` artinya menghasilkan deretan angka dari `angka1` ke `angka2` dengan langkah mundur (-1).

Pada baris [8], terdapat `print(i, angka2 + (angka1 - i), end=" - " if i != angka2 else "\n")` artinya menampilkan `i` dan hasil dari `angka2 + (angka1 - i)`. `angka2 + (angka1 - i):` artinya menghitung angka kedua ditambah selisih antara `angka1` dan `i`. `end=" - "` untuk menambahkan tanda " - " di akhir setiap pasangan angka.

SOAL 4

Pa Jack mencari seorang programmer yang bisa membuatkan program kalkulator sederhana untuk dirinya, karena dia kesulitan menghitung hasil penjualannya. Buatlah sebuah kalkulator sederhana yang di minta sesuai dengan yang Pa Jack inginkan dengan output sebagai berikut : **Pilih program**

1. **Penjumlahan**
2. **Pengurangan**
3. **Perkalian**
4. **Pembagian**
5. **Exit**

Masukkan Pilihan :

Masukkan nilai pertama :

Masukkan nilai kedua :

Hasil Pilihan antara NilaiPertama dengan NilaiKedua adalah Hasil

- Selama program belum memasukkan pilihan angka 5, maka program akan terus berjalan.
- Jika memasukkan angka 5 maka selanjutnya program selesai dan tampilkan : Terimakasih, telah menggunakan kalkulator **NAMAANDA**
- Jika memasukkan angka selain dari angka yang tertera mulai ulang programnya dan tampilkan : Input anda salah, silahkan coba lagi

Note : Lebih jelasnya untuk input output lihat dari link:

<https://bit.ly/PenjelasanSoalNo4>

✓ **yang bertanda merah diganti dengan yang sesuai dengan inputan, misal:**

Masukkan Pilihan : 2 , Nilai Pertama : 4 , dan Nilai Kedua : 2 . maka outputnya sebagai berikut = Hasil Pengurangan antara 4.00 dengan 2.00 adalah 2.00

✓ Ketelitian 2 angka dibelakang koma.

Input	Output
Pilih program 1. Penjumlahan 2. Pengurangan 3. Perkalian 4. Pembagian 5. Exit Masukkan Pilihan : 3 Masukkan nilai pertama : 12 Masukkan nilai kedua : 5	Pilih program 1. Penjumlahan 2. Pengurangan 3. Perkalian 4. Pembagian 5. Exit Masukkan Pilihan : 3 Masukkan nilai pertama :12 Masukkan nilai kedua :5 Hasil perkalian antara 12.00 dengan 5.00 adalah 60.00
Pilih program 1. Penjumlahan 2. Pengurangan 3. Perkalian 4. Pembagian 5. Exit Masukkan Pilihan : 13	Pilih program 1. Penjumlahan 2. Pengurangan 3. Perkalian 4. Pembagian 5. Exit Masukkan Pilihan : 13 Input anda salah, silahkan coba lagi

Pilih program 1. Penjumlahan 2. Pengurangan 3. Perkalian 4. Pembagian 5. Exit Masukkan Pilihan : 5	Pilih program 1. Penjumlahan 2. Pengurangan 3. Perkalian 4. Pembagian 5. Exit Masukkan Pilihan : 5 Terimakasih, telah menggunakan kalkulator NAMAANDA
--	---

Simpan coding anda dengan nama: **PRAK404-NIM-Nama.py** dan **PRAK404-NIM-Nama.c**

A. Source Code

Tabel 57. Source Code Jawaban Soal 4 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	int pilihan;
5	float a, b, hasil;
6	while (1) {
	printf("Pilih program:\n1. Penjumlahan\n2. Pengurangan\n3. Perkalian\n4. Pembagian\n5. Exit\nMasukkan Pilihan: ");
7	
8	scanf("%d", &pilihan);
9	if (pilihan == 5) {
	printf("Terimakasih, telah menggunakan kalkulator
10	NAZLASALSABILA\n");
11	break;
	} else if (pilihan < 1 pilihan
12	> 5) {
	printf("Input anda salah, silahkan coba lagi\n");
13	

14	continue;
15	}
	printf("Masukkan nilai pertama:
16	");
17	scanf("%f", &a);
18	printf("Masukkan nilai kedua: ");
19	scanf("%f", &b);
	if (pilihan == 1) hasil = a + b,
20	printf("Hasil penjumlahan %.2f dengan %.2f adalah %.2f\n", a, b, hasil);
	else if (pilihan == 2) hasil = a -
21	b, printf("Hasil pengurangan %.2f dengan %.2f adalah %.2f\n", a, b, hasil);
	else if (pilihan == 3) hasil = a *
22	b, printf("Hasil perkalian %.2f dengan %.2f adalah %.2f\n", a, b, hasil);
	else if (pilihan == 4) b == 0 ?
23	printf("Pembagian dengan nol tidak diperbolehkan!\n") : (hasil = a / b, printf("Hasil pembagian %.2f dengan %.2f adalah %.2f\n", a, b, hasil));
24	}
25	return 0;
26	}

Tabel 58. Source Code Jawaban Soal 4 Bahasa Python

1	def kalkulator():
2	nama = "NAZLASALSABILA"
3	while True:
	print("Pilih program:\n1.
	Penjumlahan\n2. Pengurangan
4	\n3. Perkalian\n4. Pembagian\n5. Exit")
5	try:
6	pilihan = int(input("Masukkan Pilihan: "))

7	<pre> if pilihan == 5: print(f"Terimakasih, telah menggunakan kalkulator {nama}") break if pilihan not in [1, 2, 3, 10 4]: print("Input anda salah, 11 silahkan coba lagi") continue a, b = float(input("Masukkan nilai pertama: ")), 13 float(input("Masukkan nilai kedua: ")) hasil = a + b if pilihan == 1 else a - b if pilihan == 2 else a * b if pilihan == 3 else (a / b if b != 0 else "Pembagian dengan nol tidak 14 diperbolehkan!") print(f"Hasil {'penjumlahan' if pilihan == 1 else 'pengurangan' if pilihan == 2 else 'perkalian' if pilihan == 3 else 'pembagian'} antara {a:.2f} dengan {b:.2f} adalah {hasil:.2f}" if isinstance(hasil, float) 15 else hasil) except ValueError: 16 print("Input anda salah, 17 silahkan coba lagi") 18 19 kalkulator() </pre>
---	---

B. Output Program

```

PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK404-2410817320001-NazlaSalsabila.c -o PRAK404-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK404-2410817320001-NazlaSalsabila }
Pilih program:
1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian
5. Exit
Masukkan Pilihan: 3
Masukkan nilai pertama: 12
Masukkan nilai kedua: 5
Hasil perkalian 12.00 dengan 5.00 adalah 60.00
Pilih program:
1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian
5. Exit
Masukkan Pilihan: 13
Input anda salah, silahkan coba lagi
Pilih program:
1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian
5. Exit
Masukkan Pilihan: 5
Terimakasih, telah menggunakan kalkulator NAZLASALSABILA
PS D:\coding>

```

Gambar 57. Screenshot Output Soal 4 Bahasa C

```

PS D:\coding> python -u "d:\coding\PRAK404-2410817320001-NazlaSalsabila.py"
Pilih program:
1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian
5. Exit
Masukkan Pilihan: 3
Masukkan nilai pertama: 12
Masukkan nilai kedua: 5
Hasil perkalian antara 12.00 dengan 5.00 adalah 60.00
Pilih program:
1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian
5. Exit
Masukkan Pilihan: 13
Input anda salah, silahkan coba lagi
Pilih program:
1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian
5. Exit
Masukkan Pilihan: 5
Terimakasih, telah menggunakan kalkulator NAZLASALSABILA
PS D:\coding>

```

Gambar 58. Screenshot Output Soal 4 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil

dijalankan. Main merupakan fungsi utama saat program dijalankan. () kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [3, 6, 9, dan 12], terdapat { kurung kurawal menandai awal blok kode.

Pada baris 4, terdapat int pilihan; artinya variabel untuk menyimpan pilihan menu tipe int (integer).

Pada baris [5], terdapat float a, b, hasil; artinya dua bilangan desimal yang diinput untuk operasi (tipe float). variabel hasil untuk menyimpan hasil perhitungan.

Pada baris [6], terdapat while (1) { artinya membuat perulangan tak terbatas hingga memilih untuk keluar (pilihan == 5).

Pada baris [7, 10, 13, 16, dan 18], terdapat printf untuk menampilkan teks atau format yang ingin ditampilkan ke layar. Tanda /n untuk mencetak baris baru.

Pada baris [8, 17, dan 19], terdapat scanf untuk membaca input pilihan, bilangan a, dan bilangan b. Tanda %d untuk membaca bilangan bulat (integer). tanda %f untuk menampilkan nilai bertipe float (bilangan desimal).

Pada baris [9], terdapat if (pilihan == 5) artinya jika pengguna memilih 5, program menampilkan pesan terima kasih dan keluar dari perulangan dengan break.

Pada baris [12], terdapat else if (pilihan < 1 || pilihan > 5) artinya jika input tidak sesuai (di luar angka 1-5), program menampilkan pesan error dan memulai ulang perulangan dengan continue.

Pada baris [20], terdapat if (pilihan == 1) hasil = a + b, printf("Hasil penjumlahan %.2f dengan %.2f adalah %.2f\n", a, b, hasil); artinya menjumlahkan a dan b, kemudian menampilkan hasilnya.

Pada baris [21], terdapat else if (pilihan == 2) hasil = a - b, printf("Hasil pengurangan %.2f dengan %.2f adalah %.2f\n", a, b, hasil); artinya mengurangi a dan b, kemudian menampilkan hasilnya.

Pada baris [22], terdapat else if (pilihan == 3) hasil = a * b, printf("Hasil perkalian %.2f dengan %.2f adalah %.2f\n", a, b, hasil); artinya mengalikan a dan b, kemudian menampilkan hasilnya.

Pada baris [23], terdapat else if (pilihan == 4) b == 0 ? printf("Pembagian dengan nol tidak diperbolehkan!\n") : (hasil = a / b, printf("Hasil pembagian %.2f dengan %.2f

adalah `%.2f\n"`, a, b, hasil)); artinya Mengecek jika `b == 0`. Jika benar, menampilkan pesan error karena pembagian dengan nol tidak valid. Jika salah, menghitung `a / b` dan menampilkan hasilnya.

Pada baris [25], terdapat `return 0`; mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [12, 15, 24, dan 26], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `def kalkulator()`: artinya memasukkan seluruh logika kalkulator dalam sebuah fungsi kalkulator.

Pada baris [2], terdapat `nama = "NAZLASALSABILA"` artinya variabel untuk menyimpan nama kalkulator, yang akan ditampilkan dalam pesan akhir.

Pada baris [3], terdapat `while True`: artinya perulangan tak hingga sehingga program terus berjalan sampai memilih untuk keluar (`pilihan == 5`).

Pada baris [4, 8, 11, 15, dan 17], terdapat `print` untuk menampilkan teks atau format yang ingin ditampilkan ke layar. Tanda `\n` untuk mencetak baris baru.

Pada baris [7], terdapat `if pilihan == 5`: artinya jika memilih 5, program menampilkan pesan terima kasih dan keluar dari perulangan dengan `break`.

Pada baris [10], terdapat `if pilihan not in [1, 2, 3, 4]`: artinya jika input bukan salah satu dari [1, 2, 3, 4], program menampilkan pesan kesalahan dan memulai ulang perulangan dengan `continue`.

Pada baris [13], terdapat `a, b = float(input("Masukkan nilai pertama: ")), float(input("Masukkan nilai kedua: "))` artinya membaca dua bilangan desimal dan menyimpannya dalam variabel a dan b.

Pada baris [14], terdapat `hasil = a + b if pilihan == 1 else a - b if pilihan == 2 else a * b if pilihan == 3 else (a / b if b != 0 else "Pembagian dengan nol tidak`

diperbolehkan!") artinya pilihan == 1: Penjumlahan \rightarrow hasil = $a + b$. pilihan == 2: Pengurangan \rightarrow hasil = $a - b$. pilihan == 3: Perkalian \rightarrow hasil = $a * b$. pilihan == 4: Pembagian \rightarrow jika $b \neq 0$, hasil = a / b . Jika $b == 0$, hasil berisi pesan error "Pembagian dengan nol tidak diperbolehkan!".

Pada baris [16], terdapat `except ValueError`: artinya Jika memasukkan input yang tidak valid (contoh: huruf atau simbol), program akan menampilkan pesan kesalahan.

Pada baris [19], terdapat `kalkulator()` artinya Fungsi utama dipanggil untuk menjalankan program kalkulator.

SOAL 5

Buatlah sebuah program untuk menghitung jumlah kelipatan pada tiap baris seperti contoh kasus.

Info:

input baris pertama, banyaknya n.

input baris kedua, kelipatan.

Output adalah hasil perhitungan dari masing-masing kelipatan pada tiap baris dan output

baris terakhir merupakan jumlah pada masing-masing baris.

Input	Output
3 2	$(1 * 2) = 2$ $(2 * 2) + (1 * 2) = 6$ $(3 * 2) + (2 * 2) + (1 * 2) = 12$ 20
5 3	$(1 * 3) = 3$ $(2 * 3) + (1 * 3) = 9$ $(3 * 3) + (2 * 3) + (1 * 3) = 18$ $(4 * 3) + (3 * 3) + (2 * 3) + (1 * 3) = 30$ $(5 * 3) + (4 * 3) + (3 * 3) + (2 * 3) + (1 * 3) = 45$ 105
2 3	$(1 * 3) = 3$ $(2 * 3) + (1 * 3) = 9$ 12

Simpan coding anda dengan nama: **PRAK405-NIM-Nama.py** dan **PRAK405-NIM-Nama.c**

A. Source Code

Tabel 59. Source Code Jawaban Soal 5 Bahasa C

1	#include <stdio.h>
---	--------------------


```
2
3 void hitung_kelipatan() {
4     int n, kelipatan, total_keseluruhan
5     = 0;
6
7     printf("Masukkan banyaknya n: ");
8     scanf("%d", &n);
9     printf("Masukkan kelipatan: ");
10    scanf("%d", &kelipatan);
11
12    for (int i = 1; i <= n; i++) {
13        int total_baris = 0;
14        printf("(");
15        for (int j = i; j > 0; j--) {
16            int hasil = j * kelipatan;
17            total_baris += hasil;
18            printf("%d * %d", j,
19            kelipatan);
20            if (j > 1) {
21                printf(") + (");
22            }
23            printf(") = %d\n", total_baris);
24            total_keseluruhan +=
25            total_baris;
26        }
27        printf("%d\n", total_keseluruhan);
28    }
29
30    int main() {
31        hitung_kelipatan();
32        return 0;
33    }
```

31	}
----	---

Tabel 60. Source Code Jawaban Soal 5 Bahasa Python

1	def hitung_kelipatan():
2	n = int(input("Masukkan banyaknya n:"))
3	kelipatan = int(input("Masukkan kelipatan: "))
4	total_keseluruhan = 0
5	for i in range(1, n + 1):
6	total_baris = 0
7	ekspresi = ""
8	for j in range(i, 0, -1):
9	hasil = j * kelipatan
10	total_baris += hasil
11	ekspresi += f"({j} * {kelipatan})"
12	if j > 1:
13	ekspresi += " + "
14	print(f"ekspresi = {total_baris}")
15	total_keseluruhan += total_baris
16	print(total_keseluruhan)
17	
18	hitung_kelipatan()

B. Output Program

```

PS D:\coding> cd "d:\coding" ; if ($?) { gcc .\P040405-2430017320001-NuzulGalsabilla.c -o P040405-2430017320001-NuzulGalsabilla } ; if ($?) { .\P040405-2430017320001-NuzulGalsabilla }
Masukkan banyaknya n: 3
Masukkan kelipatan: 2
(1 * 2) +
(2 * 2) + (1 * 2) = 6
(3 * 2) + (2 * 2) + (1 * 2) = 12
12
PS D:\coding> cd "d:\coding" ; if ($?) { gcc .\P040405-2430017320001-NuzulGalsabilla.c -o P040405-2430017320001-NuzulGalsabilla } ; if ($?) { .\P040405-2430017320001-NuzulGalsabilla }
Masukkan banyaknya n: 5
Masukkan kelipatan: 3
(1 * 3) = 3
(2 * 3) + (1 * 3) = 9
(3 * 3) + (2 * 3) + (1 * 3) = 18
(4 * 3) + (1 * 3) + (2 * 3) + (1 * 3) = 30
(5 * 3) + (4 * 3) + (1 * 3) + (2 * 3) + (1 * 3) = 45
45
PS D:\coding> cd "d:\coding" ; if ($?) { gcc .\P040405-2430017320001-NuzulGalsabilla.c -o P040405-2430017320001-NuzulGalsabilla } ; if ($?) { .\P040405-2430017320001-NuzulGalsabilla }
Masukkan banyaknya n: 2
Masukkan kelipatan: 3
(1 * 3) = 3
(2 * 3) + (1 * 3) = 9
9
PS D:\coding>

```

Gambar 59. Screenshot Output Soal 5 Bahasa C

```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\coding> python -u "d:\coding\PRAK405-2410817320001-NazlaSalsabila.py"
Masukkan banyaknya n: 3
Masukkan kelipatan: 2
(1 * 2) = 2
(2 * 2) + (1 * 2) = 6
(3 * 2) + (2 * 2) + (1 * 2) = 12
20
PS D:\coding> python -u "d:\coding\PRAK405-2410817320001-NazlaSalsabila.py"
Masukkan banyaknya n: 5
Masukkan kelipatan: 3
(1 * 3) = 3
(2 * 3) + (1 * 3) = 9
(3 * 3) + (2 * 3) + (1 * 3) = 18
(4 * 3) + (3 * 3) + (2 * 3) + (1 * 3) = 30
(5 * 3) + (4 * 3) + (3 * 3) + (2 * 3) + (1 * 3) = 45
105
PS D:\coding> python -u "d:\coding\PRAK405-2410817320001-NazlaSalsabila.py"
Masukkan banyaknya n: 2
Masukkan kelipatan: 3
(1 * 3) = 3
(2 * 3) + (1 * 3) = 9
12
PS D:\coding>

```

Gambar 60. Screenshot Output Soal 5 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `void hitung_kelipatan()` artinya untuk menghitung dan mencetak deretan operasi kelipatan, serta total keseluruhannya. Tipe `void` berarti fungsi ini tidak mengembalikan nilai.

Pada baris [3, 11, 14, 18, 26, dan 31], terdapat `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `int n, kelipatan, total_keseluruhan = 0;` artinya mendeklarasikan tiga variabel bertipe `int` (integer) yaitu `n`, `kelipatan`, dan `total_keseluruhan`.

Pada baris [6, 8, 13, 17, 19, 22 dan 25], terdapat `printf` untuk menampilkan teks atau format yang ingin ditampilkan ke layar.

Pada baris [7 dan 9], terdapat `scanf` untuk membaca input angka dan menyimpannya ke variabel `bil`. Tanda `%d` untuk membaca bilangan bulat (integer).

Pada baris [11], terdapat `for (int i = 1; i <= n; i++)` artinya perulangan menurun dari `j = i` ke 1.

Pada baris [12], terdapat `int total_baris = 0;` artinya deklarasi variabel `total_baris`.

Pada baris [14], terdapat `for (int j = i; j > 0; j--)` artinya

Pada baris [15], terdapat `j * kelipatan;` artinya menghitung hasil kelipatan.

Pada baris [16], terdapat `total_baris += hasil;` artinya menambahkan hasil kelipatan ke total baris.

Pada baris [18], terdapat `if (j > 1)` artinya mengecek apakah nilai variabel `j` lebih besar dari 1.

Pada baris [23], terdapat `total_keseluruhan += total_baris;` artinya menambahkan hasil total dari setiap baris ke variabel `total_keseluruhan`.

Pada baris [28], terdapat `int main() {`. Int menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. Main merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [29], terdapat `hitung_kelipatan();` artinya memanggil fungsi utama untuk menghitung kelipatan.

Pada baris [30], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [20, 21, 24, 26 dan 31], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `def hitung_kelipatan():` artinya fungsi utama yang akan menjalankan logika program untuk menghitung kelipatan.

Pada baris [2 dan 3], terdapat `n = int(input dan kelipatan = int(input` artinya `n` jumlah baris yang akan dihitung. Kelipatan nilai kelipatan yang digunakan untuk penghitungan. Input untuk membaca input `int` mengonversi input menjadi tipe bilangan bulat.

Pada baris [4], terdapat `total_keseluruhan = 0` artinya menyimpan akumulasi hasil dari semua baris.

Pada baris [5], terdapat `for i in range(1, n + 1):` artinya melakukan iterasi sebanyak `n` baris. `range(1, n + 1)` mencakup angka dari 1 hingga `n`

Pada baris [6], terdapat `total_baris = 0` artinya menyimpan hasil total kelipatan pada baris tertentu.

Pada baris [7], terdapat `ekspresi = ""` artinya string yang digunakan untuk menyimpan format teks perhitungan, seperti `(3 * kelipatan) + (2 * kelipatan) + (1 * kelipatan)`.

Pada baris [8], terdapat `for j in range(i, 0, -1):` artinya melakukan iterasi menurun dari `j = i` hingga `j = 1`.

Pada baris [9], terdapat `j * kelipatan` artinya menghitung nilai kelipatan untuk elemen tersebut.

Pada baris [10], terdapat `total_baris += hasil` artinya menambahkan hasil dari perhitungan `j * kelipatan` ke variabel `total_baris`.

Pada baris [11], terdapat `ekspresi += f"({j} * {kelipatan})"` artinya f-string untuk menyisipkan nilai `j` dan `kelipatan` ke dalam format `(j * kelipatan)`.

Pada baris [12], terdapat `if j > 1:` artinya `j > 1` memastikan tanda `+` hanya ditambahkan antara elemen, bukan setelah elemen terakhir.

Pada baris [15], terdapat `total_keseluruhan += total_baris` artinya `total_baris` berisi jumlah total untuk baris tersebut, kemudian ditampilkan dan ditambahkan ke total keseluruhan.

Pada baris [18], terdapat `hitung_kelipatan()` artinya untuk menghitung dan menampilkan kelipatan berdasarkan input.

MODUL 5: FUNGSI

SOAL 1

Pak Roza merupakan seorang guru SD Selalu Ngoding. Hari ini Pak Roza mengajarkan murid-muridnya tentang angka maksimal dengan cara memperbaiki code yang kurang. Agar lebih efektif Pak Roza menyediakan code nya terlebih dahulu. Jadi Lengkapilah code di bawah ini dan buat dalam bahasa Python nya !

Info:

```
#include <stdio.h>

//Buatlah Function Disini

int main() {
    int a, b, c, d;
    scanf("%d %d %d %d", &a, &b, &c, &d);
    int hasil = MaxBilangan(a, b, c, d);
    printf("%d", hasil);
    return 0;
}
```

Input	Output
1 3 4 2	4
7 5 3 9	9
11 23 51 49	51

Simpan coding anda dengan nama: **PRAK501-NIM-Nama.py** dan **PRAK501-NIM-Nama.c**

A. Source Code

Tabel 61. Source Code Jawaban Soal 1 Bahasa C

1	#include <stdio.h>
2	
3	int MaxBilangan(int a, int b, int c, int d)
4	{
5	int max = a;
6	if (b > max) max = b;
7	if (c > max) max = c;
8	if (d > max) max = d;
9	return max;
10	}
11	int main() {
12	int a, b, c, d;
13	scanf("%d %d %d %d", &a, &b, &c, &d);
14	int hasil = MaxBilangan(a, b, c, d);
15	printf("%d", hasil);
16	
17	return 0;
18	}

Tabel 62. Source Code Jawaban Soal 1 Bahasa Python

1	def MaxBilangan(a, b, c, d):
2	max_value = a
3	if b > max_value:
4	max_value = b
5	if c > max_value:
6	max_value = c
7	if d > max_value:
8	max_value = d
9	return max_value

10	
11	<code>a, b, c, d = map(int, input().split())</code>
12	<code>hasil = MaxBilangan(a, b, c, d)</code>
13	<code>print(hasil)</code>

B. Output Program

```

PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK501-2410817320001-NazlaSalsabila.c -o PRAK501-2410817320001-NazlaSalsabila ; if ($?) { .\PRAK501-2410817320001-NazlaSalsabila }
sabila )
1 3 4 2
4
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK501-2410817320001-NazlaSalsabila.c -o PRAK501-2410817320001-NazlaSalsabila ; if ($?) { .\PRAK501-2410817320001-NazlaSalsabila }
sabila )
7 5 3 9
9
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK501-2410817320001-NazlaSalsabila.c -o PRAK501-2410817320001-NazlaSalsabila ; if ($?) { .\PRAK501-2410817320001-NazlaSalsabila }
sabila )
11 23 51 49
51
PS D:\coding>

```

Gambar 61, Screenshot Output Soal 1 Bahasa C

```

PS D:\coding> python -u "d:\coding\PRAK501-2410817320001-NazlaSalsabila.py"
1 3 4 2
4
PS D:\coding> python -u "d:\coding\PRAK501-2410817320001-NazlaSalsabila.py"
7 5 3 9
9
PS D:\coding> python -u "d:\coding\PRAK501-2410817320001-NazlaSalsabila.py"
11 23 51 49
51
PS D:\coding>

```

Gambar 62. Screenshot Output Soal 1 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int MaxBilangan(int a, int b, int c, int d)` artinya fungsi `MaxBilangan` menerima empat bilangan bulat sebagai input (`a, b, c, d`) dan mengembalikan bilangan bulat sebagai hasil. Fungsi ini juga untuk mencari bilangan terbesar di antara keempat bilangan input.

Pada baris [3 dan 11], terdapat `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `int max = a;` artinya menginisialisasi variabel `max` dengan nilai `a`.

Pada baris [5], terdapat `if (b > max) max = b;` artinya akan dilakukan pengecekan apakah nilai `b` lebih besar dari `max`. Jika benar, nilai `max` diperbarui menjadi `b`.

Pada baris [6], terdapat `if (c > max) max = c;` artinya akan dilakukan pengecekan apakah nilai `c` lebih besar dari `max`. Jika benar, nilai `max` diperbarui menjadi `c`.

Pada baris [7], terdapat `if (d > max) max = d;` artinya akan dilakukan pengecekan apakah nilai `d` lebih besar dari `max`. Jika benar, nilai `max` diperbarui menjadi `d`.

Pada baris [8], terdapat `return max;` artinya mengembalikan nilai `max` sebagai bilangan terbesar.

Pada baris [11], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [12], terdapat `int a, b, c, d;` artinya mendeklarasikan empat variabel `a`, `b`, `c`, `d` untuk menyimpan bilangan bulat yang akan dimasukkan.

Pada baris [13], terdapat `scanf("%d %d %d %d", &a, &b, &c, &d);` artinya membaca empat bilangan bulat dari input dan menyimpan ke dalam variabel `a`, `b`, `c`, `d`. `&a`, `&b`, `&c`, `&d` artinya alamat memori tempat variabel `a`, `b`, `c`, `d` berada.

Pada baris [14], terdapat `int hasil = MaxBilangan(a, b, c, d);` artinya memanggil fungsi `MaxBilangan` `a`, `b`, `c`, `d`, kemudian hasil dari fungsi tersebut disimpan ke variabel `hasil`.

Pada baris [15], terdapat `printf("%d", hasil);` artinya mencetak nilai `hasil` (bilangan terbesar) ke layar.

Pada baris [17], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [9 dan 18], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `def MaxBilangan(a, b, c, d):` artinya fungsi `MaxBilangan` menerima empat bilangan bulat sebagai input (`a, b, c, d`).

Pada baris [2], terdapat `max_value = a` artinya menginisialisasi variabel `max` dengan nilai `a`.

Pada baris [3 dan 4], terdapat `if b > max_value: max_value = b` artinya akan dilakukan pengecekan apakah nilai `b` lebih besar dari `max_value`. Jika benar, nilai `max_value` diperbarui menjadi `b`.

Pada baris [5 dan 6], terdapat `if c > max_value: max_value = c` artinya akan dilakukan pengecekan apakah nilai `c` lebih besar dari `max_value`. Jika benar, nilai `max_value` diperbarui menjadi `c`.

Pada baris [7 dan 8], terdapat `if d > max_value: max_value = d` artinya akan dilakukan pengecekan apakah nilai `d` lebih besar dari `max_value`. Jika benar, nilai `max_value` diperbarui menjadi `d`.

Pada baris [9], terdapat `return max_value` artinya mengembalikan nilai `max_value` sebagai bilangan terbesar.

Pada baris [11], terdapat `a, b, c, d = map(int, input().split())` artinya `a, b, c, d` = menyimpan empat bilangan bulat dari input ke dalam variabel `a, b, c`, dan `d`. `map(int, input().split())`: mengonversi string dari hasil `split()` menjadi bilangan bulat menggunakan fungsi `int`. `input()`: membaca input dalam bentuk string. `split()`: memisahkan input sehingga menghasilkan list dari elemen string.

Pada baris [12], terdapat `hasil = MaxBilangan(a, b, c, d)` artinya memanggil fungsi `MaxBilangan` dengan nilai `a, b, c`, dan `d`, sehingga fungsi akan mengembalikan nilai bilangan terbesar di antara keempat bilangan tersebut.

Pada baris [13], terdapat `print(hasil)` artinya mencetak nilai `hasil` (bilangan terbesar) ke layar.

SOAL 2

Jarak Pulau Samosir adalah jarak dari suatu titik menuju suatu titik lainnya pada suatu sistem koordinat Kartesius dengan menyusuri bagian vertikal dan horizontal, tanpa pernah kembali lagi. Secara sederhana, sama dengan jumlah dari selisih absis dan selisih ordinat. Dengan kata lain, jarak Pulau Samosir = $|x_1 - x_2| + |y_1 - y_2|$.

Pak Ranzi ingin pergi dari koordinat (x_1, y_1) menuju (x_2, y_2) . Tentukan jarak Pulau Samosir yang harus ditempuh Pak Ranzi.

Format Masukan :

Sebuah baris berisi empat buah bilangan bulat x_1 , y_1 , x_2 , dan y_2 .

Format Keluaran :

Sebuah baris berisi sebuah bilangan bulat yang merupakan jarak Pulau Samosir dari kedua titik tersebut.

```
#include <stdio.h>
#include <math.h>

int hitung(int
nilai1, intnilai2){
//Lengkapi Function ini
}

int mutlak(int angka){
//Lengkapi Function ini
}

int main()
{
```

```

int a,b,c,d;

scanf("%d",&a);
scanf("%d",&c);
scanf("%d",&b);
scanf("%d",&d);

Hasil = hitung(a,b) + hitung(c,d);
printf("%d",mutlak(Hasil));

return 0;
}

```

Input	Output
-1 -1 1 1	4
-5 6 -4 2	5
1 2 3 4	4

Simpan coding anda dengan nama: **PRAK502-NIM-Nama.py** dan **PRAK502-NIM-Nama.c**

A. Source Code

Tabel 63. Source Code Jawaban Soal 2 Bahasa C

1	#include <stdio.h>
2	#include <math.h>
3	
4	int hitung(int nilai1, int nilai2) {
5	return nilai1 - nilai2;
6	}
7	
8	int mutlak(int angka) {
9	if (angka < 0) {

10	return -angka;
11	}
12	return angka;
13	}
14	
15	int main()
16	{
17	int a, b, c, d;
18	int Hasil;
19	
20	scanf("%d", &a);
21	scanf("%d", &c);
22	scanf("%d", &b);
23	scanf("%d", &d);
24	
25	Hasil = mutlak(hitung(a,b)) + mutlak(hitung(c,d));
26	printf("%d", mutlak(Hasil));
27	
28	return 0;
29	}

Tabel 64. Source Code Jawaban Soal 2 Bahasa Python

1	def hitung(nilai1, nilai2):
2	return nilai1 - nilai2
3	
4	def mutlak(angka):
5	if angka < 0:
6	return -angka
7	return angka
8	
9	a, c, b, d = map(int, input().split())

10	
11	Hasil = mutlak(hitung(a, b)) +
12	mutlak(hitung(c, d))
13	print(mutlak(Hasil))

B. Output Program

```

PS D:\coding> cd "d:\coding" ; if ($?) { gcc PRAK502-2410817320001-NazlaSalsabila.c -o PRAK502-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK502-2410817320001-NazlaSalsabila }
-1 -1 1 1
4
PS D:\coding> cd "d:\coding" ; if ($?) { gcc PRAK502-2410817320001-NazlaSalsabila.c -o PRAK502-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK502-2410817320001-NazlaSalsabila }
-5 6 -4 2
5
PS D:\coding> cd "d:\coding" ; if ($?) { gcc PRAK502-2410817320001-NazlaSalsabila.c -o PRAK502-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK502-2410817320001-NazlaSalsabila }
1 2 3 4
4
PS D:\coding>

```

Gambar 63. Screenshot Output Soal 2 Bahasa C

```

PS D:\coding> python -u "d:\coding\PRAK502-2410817320001-NazlaSalsabila.py"
-1 -1 1 1
4
PS D:\coding> python -u "d:\coding\PRAK502-2410817320001-NazlaSalsabila.py"
-5 6 -4 2
5
PS D:\coding> python -u "d:\coding\PRAK502-2410817320001-NazlaSalsabila.py"
1 2 3 4
4
PS D:\coding>

```

Gambar 64. Screenshot Output Soal 2 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [2], terdapat `#include <math.h>` artinya untuk melakukan operasi matematika.

Pada baris [4 dan 5], terdapat `int hitung(int nilai1, int nilai2) return nilai1 - nilai2;` artinya deklarasi fungsi bernama `hitung` bertipe integer (`nilai1` dan `nilai2`) dan mengembalikan hasil pengurangan berupa bilangan bulat.

Pada baris [4, 8, 9, dan 16], terdapat tanda { kurung kurawal menandai awal blok kode.

Pada baris [8], terdapat `int mutlak(int angka)` artinya deklarasi fungsi bernama `mutlak` bertipe integer (`angka`) dan mengembalikan nilai `mutlak` dari `angka` tersebut.

Pada baris [9 sampai 14], terdapat `if (angka < 0) { return -angka; } return angka;` artinya jika `angka` negatif, fungsi ini mengubah tanda negatif menjadi positif. Sebaliknya, jika `angka` sudah positif, nilai tetap sama.

Pada baris [15], terdapat `int main() {`. `Int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [17], terdapat `int a, b, c, d;` artinya mendeklarasikan empat variabel integer: `a`, `b`, `c`, dan `d` untuk menyimpan nilai input yang dimasukkan. Pada baris 18, terdapat `int Hasil;` artinya mendeklarasikan variabel `Hasil` bertipe integer untuk menyimpan hasil perhitungan dari operasi pengurangan dan penjumlahan dalam program.

Pada baris [20 sampai 23], terdapat `scanf("%d", &a); scanf("%d", &c); scanf("%d", &b); scanf("%d", &d);` artinya untuk membaca input dan format `%d` menunjukkan bahwa input berupa angka. Nilai yang dimasukkan akan disimpan ke dalam variabel `a`, `b`, `c`, dan `d`.

Pada baris [25], terdapat `Hasil = mutlak(hitung(a,b)) + mutlak(hitung(c,d))` artinya `hitung(a, b)` menghitung perbedaan antara `a` dan `b`, yaitu `a - b`. `mutlak(hitung(a, b))` menghitung nilai `mutlak` dari hasil perbedaan tersebut, yaitu nilai positif dari `a - b`. `hitung(c, d)` menghitung perbedaan antara `c` dan `d`, yaitu `c - d`. `mutlak(hitung(c, d))` menghitung nilai `mutlak` dari hasil perbedaan tersebut, yaitu nilai positif dari `c - d`. Kedua nilai `mutlak` tersebut dijumlahkan dan disimpan dalam variabel `Hasil`.

Pada baris [26], terdapat `printf("%d", mutlak(Hasil));` artinya fungsi `printf` digunakan untuk mencetak output ke layar. `mutlak(Hasil)` menghitung nilai mutlak dari variabel `Hasil`, yang berisi hasil penjumlahan dua nilai mutlak yang telah dihitung sebelumnya. Hasilnya akan ditampilkan berupa angka integer.

Pada baris [28], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [6, 11, 13, dan 29], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `def hitung(nilai1, nilai2):` artinya mengembalikan hasil pengurangan dari `nilai1 - nilai2`.

Pada baris [2], terdapat `return nilai1 - nilai2` artinya mengembalikan hasil pengurangan dari kedua nilai yang diterima oleh fungsi `hitung`.

Pada baris [4], terdapat `def mutlak(angka):` artinya mengembalikan nilai mutlak dari sebuah angka (bilangan bulat).

Pada baris [5 sampai 7], terdapat `if angka < 0: return -angka` `return angka` artinya jika angka negatif, fungsi ini mengubah tanda negatif menjadi positif. Sebaliknya, jika angka sudah positif, nilai tetap sama.

Pada baris [9], terdapat `a, c, b, d = map(int, input().split())` artinya `a, c, b, d =` menyimpan empat bilangan bulat dari input ke dalam variabel `a, c, b, dan d`. `map(int, input().split())`: mengonversi string dari hasil `split()` menjadi bilangan bulat menggunakan fungsi `int`. `input()`: membaca input dalam bentuk string. `split()`: memisahkan input sehingga menghasilkan list dari elemen string.

Pada baris [11], terdapat `Hasil = mutlak(hitung(a, b)) + mutlak(hitung(c, d))` artinya `hitung(a, b)` menghitung perbedaan antara `a` dan `b`. `mutlak(hitung(a, b))` menghitung nilai mutlak dari hasil perbedaan tersebut. `hitung(c, d)` menghitung perbedaan antara `c` dan `d`. `mutlak(hitung(c, d))` menghitung nilai mutlak dari hasil

perbedaan tersebut. Kedua nilai mutlak yang dihitung tersebut dijumlahkan dan disimpan dalam variabel Hasil.

Pada baris [13], terdapat `print(mutlak(Hasil))` artinya fungsi print digunakan untuk mencetak output ke layar. `mutlak(Hasil)` menghitung nilai mutlak dari variabel Hasil, yang berisi hasil penjumlahan dua nilai mutlak yang telah dihitung sebelumnya.

SOAL 3

Pak Roni Seorang master matematika yang sangat handal, Pak Roni menyuruh anda untuk membuatkan program menentukan nilai terbesar dan terkecil. Pak Roni memiliki Angka N buah bilangan bulat. Di antara bilangan-bilangan tersebut, tentukan bilangan terbesar dan bilangan terkecil.

Format Masukan

Baris pertama/awal berisi sebuah bilangan bulat N. Baris setelahnya berisi N buah bilangan bulat.

Format Keluaran

Sebuah baris berisi X Y, dengan X adalah bilangan terbesar dan Y adalah bilangan terkecil.

```
#include <stdio.h>

int maksimal(int a, int b){
    // Lengkapi Function ini
```

```
}

int
minimal(int a,
int b) {
    // Lengkapi Function ini
}

int main(){
    int batas = 0;
    int maks = -100000;
    int minim = 100000;
    int bilangan;
    scanf("%d",
    &bilangan);
```

```

while(batas <
bilangan){
    int nilai; scanf("%d",
&nilai);
    maks =
maksimal(maks, nilai);
    minim =
minimal(minim, nilai);
    batas++;
}
printf("%d%d",ma
ks,minim);
}

```

Input	Output
5 12 34 -5 -3 19	34 -5
8 1 -1 1 10 10 6 8 4	10 -1
10 1 -2 5 7 19 21 -19 12 32 10	32 -19

Simpan coding anda dengan nama: **PRAK503-NIM-Nama.py** dan **PRAK503-NIM-Nama.c**

A. Source Code

Tabel 65. Source Code Jawaban Soal 3 Bahasa C

1	#include <stdio.h>
2	
3	int maksimal(int a, int b) {
4	return (a > b) ? a : b;
5	}

6	
7	<code>int minimal(int a, int b) {</code>
8	<code> return (a < b) ? a : b;</code>
9	<code>}</code>
10	
11	<code>int main() {</code>
12	<code> int batas = 0;</code>
13	<code> int maks = -100000;</code>
14	<code> int minim = 100000;</code>
15	<code> int bilangan;</code>
16	<code> scanf("%d", &bilangan);</code>
17	<code> while (batas < bilangan) {</code>
18	<code> int nilai;</code>
19	<code> scanf("%d", &nilai);</code>
20	<code> maks = maksimal(maks, nilai);</code>
21	<code> minim = minimal(minim, nilai);</code>
22	<code> batas++;</code>
23	<code> }</code>
24	<code> printf("%d %d", maks, minim);</code>
25	
26	<code> return 0;</code>
27	<code>}</code>

Tabel 66. Source Code Jawaban Soal 3 Bahasa Python

1	<code>bilangan = int(input())</code>
2	<code>nilai_list = list(map(int,</code>
3	<code>input().split()))</code>
4	<code>maks = -100000</code>
5	<code>minim = 100000</code>
6	
7	<code>for nilai in nilai_list:</code>

8	<code>maks = max(maks, nilai)</code>
9	<code>minim = min(minim, nilai)</code>
10	
11	<code>print(maks, minim)</code>

B. Output Program

```

PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRACS03-2410017320001-NazlaSalsabila.c -o PRACS03-2410017320001-NazlaSalsabila } ; if ($?) { .\PRACS03-2410017320001-NazlaSalsabila }
5
12 34 -5 -3 19
24 -5
8
1 -1 1 10 10 6 8 4
10 -1
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRACS03-2410017320001-NazlaSalsabila.c -o PRACS03-2410017320001-NazlaSalsabila } ; if ($?) { .\PRACS03-2410017320001-NazlaSalsabila }
8
1 -1 1 10 10 6 8 4
10 -1
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRACS03-2410017320001-NazlaSalsabila.c -o PRACS03-2410017320001-NazlaSalsabila } ; if ($?) { .\PRACS03-2410017320001-NazlaSalsabila }
10
1 -2 5 7 19 21 -19 12 32 10
32 -19
PS D:\coding>

```

Gambar 65, Screenshot Output Soal 3 Bahasa C

```

PS D:\coding> python -u "d:\coding\PRACS03-2410017320001-NazlaSalsabila.py"
5
12 34 -5 -3 19
24 -5
PS D:\coding> python -u "d:\coding\PRACS03-2410017320001-NazlaSalsabila.py"
8
1 -1 1 10 10 6 8 4
10 -1
PS D:\coding> python -u "d:\coding\PRACS03-2410017320001-NazlaSalsabila.py"
10
1 -2 5 7 19 21 -19 12 32 10
32 -19
PS D:\coding>

```

Gambar 66. Screenshot Output Soal 3 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3 dan 4], terdapat `int maksimal(int a, int b) return (a > b) ? a : b;` artinya mengembalikan nilai maksimum dari dua bilangan `a` dan `b`. Jika `a > b`, maka fungsi akan mengembalikan `a`. Jika tidak, fungsi akan mengembalikan `b`.

Pada baris [3, 7, 11, dan 17], terdapat tanda `{` kurung kurawal menandai awal blok kode

Pada baris [7 dan 8], terdapat `int minimal(int a, int b) return (a < b) ? a : b;` artinya Mengembalikan nilai minimum dari dua bilangan a dan b. Jika $a < b$, maka fungsi akan mengembalikan a. Jika tidak, fungsi akan mengembalikan b.

Pada baris [11], terdapat `int main() {`. Int menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. Main merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [12], terdapat `int batas = 0;` artinya variabel untuk melacak jumlah bilangan yang sudah diproses.

Pada baris [13], terdapat `int maks = -100000;` artinya variabel untuk menyimpan nilai maksimum, diinisialisasi dengan nilai yang sangat kecil (-100000).

Pada baris [14], terdapat `int minim = 100000;` artinya variabel untuk menyimpan nilai minimum, diinisialisasi dengan nilai yang sangat besar (100000).

Pada baris [15], terdapat `int bilangan;` artinya variabel untuk menyimpan jumlah bilangan yang akan dimasukkan.

Pada baris [16], terdapat `scanf("%d", &bilangan);` artinya membaca jumlah bilangan yang akan dimasukkan dari input dan menyimpan di bilangan.

Pada baris [17 dan 18], terdapat `while (batas < bilangan) int nilai;` artinya memproses setiap bilangan yang dimasukkan.

Pada baris [19], terdapat `scanf("%d", &nilai);` artinya membaca sebuah bilangan nilai.

Pada baris [20], terdapat `maks = maksimal(maks, nilai);` artinya memperbarui nilai maksimum menggunakan fungsi `maksimal(maks, nilai)`.

Pada baris [21], terdapat `minim = minimal(minim, nilai);` artinya memperbarui nilai minimum menggunakan fungsi `minimal(minim, nilai)`.

Pada baris [22], terdapat `batas++;` artinya meningkatkan nilai batas sebesar 1 untuk mencatat bahwa satu bilangan telah diproses.

Pada baris [24], terdapat `printf("%d %d", maks, minim);` artinya mencetak nilai maksimum (maks) dan nilai minimum (minim) yang diperoleh dari semua bilangan yang dimasukkan.

Pada baris [26], terdapat `return 0`; mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [5, 9, 23, dan 27], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `bilangan = int(input())` artinya membaca jumlah bilangan yang akan diproses dari input. `input()` membaca data sebagai string. `int()` mengonversi string tersebut menjadi bilangan bulat (`int`).

Pada baris [2], terdapat `nilai_list = list(map(int, input().split()))` artinya membaca bilangan yang akan diproses dan menyimpannya dalam sebuah daftar (`list`). `input().split()`: membaca input dan memisahkannya menjadi elemen-elemen daftar string. `map(int, ...)`: mengonversi setiap elemen string dalam daftar menjadi bilangan bulat. `list(...)`: mengubah hasil dari `map()` menjadi sebuah daftar.

Pada baris [4], terdapat `maks = -100000` artinya `maks` diinisialisasi dengan angka yang sangat kecil (`-100000`) agar bisa diperbarui dengan bilangan lebih besar.

Pada baris [5], terdapat `minim = 100000` artinya `minim` diinisialisasi dengan angka yang sangat besar (`100000`) agar bisa diperbarui dengan bilangan lebih kecil.

Pada baris [7], terdapat `for nilai in nilai_list`: artinya menelusuri semua elemen dalam `nilai_list` untuk mencari nilai maksimum dan minimum.

Pada baris [8], terdapat `maks = max(maks, nilai)` artinya memperbarui `maks` dengan nilai maksimum antara `maks` sebelumnya dan nilai saat ini.

Pada baris [9], terdapat `minim = min(minim, nilai)` artinya memperbarui `minim` dengan nilai minimum antara `minim` sebelumnya dan nilai saat ini.

Pada baris [11], terdapat `print(maks, minim)` artinya menampilkan nilai maksimum dan minimum.

SOAL 4

Pa Jack ingin menguji kecerdasan Anda. Anda akan diminta untuk membalik representasi desimal dari beberapa bilangan bulat positif, dengan mengabaikan leading zero. Sebagai contoh, 45670 jika dibalik menjadi 07654; lalu karena leading zero diabaikan, maka akan dianggap menjadi 7654.

Pak Jack akan memberikan Anda dua buah bilangan bulat positif A dan B. Anda diminta untuk membalik representasi desimal kedua bilangan tersebut. Sebut saja hasil pembalikan representasi desimal keduanya sebagai A' dan B'. Kemudian, Anda diminta untuk menjumlahkan A' dan B'. Sebut saja hasil penjumlahannya sebagai C. Terakhir, Anda diminta untuk mencetak hasil pembalikan representasi desimal dari C.

Sebagai contoh, A adalah 1290 dan B adalah 452. Maka, A' dan B' secara berurut adalah 921 dan 254. Hasil penjumlahan A' dan B' adalah C, yaitu $921 + 254 = 1175$. Bilangan yang dicetak adalah pembalikan dari C, yaitu 5711.

```
#include <stdio.h> int reverse(){
// Lengkapi Function ini
}
int main() {
int A, B;
scanf("%d %d",&A,&B);
A=reverse(A); B=reverse(B); int C = A+B;
printf("%d",reverse(C));
}
```


Input	Output
1290 452	5711
5430 1120	655
932 114	56

Simpan coding anda dengan nama: **PRAK504-NIM-Nama.py** dan **PRAK504-NIM-Nama.c**

A. Source Code

Tabel 67. Source Code Jawaban Soal 4 Bahasa C

```

1  #include <stdio.h>
2
3  int reverse(int num) {
4      int reversed = 0;
5      while (num > 0) {
6          reversed = reversed * 10 + num %
7          num /= 10;
8      }
9      return reversed;
10 }
11
12 int main() {
13     int A, B;
14     scanf("%d %d", &A, &B);
15     A = reverse(A);
16     B = reverse(B);
17     int C = A + B;
18     printf("%d", reverse(C));
19
20     return 0;
21 }

```

Tabel 68. Source Code Jawaban Soal 4 Bahasa Python

1	def reverse(x):
2	reversed_num = 0
3	while x > 0:
	reversed_num = reversed_num * 10
4	+ x % 10
5	x //= 10
6	return reversed_num
7	
8	A, B = map(int, input().split())
9	
10	A_rev = reverse(A)
11	B_rev = reverse(B)
12	
13	C = A_rev + B_rev
14	
15	print(reverse(C))

B. Output Program

```

PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK504-2410817320001-NazlaSalsabila.c -o PRAK504-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK504-2410817320001-NazlaSalsabila }
1290 452
5711
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK504-2410817320001-NazlaSalsabila.c -o PRAK504-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK504-2410817320001-NazlaSalsabila }
5430 1120
655
PS D:\coding> cd "d:\coding\"; if ($?) { gcc PRAK504-2410817320001-NazlaSalsabila.c -o PRAK504-2410817320001-NazlaSalsabila }; if ($?) { .\PRAK504-2410817320001-NazlaSalsabila }
932 114
56
PS D:\coding>

```

Gambar 67, Screenshot Output Soal 4 Bahasa C

```

PS D:\coding> python -u "d:\coding\PRAK504-2410817320001-NazlaSalsabila.py"
1290 452
5711
PS D:\coding> python -u "d:\coding\PRAK504-2410817320001-NazlaSalsabila.py"
5430 1120
655
PS D:\coding> python -u "d:\coding\PRAK504-2410817320001-NazlaSalsabila.py"
932 114
56
PS D:\coding>

```

Gambar 68. Screenshot Output Soal 4 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int reverse(int num)` artinya membalik urutan digit dari bilangan bulat positif.

Pada baris [3, 5, dan 12], terdapat `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `int reversed = 0;` artinya variabel ini akan menyimpan hasil bilangan yang telah dibalik.

Pada baris [5 – 7], terdapat `while (num > 0) reversed = reversed * 10 + num % 10; num /= 10;` artinya ambil digit terakhir `num` dengan operasi modulus `% 10`. Tambahkan digit tersebut ke variabel `reversed`, setelah menggeser digit sebelumnya ke kiri (dikalikan 10). Kurangi `num` dengan membagi `num` dengan 10 untuk menghapus digit terakhir.

Pada baris [9], terdapat `return reversed;` artinya mengembalikan hasil bilangan yang telah dibalik.

Pada baris [12], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [13], terdapat `int A, B;` artinya mendeklarasikan dua variabel integer: `A` dan `B`.

Pada baris [14], terdapat `scanf("%d %d", &A, &B);` artinya membaca dua bilangan bulat dari input yang disimpan dalam variabel `A` dan `B`.

Pada baris [15 dan 16], terdapat `A = reverse(A); B = reverse(B);` artinya A dibalik urutannya dan hasilnya disimpan kembali ke A. B dibalik urutannya dan hasilnya disimpan kembali ke B.

Pada baris [17], terdapat `int C = A + B;` artinya menjumlahkan bilangan A dan B yang sudah dibalik dan menyimpannya dalam variabel C.

Pada baris [18], terdapat `printf("%d", reverse(C));` artinya memanggil fungsi reverse untuk membalik hasil penjumlahan C dan mencetak hasil akhir bilangan yang sudah dibalik.

Pada baris [20], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [8, 10, dan 21], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `def reverse(x):` artinya fungsi ini membalikkan digit-digit dari sebuah bilangan.

Pada baris [2], terdapat `reversed_num = 0` artinya menyimpan bilangan hasil pembalikan.

Pada baris [3], terdapat `while x > 0:` artinya proses pembalikan dijalankan.

Pada baris [4], terdapat `reversed_num = reversed_num * 10 + x % 10` artinya ekstraksi digit terakhir dari x menggunakan `x % 10`. Tambahkan digit ini ke `reversed_num` dengan menggeser angka yang sudah ada di `reversed_num` ke kiri (dikalikan 10).

Pada baris [5], terdapat `x //= 10` artinya buang digit terakhir dari x menggunakan `x //= 10` (pembagian bilangan bulat).

Pada baris [6], terdapat `return reversed_num` artinya kembalikan nilai `reversed_num`.

Pada baris [8], terdapat `A, B = map(int, input().split())` artinya membaca dua bilangan dari input dan menyimpannya ke dalam variabel A dan B. `input().split():`

membaca input sebagai string, memisahkannya menjadi dua bagian. `map(int, ...)`: mengonversi kedua bagian string tersebut menjadi bilangan bulat (int).

Pada baris [10 dan 11], terdapat `A_rev = reverse(A)` `B_rev = reverse(B)` artinya memanggil fungsi `reverse` untuk masing-masing bilangan A dan B. Contohnya jika `A = 123`, maka `A_rev = reverse(123) = 321`.

Pada baris [13], terdapat `C = A_rev + B_rev` artinya menjumlahkan bilangan hasil pembalikan.

Pada baris [15], terdapat `print(reverse(C))` artinya memanggil fungsi `reverse` untuk hasil penjumlahan C dan mencetak hasilnya.

SOAL 5

Pak Denni meminta anda untuk melengkapi function berikut supaya programnya bisa dijalankan dengan baik dan benar.

Format Masukkan : yang pertama tahun lahir, yang kedua nama dan yang terakhir asal.

```
#include <stdio.h>

void Biodata( , , ){ int tahun_sekarang = 2020;
// Lengkapi Function ini
}

int main() {
    int tahunLahir;
    char A[20], B[15];
    scanf(" %d",&tahunLahir);
    scanf(" %[^\\n]*c",&A);
    scanf(" %[^\\n]*c",&B);
    Biodata(tahunLahir, Namaku, Asal); return 0;
}
```

Input	Output
2001 Doni Banjarmasin	Perkenalkan Nama Saya : Doni Umur Saya : 19 Saya Adalah Angkatan : 2020 Asal Saya dari : Banjarmasin
2003 Rina Martapura	Perkenalkan Nama Saya : Rina Umur Saya : 17 Saya Adalah Angkatan : 2020 Asal Saya dari : Martapura

Simpan coding anda dengan nama: **PRAK505-NIM-Nama.py** dan **PRAK505-NIM-Nama**.

A. Source Code

Tabel 69. Source Code Jawaban Soal 5 Bahasa C

1	#include <stdio.h>
2	
3	void Biodata(int tahunLahir, char Namaku[], char Asal[]) {
4	int tahun_sekarang = 2020;
5	int umur = tahun_sekarang - tahunLahir;
6	int angkatan = tahun_sekarang;
7	
8	printf("Perkenalkan Nama Saya : %s\n", Namaku);
9	printf("Umur Saya : %d\n", umur);
10	printf("Saya Adalah Angkatan : %d\n", angkatan);
11	printf("Asal Saya dari : %s\n", Asal);
12	}
13	
14	int main() {
15	int tahunLahir;
16	char Namaku[20], Asal[15];
17	scanf("%d",&tahunLahir);
18	scanf(" %[^\\n]*c",&Namaku);
19	scanf(" %[^\\n]*c",&Asal);
20	Biodata(tahunLahir, Namaku, Asal);
21	
22	return 0;
23	}

Tabel 70. Source Code Jawaban Soal 5 Bahasa Python

1	def Biodata(tahunLahir, Namaku, Asal):
2	tahun_sekarang = 2020

```

3      umur = tahun_sekarang - tahunLahir
4      angkatan = tahun_sekarang
5
6      print(f"Perkenalkan Nama Saya :
7      {Namaku}")
8      print(f"Umur Saya : {umur}")
9      print(f"Saya Adalah Angkatan :
10     {angkatan}")
11     print(f"Asal Saya dari : {Asal}")
12
13     tahunLahir = int(input())
14     Namaku = input()
15     Asal = input()
16
17     Biodata(tahunLahir, Namaku, Asal)

```

B. Output Program

```

PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK505-2410817320001-NazlaSalsabila.c -o PRAK505-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK505-2410817320001-NazlaSalsabila }
2001
Doni
Perkenalkan Nama Saya : Doni
Umur Saya : 19
Saya Adalah Angkatan : 2020
Asal Saya dari : Banjarmasin
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK505-2410817320001-NazlaSalsabila.c -o PRAK505-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK505-2410817320001-NazlaSalsabila }
2003
Rina
Perkenalkan Nama Saya : Rina
Umur Saya : 17
Saya Adalah Angkatan : 2020
Asal Saya dari : Martapura
PS D:\coding>

```

Gambar 69, Screenshot Output Soal 5 Bahasa C

```

PS D:\coding> python -u "d:\coding\PRAK505-2410817320001-NazlaSalsabila.py"
2001
Doni
Perkenalkan Nama Saya : Doni
Umur Saya : 19
Saya Adalah Angkatan : 2020
Asal Saya dari : Banjarmasin
PS D:\coding> python -u "d:\coding\PRAK505-2410817320001-NazlaSalsabila.py"
2003
Rina
Perkenalkan Nama Saya : Rina
Umur Saya : 17
Saya Adalah Angkatan : 2020
Asal Saya dari : Martapura
PS D:\coding>

```

Gambar 70. Screenshot Output Soal 5 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `void Biodata(int tahunLahir, char Namaku[], char Asal[])` artinya `tahunLahir` bertipe `int` untuk menyimpan tahun kelahiran. `Namaku[]` bertipe array karakter (`char[]`) untuk menyimpan nama. `Asal[]` bertipe array karakter (`char[]`) untuk menyimpan asal kota atau daerah.

Pada baris [3 dan 14], terdapat `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `int tahun_sekarang = 2020;` artinya mendeklarasikan variabel `tahun_sekarang` dengan nilai 2020. Tahun ini akan digunakan untuk menghitung umur dan angkatan.

Pada baris [5], terdapat `int umur = tahun_sekarang - tahunLahir;` artinya menghitung umur dengan mengurangi tahun kelahiran (`tahunLahir`) dari tahun sekarang (`tahun_sekarang`).

Pada baris [6], terdapat `int angkatan = tahun_sekarang;` artinya menyimpan nilai `tahun_sekarang` ke dalam variabel `Angkatan` yaitu angkatan 2020.

Pada baris [8], terdapat `printf("Perkenalkan Nama Saya : %s\n", Namaku);` artinya untuk mencetak nama yang disimpan dalam array `Namaku`. `%s` adalah format untuk mencetak string.

Pada baris [9], terdapat `printf("Umur Saya : %d\n", umur);` artinya mencetak umur dan `%d` digunakan untuk mencetak nilai integer.

Pada baris [10], terdapat `printf("Saya Adalah Angkatan : %d\n", angkatan);` artinya mencetak nilai angkatan. `\n` adalah format untuk mencetak garis baru.

Pada baris [11], terdapat `printf("Asal Saya dari : %s\n", Asal)` artinya mencetak asal kota atau daerah yang disimpan dalam array `Asal`.

Pada baris [14], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [15 dan 16], terdapat `int tahunLahir;` `char Namaku[20];` `Asal[15];` artinya `tahunLahir` adalah variabel untuk menyimpan tahun kelahiran. `Namaku[20]` adalah array karakter untuk menyimpan nama. `Asal[15]` adalah array karakter untuk menyimpan asal.

Pada baris [17], terdapat `scanf("%d",&tahunLahir);` artinya untuk membaca input. Format `%d` akan memasukkan angka (tahun kelahiran) yang akan disimpan dalam variabel `tahunLahir`.

Pada baris [18], terdapat `scanf(" %[^\\n]*c",&Namaku);` artinya untuk membaca input nama. Format `%[^\\n]*c` membaca seluruh input hingga baris baru (`\\n`) dan `%*c` mengabaikan karakter newline setelahnya.

Pada baris [19], terdapat `scanf(" %[^\\n]*c",&Asal)` artinya untuk membaca input asal.

Pada baris [20], terdapat `Biodata(tahunLahir, Namaku, Asal);` artinya mencetak data yang dimasukkan.

Pada baris [22], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [12 dan 23], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `def Biodata(tahunLahir, Namaku, Asal):` artinya `tahunLahir` untuk menyimpan tahun kelahiran bertipe integer. `Namaku` untuk menyimpan nama bertipe string. `Asal` untuk menyimpan asal kota atau daerah bertipe string.

Pada baris [2], terdapat `tahun_sekarang = 2020` artinya mendeklarasikan variabel `tahun_sekarang` dengan nilai 2020. Tahun ini akan digunakan untuk menghitung umur dan angkatan.

Pada baris [3], terdapat `umur = tahun_sekarang - tahunLahir` artinya menghitung umur dengan mengurangi tahun kelahiran (`tahunLahir`) dari tahun sekarang (`tahun_sekarang`).

Pada baris [4], terdapat `angkatan = tahun_sekarang` artinya menyimpan nilai `tahun_sekarang` ke dalam variabel `Angkatan` yaitu angkatan 2020.

Pada baris [6], terdapat `print(f"Perkenalkan Nama Saya : {Namaku}")` artinya untuk menampilkan string ke layar. f-string (`f"..."`) untuk memformat string dengan memasukkan nilai dari variabel `Namaku`.

Pada baris [7], terdapat `print(f"Umur Saya : {umur}")` artinya mencetak umur yang dihitung sebelumnya. Umur adalah hasil pengurangan tahun sekarang dan tahun kelahiran.

Pada baris [8], terdapat `print(f"Saya Adalah Angkatan : {angkatan}")` artinya mencetak angkatan yaitu memakai tahun sekarang (2020).

Pada baris [9], terdapat `print(f"Asal Saya dari : {Asal}")` artinya mencetak asal yang diinputkan.

Pada baris [11], terdapat `tahunLahir = int(input())` artinya membaca input untuk tahun kelahiran. Fungsi `input()` membaca input sebagai string dan `int()` mengonversi input menjadi integer.

Pada baris [12], terdapat `Namaku = input()` artinya Membaca input nama dan menyimpannya dalam variabel `Namaku`.

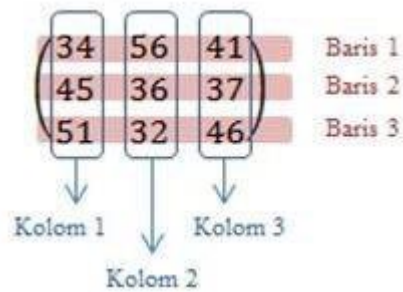
Pada baris [13], terdapat `Asal = input()` artinya membaca input asal kota atau daerah dan menyimpannya dalam variabel `Asal`.

Pada baris [15], terdapat `Biodata(tahunLahir, Namaku, Asal)` artinya mencetak data yang dimasukkan.

MODUL 6: ARRAY

SOAL 1

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



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

Format input:

Input pada baris pertama berupa jumlah baris dan kolom.

Input pada baris kedua berupa angka yang ada di dalam matriks tersebut.

Jumlah input pada baris kedua = (baris x kolom) yang dipisahkan dengan spasi.

Format output:

Output merupakan sebuah matriks .

Input	Output
2 3	1 2 3
1 2 3 4 5 6	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

Simpan coding anda dengan nama: **PRAK601-NIM-Nama.py** dan **PRAK5601-NIM-Nama**.

A. Source Code

Tabel 71. Source Code Jawaban Soal 1 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	int baris, kolom;
5	printf("Masukkan jumlah baris dan
6	kolom: ");
7	scanf("%d %d", &baris, &kolom);
8	int data[baris * kolom];
9	printf("Masukkan angka-angka dalam
10	matriks: ");
11	for (int i = 0; i < baris * kolom;
12	i++) {
13	scanf("%d", &data[i]);
14	}
15	int index = 0;
16	for (int i = 0; i < baris; i++) {
17	for (int j = 0; j < kolom; j++) {
	printf("%d", data[index++]);

18	if (j < kolom - 1) printf("
19	");
20	}
21	printf("\n");
22	}
23	return 0;
24	}

Tabel 72. Source Code Jawaban Soal 1 Bahasa Python

1	def print_matrix(baris, kolom, data):
2	index = 0
3	for i in range(baris):
4	for j in range(kolom):
5	if j != kolom - 1:
6	print(data[index], end=" ")
7	else:
8	print(data[index])
9	index += 1
10	
11	print("Masukkan jumlah baris dan kolom:",
12	end=" ")
13	baris, kolom = map(int, input().split())
14	print("Masukkan angka-angka dalam
15	matriks:", end=" ")
	data = list(map(int, input().split()))
	print_matrix(baris, kolom, data)

B. Output Program

```

PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK601-2410817320001-NazlaSalsabila.c -o PRAK601-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK601-2410817320001-NazlaSalsabila }
Masukkan jumlah baris dan kolom: 2 3
Masukkan angka-angka dalam matriks: 1 2 3 4 5 6
1 2 3
4 5 6
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK601-2410817320001-NazlaSalsabila.c -o PRAK601-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK601-2410817320001-NazlaSalsabila }
Masukkan jumlah baris dan kolom: 3 3
Masukkan angka-angka dalam matriks: 34 56 41 45 36 37 51 32 46
34 56 41
45 36 37
51 32 46
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK601-2410817320001-NazlaSalsabila.c -o PRAK601-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK601-2410817320001-NazlaSalsabila }
Masukkan jumlah baris dan kolom: 4 5
Masukkan angka-angka dalam matriks: 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 D:\coding>

```

Gambar 71, Screenshot Output Soal 1 Bahasa C

```

PS D:\coding> python -u "d:\coding\PRAK601-2410817320001-NazlaSalsabila.py"
Masukkan jumlah baris dan kolom: 2 3
Masukkan angka-angka dalam matriks: 1 2 3 4 5 6
1 2 3
4 5 6
PS D:\coding> python -u "d:\coding\PRAK601-2410817320001-NazlaSalsabila.py"
Masukkan jumlah baris dan kolom: 3 3
Masukkan angka-angka dalam matriks: 34 56 41 45 36 37 51 32 46
34 56 41
45 36 37
51 32 46
PS D:\coding> python -u "d:\coding\PRAK601-2410817320001-NazlaSalsabila.py"
Masukkan jumlah baris dan kolom: 4 5
Masukkan angka-angka dalam matriks: 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 D:\coding>

```

Gambar 72. Screenshot Output Soal 1 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [3, 10, 15, dan 16], terdapat `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `int baris, kolom;` artinya mendeklarasikan dua variabel integer baris dan kolom untuk menyimpan jumlah baris dan kolom dari matriks.

Pada baris [5], terdapat `printf("Masukkan jumlah baris dan kolom: ");` artinya program meminta untuk memasukkan jumlah baris dan kolom.

Pada baris [6], terdapat `scanf("%d %d", &baris, &kolom);` artinya program membaca dua bilangan integer yang dimasukkan dan menyimpan ke variabel baris dan kolom.

Pada baris [8], terdapat `int data[baris * kolom];` artinya mendeklarasikan array data dengan ukuran dinamis berdasarkan hasil perkalian baris * kolom.

Pada baris [9], terdapat `printf("Masukkan angka-angka dalam matriks: ");` artinya program meminta memasukkan angka-angka matriks secara berurutan.

Pada baris [10 dan 11], terdapat `for (int i = 0; i < baris * kolom; i++) scanf("%d", &data[i]);` artinya for digunakan untuk membaca semua elemen matriks sebanyak baris * kolom dan menyimpannya di array data.

Pada baris [14], terdapat `int index = 0;` artinya variabel ini untuk melacak posisi elemen dalam array data saat mencetak matriks.

Pada baris [15 dan 16], terdapat `for (int i = 0; i < baris; i++) { for (int j = 0; j < kolom; j++) {` artinya dua loop bersarang (nested loop) digunakan untuk mencetak elemen-elemen matriks dalam bentuk baris x kolom. Loop luar (`for (int i = 0; i < baris; i++)`) untuk setiap baris. Loop dalam (`for (int j = 0; j < kolom; j++)`) untuk setiap elemen di baris tersebut.

Pada baris [17], terdapat `printf("%d", data[index++]);` artinya untuk mencetak elemen dari array data sesuai dengan indeks index, lalu index dinaikkan (increment).

Pada baris [18], terdapat `if (j < kolom - 1) printf(" ");` artinya memastikan ada spasi antar elemen kecuali di elemen terakhir baris tersebut.

Pada baris [13], terdapat `printf("\n");` artinya untuk mencetak garis baru setelah semua elemen di satu baris selesai dicetak.

Pada baris [23], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [12, 19, 21, dan 24], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `def print_matrix(baris, kolom, data):` artinya mendeklarasikan fungsi `print_matrix` dengan tiga parameter, yaitu `baris`: jumlah baris matriks. `kolom`: jumlah kolom matriks. `data`: list satu dimensi yang berisi elemen-elemen matriks.

Pada baris [2], terdapat `index = 0` artinya untuk melacak posisi elemen dalam array `data` saat mencetak matriks.

Pada baris [3 dan 4], terdapat `for i in range(baris):` `for j in range(kolom)` artinya `for i in range(baris)`: loop luar untuk iterasi sebanyak jumlah baris. `for j in range(kolom)`: loop dalam untuk iterasi sebanyak jumlah kolom.

Pada baris [5 – 8], terdapat `if j != kolom - 1: print(data[index], end=" ")` `else: print(data[index])` artinya mengecek apakah elemen adalah elemen terakhir di baris: jika bukan elemen terakhir, gunakan `end=" "` untuk mencetak elemen tanpa pindah baris, diikuti spasi, sebaliknya jika elemen terakhir, gunakan `print()` default untuk mencetak elemen lalu pindah ke baris baru.

Pada baris [9], terdapat `index += 1` artinya `index` dinaikkan untuk mengakses elemen berikutnya di list `data`.

Pada baris [11], terdapat `print("Masukkan jumlah baris dan kolom:", end=" ")` artinya menampilkan teks untuk memasukkan input. `end=" "` memastikan input ada di baris yang sama dengan teks.

Pada baris [12], terdapat `baris, kolom = map(int, input().split())` artinya `baris, kolom = ...`: menyimpan dua nilai integer ke dalam variabel `baris` dan `kolom`. `map(int, input().split())`: mengonversi daftar string hasil `split()` menjadi integer menggunakan fungsi `map`.

Pada baris [13], terdapat `print("Masukkan angka-angka dalam matriks:", end=" ")` artinya untuk meminta memasukkan elemen-elemen matriks.

Pada baris [14], terdapat `data = list(map(int, input().split()))` artinya **`list(...)`**: mengonversi hasil `map` menjadi list Python untuk menyimpan elemen-elemen matriks. `map(int, input().split())`: mengonversi daftar string angka menjadi integer.

`input().split()`: membaca input angka dari pengguna sebagai string, lalu memisahkannya menjadi daftar string berdasarkan spasi.

Pada baris [15], terdapat `print_matrix(baris, kolom, data)` artinya memanggil fungsi `print_matrix` sebelumnya.

SOAL 2

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

Misalnya terdapat 3 ruangan dengan rincian sebagai berikut:

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

Format input:

Baris pertama merupakan jumlah ruangan.

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

Format output:

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

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

Simpan coding anda dengan nama: **PRAK602-NIM-Nama.py** dan **PRAK5602-NIM-Nama**.

A. Source Code

Tabel 73. Source Code Jawaban Soal 2 Bahasa C

1	#include <stdio.h>
---	--------------------

2	
3	int main() {
4	int n;
5	printf("Masukkan jumlah ruangan: ");
6	scanf("%d", &n);
7	
8	int zetsu[n];
9	printf("Masukkan banyaknya zetsu putih pada setiap ruangan: ");
10	for (int i = 0; i < n; i++) {
11	scanf("%d", &zetsu[i]);
12	}
13	
14	for (int i = 0; i < n; i++) {
15	printf("%d ", zetsu[i] * (i +
16	1));
17	}
18	printf("\n");
19	return 0;
20	}

Tabel 74. Source Code Jawaban Soal 2 Bahasa Python

1	n = int(input("Masukkan jumlah ruangan: "))
2	zetsu = list(map(int, input("Masukkan banyaknya zetsu putih pada setiap ruangan: ").split()))
3	
4	for i in range(n):
5	print(zetsu[i] * (i + 1), end=" ")
6	print()

B. Output Program

```

PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK602-2410817320001-NazlaSalsabila.c -o PRAK602-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK602-2410817320001-NazlaSalsabila }
Masukkan jumlah ruangan: 3
Masukkan banyaknya zetsu putih pada setiap ruangan: 4 7 9
4 14 27
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK602-2410817320001-NazlaSalsabila.c -o PRAK602-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK602-2410817320001-NazlaSalsabila }
Masukkan jumlah ruangan: 5
Masukkan banyaknya zetsu putih pada setiap ruangan: 1 2 3 4 5
1 4 9 16 25
PS D:\coding> cd "d:\coding\" ; if ($?) { gcc PRAK602-2410817320001-NazlaSalsabila.c -o PRAK602-2410817320001-NazlaSalsabila } ; if ($?) { .\PRAK602-2410817320001-NazlaSalsabila }
Masukkan jumlah ruangan: 10
Masukkan banyaknya zetsu putih pada setiap ruangan: 5 6 45 78 21 3 6 8 45 1
5 12 135 312 105 18 42 64 405 10
PS D:\coding>

```

Gambar 73. Screenshot Output Soal 2 Bahasa C

```

PS D:\coding> python -u "d:\coding\PRAK602-2410817320001-NazlaSalsabila.py"
Masukkan jumlah ruangan: 3
Masukkan banyaknya zetsu putih pada setiap ruangan: 4 7 9
4 14 27
PS D:\coding> python -u "d:\coding\PRAK602-2410817320001-NazlaSalsabila.py"
Masukkan jumlah ruangan: 5
Masukkan banyaknya zetsu putih pada setiap ruangan: 1 2 3 4 5
1 4 9 16 25
PS D:\coding> python -u "d:\coding\PRAK602-2410817320001-NazlaSalsabila.py"
Masukkan jumlah ruangan: 10
Masukkan banyaknya zetsu putih pada setiap ruangan: 5 6 45 78 21 3 6 8 45 1
5 12 135 312 105 18 42 64 405 10
PS D:\coding>

```

Gambar 74. Screenshot Output Soal 2 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [3, 10, dan 14], terdapat tanda `{` kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `int n`; artinya variabel `n` digunakan untuk menyimpan jumlah ruangan.

Pada baris [5], terdapat `printf("Masukkan jumlah ruangan: ");` artinya untuk memasukkan jumlah ruangan.

Pada baris [6], terdapat `scanf("%d", &n);` artinya membaca nilai integer yang dimasukkan dan menyimpannya di variabel `n`.

Pada baris [8], terdapat `int zetsu[n];` artinya menyimpan jumlah Zetsu Putih di setiap ruangan.

Pada baris [9], terdapat `printf("Masukkan banyaknya zetsu putih pada setiap ruangan: ");` artinya untuk memasukkan banyaknya Zetsu Putih untuk setiap ruangan.

Pada baris [10], terdapat `for (int i = 0; i < n; i++)` artinya untuk mengulangi proses input sebanyak `n` kali.

Pada baris [11], terdapat `scanf("%d", &zetsu[i]);` artinya membaca nilai integer yang dimasukkan dan menyimpannya ke elemen ke-`i` dari array `zetsu`.

Pada baris [14], terdapat `for (int i = 0; i < n; i++)` artinya untuk memproses setiap elemen dalam array `zetsu`.

Pada baris [15], terdapat `printf("%d ", zetsu[i] * (i + 1));` artinya elemen `zetsu[i]` dikalikan dengan `i + 1`, yaitu indeks elemen ditambah 1 (karena indeks array dimulai dari 0).

Pada baris [17], terdapat `printf("\n");` artinya menambahkan baris baru setelah mencetak semua hasil perhitungan.

Pada baris [19], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [12, 16, dan 20], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `n = int(input("Masukkan jumlah ruangan: "))` artinya variabel `n` yang menyimpan jumlah ruangan. `int(...)`: mengonversi input string menjadi integer. `input("Masukkan jumlah ruangan: ")` untuk meminta memasukkan jumlah ruangan.

Pada baris [2], terdapat `zetsu = list(map(int, input("Masukkan banyaknya zetsu putih pada setiap ruangan: ").split()))` `zetsu =` artinya variabel `Zetsu` yang menyimpan

daftar jumlah Zetsu Putih untuk setiap ruangan. `list(...)`: Mengonversi hasil map menjadi list Python. `map(int, ...)`: mengonversi elemen-elemen daftar string menjadi integer. `input("Masukkan banyaknya zetsu putih pada setiap ruangan: ")`: untuk meminta memasukkan jumlah Zetsu Putih di setiap ruangan dalam satu baris. `split()`: memisahkan input string menjadi daftar berdasarkan spasi.

Pada baris [4], terdapat `for i in range(n)`: artinya untuk mengiterasi sebanyak `n` kali, di mana `i` adalah indeks (dimulai dari 0 hingga `n-1`).

Pada baris [5], terdapat `print(zetsu[i] * (i + 1), end=" ")` artinya print mencetak hasil perhitungan pada satu baris. `zetsu[i] * (i + 1)`: mengambil elemen ke-`i` dari list `zetsu`, kemudian mengalikan elemen tersebut dengan `(i + 1)` untuk memberikan "bobot" sesuai urutan ruangan (dimulai dari 1). `end=" "` mencegah pindah ke baris baru setelah mencetak.

Pada baris [6], terdapat `print()` artinya mencetak baris baru setelah loop.

SOAL 3

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

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

2	3
4	5
Jadi hasilnya itu 8 15	

Format input :

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

Format Output :

hasil output adalah baris pertama di kali baris ke dua,

jadi liat contoh input output untuk lebih jelas

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

Simpan coding anda dengan nama: **PRAK603-NIM-Nama.py** dan **PRAK603-NIM-Nama**.

A. Source Code

Tabel 75. Source Code Jawaban Soal 3 Bahasa C

1	#include <stdio.h>
2	
3	int main() {
4	int n1, n2;
5	printf("Masukkan n1 dan n2: ");
6	scanf("%d %d", &n1, &n2);
7	
8	if (n1 != n2) {
9	printf("Jumlah tidak sama\n");
10	} else {
11	int baris1[n1], baris2[n2];
12	printf("Masukkan %d angka untuk baris pertama: ", n1);
13	for (int i = 0; i < n1; i++) {
14	scanf("%d", &baris1[i]);
15	}
16	
17	printf("Masukkan %d angka untuk baris kedua: ", n2);
18	for (int i = 0; i < n2; i++) {
19	scanf("%d", &baris2[i]);
20	}
21	
22	printf("Hasil perkalian:\n");
23	for (int i = 0; i < n1; i++) {
24	printf("%d ", baris1[i] * baris2[i]);
25	}
26	}
27	

28	return 0;
29	}

Tabel 76. Source Code Jawaban Soal 3 Bahasa Python

1	n1, n2 = map(int, input("Masukkan n1 dan n2:").split())
2	if n1 != n2:
3	print("Jumlah tidak sama")
4	else:
5	print(f"Masukkan {n1} angka untuk baris pertama:", end=" ")
6	baris1 = list(map(int, input().split()))
7	print(f"Masukkan {n2} angka untuk baris kedua:", end=" ")
8	baris2 = list(map(int, input().split()))
9	hasil = [baris1[i] * baris2[i] for i in range(n1)]
10	print("Hasil perkalian:")
11	print(" ".join(map(str, hasil)))

B. Output Program

```

PS D:\coding> cd "d:\coding"; if ($?) { gcc PRAK003-2410017320001-NazlaSalsabila.c -o PRAK003-2410017320001-NazlaSalsabila }; if ($?) { .\PRAK003-2410017320001-NazlaSalsabila }
Masukkan n1 dan n2: 3 3
Masukkan 3 angka untuk baris pertama: 1 2 3
Masukkan 3 angka untuk baris kedua: 4 5 6
Hasil perkalian:
4 10 18
PS D:\coding> cd "d:\coding"; if ($?) { gcc PRAK003-2410017320001-NazlaSalsabila.c -o PRAK003-2410017320001-NazlaSalsabila }; if ($?) { .\PRAK003-2410017320001-NazlaSalsabila }
Masukkan n1 dan n2: 2 3
Jumlah tidak sama
PS D:\coding> cd "d:\coding"; if ($?) { gcc PRAK003-2410017320001-NazlaSalsabila.c -o PRAK003-2410017320001-NazlaSalsabila }; if ($?) { .\PRAK003-2410017320001-NazlaSalsabila }
Masukkan n1 dan n2: 5 5
Masukkan 5 angka untuk baris pertama: 1 2 3 4 5
Masukkan 5 angka untuk baris kedua: 5 4 3 2 1
Hasil perkalian:
5 8 9 8 5
PS D:\coding>

```

Gambar 75. Screenshot Output Soal 3 Bahasa C

```

PS D:\coding> python -u "d:\coding\PRAK003-2410017320001-NazlaSalsabila.py"
Masukkan n1 dan n2: 3 3
Masukkan 3 angka untuk baris pertama: 1 2 3
Masukkan 3 angka untuk baris kedua: 4 5 6
Hasil perkalian:
4 10 18
PS D:\coding> python -u "d:\coding\PRAK003-2410017320001-NazlaSalsabila.py"
Masukkan n1 dan n2: 2 3
Jumlah tidak sama
PS D:\coding> python -u "d:\coding\PRAK003-2410017320001-NazlaSalsabila.py"
Masukkan n1 dan n2: 5 5
Masukkan 5 angka untuk baris pertama: 1 2 3 4 5
Masukkan 5 angka untuk baris kedua: 5 4 3 2 1
Hasil perkalian:
5 8 9 8 5
PS D:\coding>

```

Gambar 76. Screenshot Output Soal 3 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [3, 8, 10, 13, 18, dan 23], terdapat tanda { kurung kurawal menandai awal blok kode.

Pada baris [4], terdapat `int n1, n2;` artinya deklarasi dua variabel integer `n1` dan `n2` untuk menyimpan jumlah elemen dalam dua array.

Pada baris [5], terdapat `printf("Masukkan n1 dan n2: ");` artinya mencetak teks ke layar untuk memasukkan dua angka (`n1` dan `n2`).

Pada baris [6], terdapat `scanf("%d %d", &n1, &n2);` artinya membaca dua bilangan bulat dari input pengguna dan menyimpannya ke `n1` dan `n2`.

Pada baris [8 – 10], terdapat `if (n1 != n2) printf("Jumlah tidak sama\n"); else` artinya `if` memeriksa apakah `n1` tidak sama dengan `n2`. Jika ukurannya berbeda, program mencetak Jumlah tidak sama dan tidak melanjutkan ke langkah berikutnya. `else` akan dijalankan jika `n1 = n2`.

Pada baris [11], terdapat `int baris1[n1], baris2[n2];` artinya dua array dinamis (`baris1` dan `baris2`) dideklarasikan dengan ukuran masing-masing `n1` dan `n2`.

Pada baris [12 – 19], terdapat `printf("Masukkan %d angka untuk baris pertama: ", n1); for (int i = 0; i < n1; i++) scanf("%d", &baris1[i]); printf("Masukkan %d angka untuk baris kedua: ", n2); for (int i = 0; i < n2; i++) scanf("%d", &baris2[i]);` artinya `printf` meminta memasukkan elemen untuk array pertama dan kedua. `for`

digunakan untuk mengulangi proses input elemen sebanyak $n1$ dan $n2$ kali. Setiap elemen disimpan di `baris1` dan `baris2[i]`.

Pada baris [22 – 24], terdapat `printf("Hasil perkalian:\n")` `for (int i = 0; i < n1; i++) printf("%d ", baris1[i] * baris2[i]);` artinya `printf` mencetak pesan hasil perkalian. `for` mengiterasi setiap indeks dari 0 hingga $n1 - 1$. Pada setiap iterasi, elemen pada indeks i dari `baris1` dan `baris2` dikalikan, lalu hasilnya dicetak.

Pada baris [28], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi `main` dan menandakan akhir dari eksekusi program.

Pada baris [10, 15, 20, 25, 26, dan 29], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `n1, n2 = map(int, input("Masukkan n1 dan n2: ").split())` artinya `map(int, ...)` mengonversi string menjadi bilangan bulat (`int`). `input()` meminta memasukkan dua angka dengan spasi sebagai pemisah. `split()` memisahkan input menjadi dua string. Hasilnya disimpan ke dalam variabel `n1` dan `n2`.

Pada baris [2 – 4], terdapat `if n1 != n2: print("Jumlah tidak sama") else:` artinya `if` memeriksa apakah `n1` tidak sama dengan `n2`. Jika ukurannya berbeda, maka akan mencetak Jumlah tidak sama, dan program berhenti. Jika `n1 == n2`, maka blok `else` dijalankan.

Pada baris [5 – 8], terdapat `print(f"Masukkan {n1} angka untuk baris pertama:", end=" ")` `baris1 = list(map(int, input().split()))` `print(f"Masukkan {n2} angka untuk baris kedua:", end=" ")` `baris2 = list(map(int, input().split()))` artinya `print(f"...", end=" ")` mencetak teks tanpa pindah baris. `list()` mengubah hasil `map` menjadi daftar (`baris1` dan `baris2`). `input()` meminta memasukkan angka-angka untuk baris pertama dan kedua, dipisahkan oleh spasi. `split()` memisahkan input menjadi daftar string, kemudian `map(int, ...)` mengonversinya menjadi daftar bilangan bulat.

Pada baris [9], terdapat `hasil = [baris1[i] * baris2[i] for i in range(n1)]` artinya untuk menghitung hasil perkalian elemen pada indeks yang sama dari `baris1` dan `baris2`.

`range(n1)` menghasilkan angka dari 0 hingga `n1-1` yang digunakan untuk mengakses elemen pada kedua daftar. `baris1[i] * baris2[i]` menghitung perkalian elemen pada indeks `i`.

Pada baris [10], terdapat `print("Hasil perkalian:")` artinya mencetak hasil perkalian.

Pada baris [11], artinya `print(" ".join(map(str, hasil)))` artinya mengubah setiap elemen dalam daftar hasil menjadi string, kemudian menggabungkan spasi sebagai pemisah.

SOAL 4

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

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

Format input:

Baris pertama merupakan kode yang dimiliki oleh Shikamaru

Baris kedua merupakan pesan yang diterima oleh Shikamaru

Format output:

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

Baris kedua merupakan jumlah tanda bintang

Baris ketiga merupakan jumlah tanda pagar

Baris keempat merupakan pemberitahuan pesan asli atau pesan palsu

Input	Output
Bahasa Pemrograman	Panjang kalimat berbeda, pesan palsu
Ini Pesan Rahasia Ini p354n Rahas14	*** #####* *****## * = 8

	# = 7 Pesan Asli
Aku Pasti Bisa Berjuang lebih	#####*##### *= 1 #= 13 Pesan Palsu

Simpan coding anda dengan nama: **PRAK604-NIM-Nama.py** dan **PRAK604-NIM-Nama**.

A. Source Code

Tabel 77. Source Code Jawaban Soal 4 Bahasa C

1	#include <stdio.h>
2	#include <string.h>
3	
4	int main() {
	char kode[100], pesan[100],
5	hasil[100];
	int jumlah_bintang = 0, jumlah_pagar
6	= 0;
7	
8	printf("Masukkan kode Shikamaru: ");
9	scanf(" %[^\\n]", kode);
	printf("Masukkan pesan yang diterima:
10	");
11	scanf(" %[^\\n]", pesan);
12	
13	if (strlen(kode) != strlen(pesan)) {
	printf("Panjang kalimat berbeda,
14	pesan palsu\\n");
15	} else {

16	int panjang = strlen(kode);
	for (int i = 0; i < panjang; i++)
17	{
	hasil[i] = (kode[i] == ' ' &&
18	pesan[i] == ' ') ? ' ' :
	(kode[i] ==
	pesan[i] ? (jumlah_bintang++, '*') :
19	(jumlah_pagar++, '#'));
20	}
21	hasil[strlen(kode)] = '\0';
	printf("%s\n* = %d\n# = %d\n",
22	hasil, jumlah_bintang, jumlah_pagar);
	printf("%s\n", jumlah_bintang >=
	jumlah_pagar ? "Pesan Asli" : "Pesan
23	Palsu");
24	}
25	
26	return 0;
27	}

Tabel 78. Source Code Jawaban Soal 4 Bahasa Python

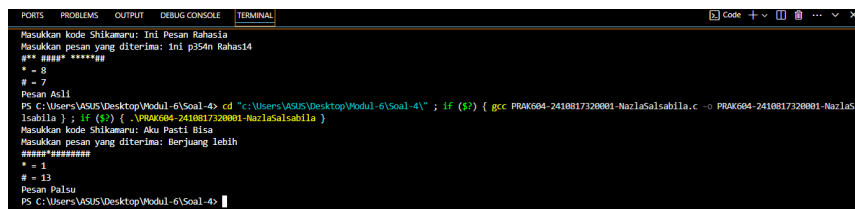
1	kode = input("Masukkan kode Shikamaru: ")
	pesan = input("Masukkan pesan yang
2	diterima: ")
3	
4	if len(kode) != len(pesan):
	print("Panjang kalimat berbeda, pesan
5	palsu")
6	else:
7	hasil = []
8	jumlah_bintang = 0
9	jumlah_pagar = 0
10	
11	for k, p in zip(kode, pesan):
12	if k == " " and p == " ":
13	hasil.append(" ")
14	elif k == p:
15	hasil.append("*")


```

16         jumlah_bintang += 1
17     else:
18         hasil.append("#")
19         jumlah_pagar += 1
20
21     print("".join(hasil))
22     print(f"* = {jumlah_bintang}")
23     print(f"# = {jumlah_pagar}")
24
25     if jumlah_bintang >= jumlah_pagar:
26         print("Pesan Asli")
27     else:
28         print("Pesan Palsu")

```

B. Output Program

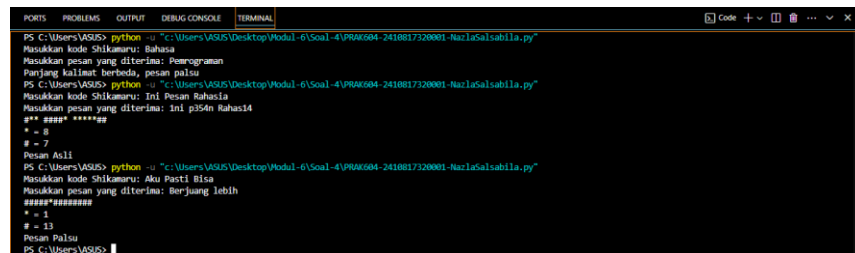


```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Masukkan kode Shikamaru: Ini Pesan Rahasia
Masukkan pesan yang diterima: ini p354n Rahasi4
** #####
# = 8
# = 7
Pesan Asli
PS C:\Users\VASUS\Desktop\Modul-6\Soal-4> cd "C:\Users\VASUS\Desktop\Modul-6\Soal-4"; if ($?) { gcc PRAG604-2410817320001-NazlaSalsabila.c -o PRAG604-2410817320001-NazlaSalsabila }; if ($?) { .\PRAG604-2410817320001-NazlaSalsabila }
Masukkan kode Shikamaru: Aku Pasti Bisa
Masukkan pesan yang diterima: Berjuang lebih
#####
# = 1
# = 13
Pesan Palsu
PS C:\Users\VASUS\Desktop\Modul-6\Soal-4>

```

Gambar 77. Screenshot Output Soal 4 Bahasa C



```

PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\VASUS> python -u "C:\Users\VASUS\Desktop\Modul-6\Soal-4\PRAG604-2410817320001-NazlaSalsabila.py"
Masukkan kode Shikamaru: Bahasa
Masukkan pesan yang diterima: Perang
Panjang kalimat berbeda, pesan palsu
PS C:\Users\VASUS> python -u "C:\Users\VASUS\Desktop\Modul-6\Soal-4\PRAG604-2410817320001-NazlaSalsabila.py"
Masukkan kode Shikamaru: Ini Pesan Rahasia
Masukkan pesan yang diterima: ini p354n Rahasi4
** #####
# = 8
# = 7
Pesan Asli
PS C:\Users\VASUS> python -u "C:\Users\VASUS\Desktop\Modul-6\Soal-4\PRAG604-2410817320001-NazlaSalsabila.py"
Masukkan kode Shikamaru: Aku Pasti Bisa
Masukkan pesan yang diterima: Berjuang lebih
#####
# = 1
# = 13
Pesan Palsu
PS C:\Users\VASUS>

```

Gambar 78. Screenshot Output Soal 4 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya `printf()`. Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [2], terdapat `#include <string.h>` artinya mengecek apakah panjang string pada variabel kode sama dengan panjang string pada variabel pesan. Jika panjangnya berbeda, maka pesan tersebut palsu.

Pada baris [4], terdapat `int main() {`. `int` menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. `Main` merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [4, 13, 15, dan 17], terdapat tanda `{` kurung kurawal menandai awal blok kode.

Pada baris [5 – 6], terdapat `char kode[100], pesan[100], hasil[100]; int jumlah_bintang = 0, jumlah_pagar = 0;` artinya kode: menyimpan input string yang disebut "kode Shikamaru". pesan: menyimpan input string yang disebut "pesan yang diterima". hasil: menyimpan hasil perbandingan tiap karakter antara kode dan pesan. jumlah_bintang: menghitung jumlah karakter yang cocok (ditandai dengan *). jumlah_pagar: menghitung jumlah karakter yang tidak cocok (ditandai dengan #)

Pada baris [8 – 11], terdapat `printf("Masukkan kode Shikamaru: "); scanf("%[^\n]", kode); printf("Masukkan pesan yang diterima: "); scanf("%[^\n]", pesan);` artinya meminta memasukkan string untuk variabel kode dan pesan. Format `%[^\n]` digunakan agar dapat membaca string termasuk spasi hingga karakter newline (`\n`).

Pada baris [13 – 14], terdapat `if (strlen(kode) != strlen(pesan)) printf("Panjang kalimat berbeda, pesan palsu\n"); else` artinya `strlen` untuk menghitung panjang string. Jika panjang string kode dan pesan berbeda, langsung cetak "Panjang kalimat berbeda, pesan palsu" dan selesai.

Pada baris [16 – 21], terdapat `int panjang = strlen(kode); for (int i = 0; i < panjang; i++) hasil[i] = (kode[i] == ' ' && pesan[i] == ' ') ? ' ' : (kode[i] == pesan[i] ? (jumlah_bintang++, '*') : (jumlah_pagar++, '#')); hasil[strlen(kode)] = '\0';` artinya `int panjang = strlen(kode);` Menyimpan panjang string untuk diiterasi. Jika kedua karakter adalah spasi (`kode[i] == ' ' && pesan[i] == ' '`), tambahkan ' ' ke hasil. Jika karakter sama (`kode[i] == pesan[i]`), tambahkan '*' ke hasil dan tingkatkan jumlah_bintang. Jika karakter berbeda, tambahkan '#' ke hasil dan tingkatkan jumlah_pagar.

hasil[strlen(kode)] = '\0'; artinya menambahkan karakter null-terminator untuk menandai akhir string hasil.

Pada baris [22 – 24], terdapat `printf("%s\n* = %d\n# = %d\n", hasil, jumlah_bintang, jumlah_pagar); printf("%s\n", jumlah_bintang >= jumlah_pagar ? "Pesan Asli" : "Pesan Palsu");` artinya menampilkan hasil perbandingan (hasil) yang berisi kombinasi simbol *, #, atau spasi. Menampilkan jumlah simbol * dan #. Mengevaluasi apakah pesan asli atau palsu. Jika jumlah * lebih besar atau sama dengan #, cetak "Pesan Asli". Jika tidak, cetak "Pesan Palsu".

Pada baris [26], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [15, 20, 24, dan 27], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `kode = input("Masukkan kode Shikamaru: ")` artinya variabel untuk menyimpan input string yang disebut sebagai "kode Shikamaru".

Pada baris [2], terdapat `pesan = input("Masukkan pesan yang diterima: ")` artinya variabel untuk menyimpan input string yang disebut sebagai "pesan yang diterima"

Pada baris [4 – 6], terdapat `if len(kode) != len(pesan): print("Panjang kalimat berbeda, pesan palsu") else:` artinya mengecek apakah panjang string kode dan pesan berbeda. Jika berbeda, program langsung mencetak pesan "Panjang kalimat berbeda, pesan palsu" dan selesai di sini.

Pada baris [7 – 9], terdapat `hasil = [] jumlah_bintang = 0 jumlah_pagar = 0` artinya hasil: untuk menyimpan simbol hasil perbandingan karakter kode dan pesan satu per satu. `jumlah_bintang`: untuk menghitung jumlah simbol * yang menunjukkan karakter cocok. `jumlah_pagar`: untuk menghitung jumlah simbol # yang menunjukkan karakter tidak cocok.

Pada baris [11 – 19], terdapat `for k, p in zip(kode, pesan): if k == " " and p == " ": hasil.append(" ") elif k == p: hasil.append("*") jumlah_bintang +=`

1 else: hasil.append("#") jumlah_pagar += 1 artinya for k, p in zip(kode, pesan): Mengambil masing-masing karakter dari kode (k) dan pesan (p) secara bersamaan. Jika kedua karakter adalah spasi (k == " " dan p == " "), tambahkan spasi (" ") ke list hasil. Jika karakter sama (k == p), tambahkan * ke hasil dan tingkatkan jumlah_bintang sebanyak 1. Jika karakter berbeda, tambahkan # ke hasil dan tingkatkan jumlah_pagar sebanyak 1.

Pada baris [21 – 23], terdapat print("".join(hasil)) print(f"* = {jumlah_bintang}") print(f"# = {jumlah_pagar}") artinya"".join(hasil): menggabungkan isi list hasil menjadi sebuah string untuk ditampilkan. jumlah_bintang dan jumlah_pagar: dicetak untuk menunjukkan jumlah masing-masing simbol.

Pada baris [25 – 28], terdapat if jumlah_bintang >= jumlah_pagar: print("Pesan Asli") else: print("Pesan Palsu") artinya jika jumlah simbol * lebih besar atau sama dengan jumlah simbol #, maka pesan dinyatakan "Pesan Asli". Jika tidak, pesan dinyatakan "Pesan Palsu".

SOAL 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

Simpan coding anda dengan nama: **PRAK605-NIM-Nama.py** dan **PRAK605-NIM-Nama**.

A. Source Code

Tabel 79. Source Code Jawaban Soal 5 Bahasa C

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

29	}
30	}
31	}
32	
33	printf("Matriks AXB\n");
34	for (int i = 0; i < n; i++) {
35	for (int j = 0; j < n; j++) {
36	printf("%d ", C[i][j]);
37	}
38	printf("\n");
39	}
40	
41	return 0;
42	}

Tabel 80. Source Code Jawaban Soal 5 Bahasa Python

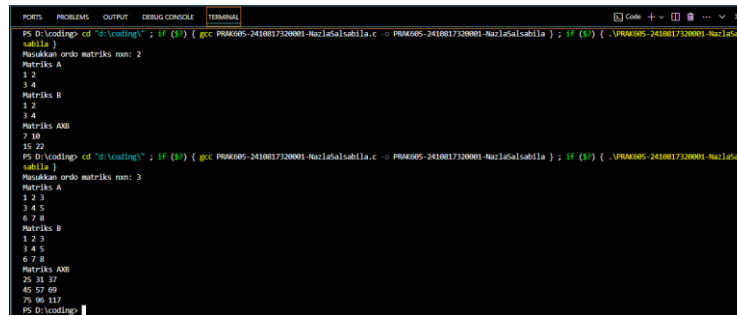
	n = int(input("Masukkan ordo matriks nxn:"))
1)
2	A = []
3	print("Matriks A")
4	for i in range(n):
	A.append(list(map(int,
5	input().split()))
6	
7	B = []
8	print("Matriks B")
9	for i in range(n):
	B.append(list(map(int,
10	input().split()))
11	
12	C = [[0] * n for _ in range(n)]
13	

```

14     for i in range(n):
15         for j in range(n):
16             for k in range(n):
17                 C[i][j] += A[i][k] * B[k][j]
18
19     print("Matriks AXB")
20     for row in C:
21         print(" ".join(map(str, row)))

```

B. Output Program

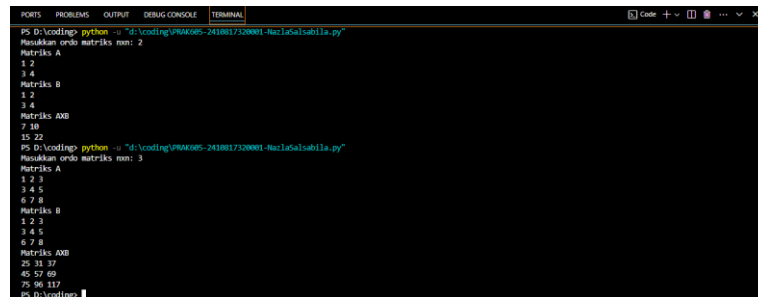


```

PS D:\coding> cd "d:\coding"; if ($?) { gcc PRAK005-241001720001-NuzulGalsabilla.c -o PRAK005-241001720001-NuzulGalsabilla }; if ($?) { .\PRAK005-241001720001-NuzulGalsabilla }
Masukkan ordo matriks mm: 2
Matriks A
1 2
3 4
Matriks B
1 2
3 4
Matriks AxB
7 10
15 22
PS D:\coding> cd "d:\coding"; if ($?) { gcc PRAK005-241001720001-NuzulGalsabilla.c -o PRAK005-241001720001-NuzulGalsabilla }; if ($?) { .\PRAK005-241001720001-NuzulGalsabilla }
Masukkan ordo matriks mm: 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 D:\coding>

```

Gambar 79. Screenshot Output Soal 5 Bahasa C



```

PS D:\coding> python -i "d:\coding\PRAK005-241001720001-NuzulGalsabilla.py"
Masukkan ordo matriks mm: 2
Matriks A
1 2
3 4
Matriks B
1 2
3 4
Matriks AxB
7 10
15 22
PS D:\coding> python -i "d:\coding\PRAK005-241001720001-NuzulGalsabilla.py"
Masukkan ordo matriks mm: 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 D:\coding>

```

Gambar 80. Screenshot Output Soal 5 Bahasa Python

C. Pembahasan

Bahasa C:

Pada baris [1], terdapat `#include <stdio.h>`. `#include` adalah preprocessor untuk mengakses file `stdio.h` agar codingan dapat mengakses data yang ada. Sedangkan `stdio.h` adalah berisi deklarasi untuk fungsi-fungsi standar input-output salah satunya

printf(). Dengan menyertakan file ini, program dapat menggunakan fungsi untuk menampilkan output ke layar.

Pada baris [3], terdapat `int main() {`. Int menyatakan fungsi mengembalikan sejumlah bilangan bulat, biasanya 0 untuk menunjukkan bahwa program berhasil dijalankan. Main merupakan fungsi utama saat program dijalankan. `()` kurung bulat berfungsi untuk mendefinisikan parameter yang diterima oleh fungsi.

Pada baris [3, 11, 12, 18, 19, 24, 25, 27, 34, dan 35], terdapat `{` kurung kurawal menandai awal blok kode.

Pada baris [4 – 6], terdapat `printf("Masukkan ordo matriks nxn: "); int n; scanf("%d", &n);` artinya printf meminta untuk memasukkan ordo matriks ($n \times n$). scanf digunakan untuk membaca nilai n yang menunjukkan ukuran matriks $n \times n$.

Pada baris [8], terdapat `int A[n][n], B[n][n], C[n][n];` artinya tiga matriks 2D dideklarasikan, yaitu A, B, dan C dengan berukuran $n \times n$. Matriks A dan B adalah matriks input, sedangkan C akan menyimpan hasil perkalian matriks A dan B.

Pada baris [10 – 13], terdapat `printf("Matriks A\n") for (int i = 0; i < n; i++) for (int j = 0; j < n; j++) scanf("%d", &A[i][j]);` artinya program meminta untuk memasukkan elemen-elemen matriks A baris per baris. Dua loop for digunakan untuk mengisi semua elemen matriks A dengan scanf.

Pada baris [17 – 20], terdapat `printf("Matriks B\n"); for (int i = 0; i < n; i++) for (int j = 0; j < n; j++) scanf("%d", &B[i][j]);` artinya program meminta untuk memasukkan elemen-elemen matriks B baris per baris. Dua loop for digunakan untuk mengisi semua elemen matriks B dengan scanf.

Pada baris [24 – 28], terdapat `for (int i = 0; i < n; i++) for (int tj = 0; j < n; j++) C[i][j] = 0; for (int k = 0; k < n; k++) C[i][j] += A[i][k] * B[k][j];` artinya matriks hasil perkalian disimpan dalam matriks C. Tiga loop for digunakan untuk melakukan perkalian matriks A dan B: i dan j adalah indeks baris dan kolom pada matriks C. k adalah indeks untuk iterasi perkalian elemen-elemen matriks A dan B. Setiap elemen `C[i][j]` dihitung dengan menjumlahkan hasil perkalian elemen-elemen dari baris ke-i pada matriks A dan kolom ke-j pada matriks B.

Pada baris [33 – 38], terdapat `printf("Matriks AXB\n");` `for (int i = 0; i < n ;i++)` `for (int j = 0 ;j<n;j++)` `printf("%d", C[i]);` `printf("\n");` artinya program mencetak hasil matriks C (hasil perkalian matriks A dan B). Dua loop digunakan untuk mencetak setiap elemen matriks C, setiap baris dipisahkan oleh newline (`\n`).

Pada baris [41], terdapat `return 0;` mengembalikan nilai 0 sebagai tanda bahwa program berhasil dijalankan tanpa error. Ini merupakan bagian akhir dari fungsi main dan menandakan akhir dari eksekusi program.

Pada baris [14, 15, 21, 22, 29, 30, 31, 37, 39, dan 42], terdapat tanda `}` yang berfungsi untuk menandai akhir blok kode.

Bahasa Python:

Pada baris [1], terdapat `n = int(input("Masukkan ordo matriks nxn: "))` artinya program meminta untuk memasukkan ordo matriks $n \times n$. `input()` mengambil input berupa string, lalu `int()` mengonversi menjadi bilangan bulat dan menyimpannya dalam variabel `n`.

Pada baris [2 – 5], terdapat `A = []` `print("Matriks A")` `for i in range(n):` `A.append(list(map(int, input().split())))` artinya Matriks A dideklarasikan sebagai list kosong (`A = []`). Kemudian memasukkan elemen-elemen matriks A baris per baris. `input().split()` memisahkan input berdasarkan spasi. `map(int, ...)` mengonversi elemen-elemen string tersebut menjadi integer. `list()` digunakan untuk mengubah hasil `map` menjadi list yang mewakili satu baris matriks.

Pada baris [7 – 10], terdapat `B = []` `print("Matriks B")` `for i in range(n):` `B.append(list(map(int, input().split())))` artinya Matriks B dideklarasikan sebagai list kosong (`B = []`). Kemudian memasukkan elemen-elemen matriks B baris per baris. `input().split()` memisahkan input berdasarkan spasi. `map(int, ...)` mengonversi elemen-elemen string tersebut menjadi integer. `list()` digunakan untuk mengubah hasil `map` menjadi list yang mewakili satu baris matriks.

Pada baris [12], terdapat `C = [[0] * n for _ in range(n)]` artinya Matriks C adalah hasil perkalian matriks A dan B berukuran $n \times n$.

Pada baris [14 – 17], terdapat `for i in range(n): for j in range(n): for k in range(n): C[i][j] += A[i][k] * B[k][j]` artinya tiga loop `for` digunakan untuk melakukan perkalian matriks: `i` adalah indeks baris matriks `A` dan `C`. `j` adalah indeks kolom matriks `B` dan `C`. `k` digunakan untuk melakukan perkalian elemen-elemen dari baris `i` pada matriks `A` dan kolom `j` pada matriks `B`. Matriks `C[i][j]` dihitung dengan menjumlahkan hasil perkalian elemen `A[i][k]` dan `B[k][j]` untuk setiap `k` dari 0 hingga `n-1`.

Pada baris [19 – 21], terdapat `print("Matriks AXB") for row in C: print("".join(map(str, row)))` artinya program mencetak matriks hasil perkalian `A * B`, yaitu matriks `C`. Setiap baris dalam matriks `C`, elemen-elemen baris tersebut diubah menjadi string menggunakan `map(str, row)` dan digabungkan dengan spasi sebagai pemisah menggunakan `"".join(...)`.

RANGKUMAN

- **MODUL 0: HOW TO PROGRAM**

Bahasa C dan Python sama-sama bisa menampilkan output ke layar, tetapi dengan cara yang berbeda. Di Bahasa C harus memakai `#include <stdio.h>` supaya bisa menggunakan fungsi seperti `printf()`. Sedangkan di Python langsung bisa memakai `print()` tanpa menambahkan yang lain. Bahasa C lebih rumit karena harus memakai `int main()` sebagai fungsi utama, memakai `{ }` untuk menandai awal dan akhir kode, dan titik koma `;` di setiap akhir baris. Sedangkan Python lebih simpel, cukup tulis perintahnya langsung tanpa tanda-tanda khusus.

- **MODUL 1: VARIABLE, TIPE DATA, DAN OPERATOR**

Variabel adalah tempat untuk menyimpan nilai dalam program dengan jenis nilai yang berbeda, seperti `int` untuk bilangan bulat dan `float` untuk desimal. Operator digunakan untuk melakukan operasi pada variabel, seperti penjumlahan (`+`), pengurangan (`-`), dan perbandingan (`<`, `>`). Di Bahasa C, tipe data variabel harus dideklarasikan terlebih dahulu, sedangkan di Python, tipe data ditentukan otomatis.

- **MODUL 2: INPUT, DAN OUTPUT**

Input digunakan untuk menerima data, sedangkan output digunakan untuk menampilkan hasil ke layar. Di Bahasa C, input biasanya menggunakan fungsi `scanf()` untuk membaca data dan output menggunakan `printf()` untuk menampilkan hasil. Di **Python**, input menggunakan fungsi `input()` untuk menerima data dan output menggunakan `print()` untuk menampilkan hasil.

- **MODUL 3: KONDISIONAL**

Kondisional di Bahasa C menggunakan perintah `if`, `else if`, dan `else` untuk memeriksa suatu kondisi dengan penggunaan tanda kurung `()` dan kurung kurawal `{ }` untuk mengelompokkan blok kode yang dijalankan. Di Python, menggunakan perintah yang sama seperti `if`, `elif`, dan `else`, namun tanpa tanda kurung dan kurung kurawal.

- **MODUL 4: LOOP**

loop digunakan untuk menjalankan perintah secara berulang. Ada beberapa jenis loop, seperti for, while, dan do-while. Di Bahasa C dan Python, loop digunakan untuk mengulang suatu blok kode hingga kondisi tertentu terpenuhi. Di C, penulisan loop biasanya membutuhkan tanda kurung dan kurung kurawal, sedangkan di Python, loop ditandai dengan indentasi.

- **MODUL 5: FUNGSI**

Fungsi adalah blok kode yang dirancang untuk menyelesaikan tugas dan dapat dipanggil berulang kali dalam program. Fungsi menerima input (parameter) dan menghasilkan output. Di Bahasa C, fungsi harus dideklarasikan dengan tipe data tertentu, sedangkan di Python, fungsi didefinisikan dengan kata kunci def dan tidak perlu mendeklarasikan tipe data. Fungsi membantu membuat kode lebih terstruktur dan mudah dipelihara.

- **MODUL 6: ARRAY**

Array adalah struktur data yang digunakan untuk menyimpan beberapa nilai dalam satu variabel. Semua elemen dalam array memiliki tipe data yang sama dan bisa diakses menggunakan indeks. Di Bahasa C, array harus dideklarasikan dengan ukuran yang tetap, seperti: `int angka [5] = {1, 2, 3, 4, 5};` Sedangkan di Python, kita menggunakan list yang lebih fleksibel, karena ukurannya bisa berubah dan bisa menyimpan berbagai tipe data, contohnya: `angka = [1, 2, 3, 4, 5]`

TAUTAN GIT

<https://github.com/nazlasalsabilaa/Pemograman-1.git>