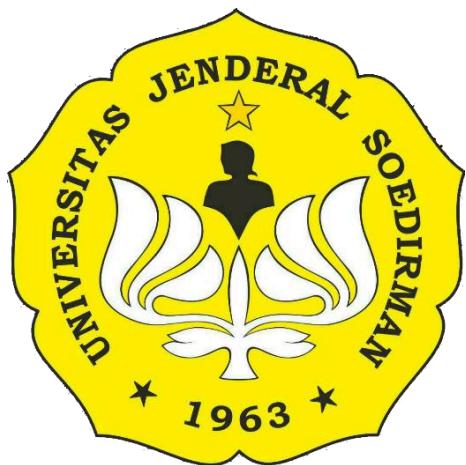


MODUL PRAKTIKUM

PEMROGRAMAN



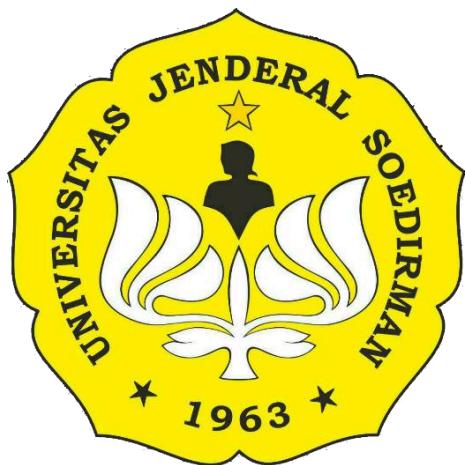
Acep Taryana, S.Si., M.T.
Hari Siswantoro, S.T., M.T., Ph.D.
Ari Fadli, S.T., M.Eng.
Yogi Ramadhani, S.T., M.Eng

PROGRAM STUDI TEKNIK ELEKTRO
UNIVERSITAS JENDERAL
SOEDIRMAN PURWOKERTO
KAMPUS PURBALINGGA

2025

MODUL PRAKTIKUM

PEMROGRAMAN



Acep Taryana, S.Si., M.T.
Hari Siswantoro, S.T., M.T., Ph.D.
Ari Fadli, S.T., M.Eng.
Yogi Ramadhani, S.T., M.Eng

PROGRAM STUDI TEKNIK ELEKTRO
UNIVERSITAS JENDERAL
SOEDIRMAN PURWOKERTO
KAMPUS PURBALINGGA

2025

DAFTAR ISI

UNIT I IDENTIFIER, TIPE DATA DAN OPERATOR.....	1
1.1. Tujuan Praktikum.....	1
1.2. Dasar Teori.....	1
1.2.1. Fungsi.....	1
1.2.2. Fungsi main().....	2
1.2.3. Fungsi Output.....	2
1.2.4. Fungsi Input.....	4
1.2.5. Praprosesor Include.....	7
1.2.6. Pengenal.....	7
1.2.7. Kata Kunci (<i>Reserved Word</i>).....	7
1.2.8. Tipe Data Dasar.....	8
1.3. Alat dan Bahan.....	10
1.4. Percobaan dan Pengamatan.....	11
1.4.1. Menghitung masukan.....	11
1.4.2. Menghitung nilai tegangan.....	12
1.4.3. Menghitung Keliling dan Luas Lingkaran.....	14
1.4.4. Tipe Data.....	16
1.5. Pertanyaan.....	19
UNIT II PENGAMBILAN KEPUTUSAN.....	20
2.1 Tujuan Praktikum.....	20
2.2. Dasar Teori.....	20
2.2.1. Operator Relasi.....	20
2.2.2. Operator Logika.....	20
2.2.3. Prioritas Operator Logika dan Relasi.....	21
2.2.4. Pernyataan Percabangan dalam Bahasa C Pernyataan if.....	21
2.2.5. Pernyataan Percabangan dalam Python.....	23
2.2.6. Pernyataan Switch.....	24
2.3. Alat dan Bahan.....	25
2.4. Pengamatan dan Percobaan.....	25
2.4.1. Pernyataan tunggal.....	25
2.4.2. Pernyataan bertingkat.....	27

2.4.3. Nested if (if bersarang).....	28
2.4.4. Switch Sederhana.....	30
2.4.5. “Switch” – contoh kasus: kalkulator sederhana.....	32
2.5. Pertanyaan.....	33
UNIT III: PERULANGAN.....	35
3.1. Tujuan Praktikum.....	35
3.2. Dasar Teori.....	35
3.2.1. Pernyataan do-while.....	35
3.2.2. Pernyataan While.....	36
3.2.3. Pernyataan For.....	37
3.2.4. Pernyataan Break dan Goto.....	38
3.3. Alat dan Bahan.....	38
3.4. Percobaan dan Pengamatan.....	39
3.4.1. Penjumlahan Beberapa bilangan.....	39
3.4.2. For.....	40
3.4.3. While do.....	41
3.4.4. Do While – lagu anak ayam.....	43
3.5. Pertanyaan.....	45
UNIT IV: ARRAY DAN STRING.....	46
4.1. Tujuan Praktikum.....	46
4.2. Dasar Teori.....	46
4.2.1. Array Satu Dimensi.....	46
4.2.2. Array Dua Dimensi.....	46
4.2.3. Mendeklarasikan variable array.....	47
4.2.4. Mendefinisikan Variabel Array.....	48
4.2.5. Pengaksesan Elemen Array.....	49
4.2.6. Struktur.....	51
4.3. Alat dan Bahan.....	53
4.4. Percobaan dan Pengamatan.....	53
4.4.1. Percobaan 1 ‘Array 1 Dimensi’.....	53
4.4.2. Percobaan 2 ‘Array 2 Dimensi’	56
4.4.3. Percobaan 3 ‘Matriks’	60

4.5. Data Pengamatan.....	61
4.5.1. Percobaan 1 Array 1 Dimensi.....	61
4.5.2. Percobaan 2 Array 2 Dimensi.....	62
4.5.3. Percobaan 3 Matriks.....	62
4.6. Pertanyaan.....	62
UNIT V FUNGSI DAN PROSEDUR.....	63
5.1. Tujuan Praktikum.....	63
5.2. Dasar Teori.....	63
5.2.1. Tujuan fungsi.....	63
5.2.2. Fungsi Dasar.....	63
5.2.3. Cara melewatkkan parameter terhadap fungsi.....	64
5.3. Alat dan Bahan.....	65
5.4. Pengamatan dan Percobaan.....	65
5.4.1. Fungsi create titik.....	65
5.4.2. Implementasi prosedur dan fungsi dalam program jam.....	68
5.5. Pertanyaan.....	71
UNIT VI: DASAR ALGORTIMA DAN STRUKTUR DATA PYTHON.....	72
6.1. Tujuan Praktikum.....	72
6.2. Dasar Teori.....	72
6.3. Alat dan Bahan.....	72
6.4 Langkah Kerja.....	73
6.4.1 Function.....	73
6.4.2 List.....	75
6.4.3 Tuples, Sets, Lambda, & List Searching.....	79
6.4.4 Recursive dengan Python.....	81
6.4.5 Object Oriented Programming dengan Python.....	83
6.4.5 Double Linked List & Object Relations.....	66
6.5 Pertanyaan.....	68
UNIT VII: CLASS GERBONG.....	69
7.1 Tujuan Praktikum.....	69
7.2 Dasar Teori.....	69
7.2.1 Pengenalan Konsep Class.....	69

7.2.2 Penciptaan dan Pemusnahan Objek Tanpa Menggunakan Pointer.....	70
7.2.3 Penciptaan Objek Dengan Menggunakan Pointer.....	70
7.3 Alat dan Bahan.....	70
7.4 Percobaan dan Pengamatan.....	70
7.5 Pertanyaan.....	77
UNIT VIII: MODIFIKASI DAN VISUALISASI CLASS GERBONG.....	78
8.1 Tujuan Praktikum.....	78
8.2 Dasar Teori.....	78
8.2.1 Deskripsi Class Secara Visual.....	78
8.3 Alat dan Bahan.....	79
8.4 Percobaan dan Pengamatan.....	79
8.5 Pertanyaan.....	83
UNIT IX : MANIPULASI STRUKTUR DATA KERETA API.....	84
9.1 Tujuan Praktikum.....	84
9.2 Dasar Teori.....	84
9.3 Alat dan Bahan.....	84
9.4 Percobaan dan Pengamatan.....	84
9.5 Pertanyaan.....	87

UNIT I IDENTIFIER, TIPE DATA DAN OPERATOR

1.1. Tujuan Praktikum

1. Mengenal Tipe Data Dasar (Integer, Real, Character), String, Enumerasi dan Tipe Buatan
2. Mengenal dan memahami fungsi dasar dalam Bahasa C, Python, R dan Matlab.
3. Mengerti dan memahami header program Bahasa C, Python, R dan Matlab.
4. Dapat menggunakan fungsi-fungsi dasar pemrograman C, Python, R dan Matlab.
5. Mengenal dan mengerti pengenal dan Kata Kunci dalam Bahasa C, Python, R dan Matlab.
6. Mengenal dan mengerti Variabel, Tipe Data dalam Bahasa C, Python, R dan Matlab.
7. Dapat menggunakan Variabel dan Tipe Variabel dalam program.

1.2. Dasar Teori

1.2.1. Fungsi

A. Bahasa C

Program C pada hakekatnya tersusun atas sejumlah blok fungsi. Sebuah program minimal mengandung sebuah fungsi. Setiap fungsi terdiri dari satu atau beberapa pernyataan yang secara keseluruhan dimaksudkan untuk melaksanakan tugas khusus. Setiap pernyataan wajib diakhiri dengan tanda titik-koma.

Secara umum, fungsi mempunyai bentuk sebagai berikut

```
nama-fungsi (daftar parameter)
deklarasi parameter;
{
    bagian isi fungsi
}
```

B. Python

Python adalah bahasa pemrograman tujuan umum yang tergolong bahasa tingkat tinggi. Dibuat oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, filosofi desain Python menekankan keterbacaan kode dengan penggunaan spasi putih (*whitespace*) yang signifikan. Konstruksi bahasanya dan pendekatan berorientasi objek bertujuan untuk membantu program menulis kode yang jelas dan logis untuk proyek skala kecil dan besar.

Python mendukung beberapa paradigma pemrograman, termasuk pemrograman terstruktur, berorientasi objek, dan fungsional. Python sering dideskripsikan sebagai bahasa "*battery-included*" karena perpustakaan standarnya yang komprehensif. Aturan pada Python dapat dibilang cukup sederhana. Dalam

penulisan satu pernyataan/perintah tidak perlu diakhiri dengan tanda titik-koma/*semicolon* (;).

C. Matlab

MATLAB adalah sebuah bahasa dengan kinerja tinggi untuk komputasi masalah teknik. MATLAB mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah untuk pakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar.

D. Bahasa R

R adalah bahasa pemrograman dan perangkat lunak gratis untuk komputasi statistik dan grafik. Bahasa R banyak digunakan oleh para ahli statistik dan penambang data (*data miner*) untuk mengembangkan perangkat lunak statistik dan analisis data. R dan *library*-nya menerapkan berbagai teknik statistik dan grafis, termasuk pemodelan linier dan nonlinier, uji statistik klasik, analisis spasial dan *time-series*, klasifikasi, pengelompokan, dan lain-lain. R mudah dikembangkan melalui fungsi dan ekstensi, dan komunitas R terkenal karena kontribusi aktifnya dalam hal *packages* atau fitur tambahan pada suatu perangkat lunak tertentu yang membuat bahasa R mudah untuk dipakai hampir dimana saja.

1.2.2. Fungsi main()

Fungsi **main()** harus ada pada program bahasa C, sebab fungsi inilah yang menjadi titik awal dan akhir eksekusi program. Tanda { diawal fungsi menyatakan awal tubuh fungsi dan juga sebagai awal program dijalankan, sedangkan tanda } di akhir fungsi merupakan akhir bagian isi fungsi dan sekaligus adalah akhir dari eksekusi program.

Pada python fungsi main ditulis dengan **def main():** dengan format pemanggilan fungsi :

```
If__name__ == "__main__":
```

```
    main()
```

Fungsi ini lebih sering digunakan pada bahasa C atau C++. Penggunaan pembatas kurung kurawal/*curly bracket* "{}" juga hampir tidak dibutuhkan oleh bahasa Python, Matlab, dan R. Pada bahasa-bahasa tersebut umumnya tanda batas program ditandai menggunakan *whitespace* berupa TAB, atau sekadar spasi.

1.2.3. Fungsi Output

A. Bahasa C

Fungsi **printf()** merupakan fungsi yang umum dipakai untuk menampilkan suatu keluaran pada layar peraga. Dalam bentuk umum, format **printf()** adalah sebagai berikut :

```
printf( kontrol string , daftar pernyataan);
```

String kontrol dapat berupa satu atau sejumlah karakter yang akan ditampilkan ataupun berupa penentu format yang akan mengatur penampilan dari argumen yang terletak pada **daftar argumen**. Mengenai penentu format diantaranya berupa :

- %d Untuk menampilkan bilangan bulat (integer)
- %f Untuk menampilkan bilangan pecahan
- %c Untuk menampilkan sebuah karakter
- %s Untuk menampilkan sebuah string

B. Python

Pada python menggunakan fungsi **print()**. Fungsi **print()** berfungsi untuk mencetak atau menampilkan objek ke perangkat keluaran (layar) atau ke file teks. Contoh penulisan”

```
print("teks")
```

C. Matlab

Pada matlab, untuk fungsi output dapat menggunakan dua fungsi yaitu **disp()** dan **fprintf()**.

1. Fungsi disp()

Syntax disp adalah syntax sederhana yang digunakan untuk menampilkan string dan output tanpa nama variabel yang mewakili output yang ditampilkan.

```
disp(nama_variabel) atau disp('teks')
```

2. Fungsi fprintf()

Perintah output fprintf dapat digunakan untuk menampilkan output pada command window.

```
fprintf('teks')
```

Untuk menampilkan output gabungan kontrol string dan variabel dapat menggunakan

```
fprintf('.....%g.....%g.%f',variabel1,variabel2,variabeln)
```

D. Bahasa R

Pada Bahasa R menggunakan fungsi yang mirip dengan python yaitu ***print()*** untuk menampilkan teks ke layar atau nilai variabel. Contoh penulisan

```
print("teks") atau print(nama_variabel)
```

1.2.4. Fungsi Input

A. Bahasa C

Fungsi ini digunakan untuk memasukan berbagai macam jenis data. Misalnya untuk memasukan data jari-jari lingkaran pada program menghitung luas lingkaran.

```
scanf("f", &radius)
```

Bentuk scanf sesungguhnya menyerupai bentuk fungsi printf. Fungsi ini melibatkan penentu format pada dasarnya sama digunakan pada printf(). Secara umum bentuk scanf adalah sebagai berikut :

```
scanf("string control", daftar_argumen);
```

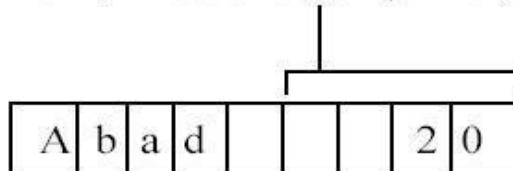
Dengan string control dapat berupa :

- A. Penentu format
- B. Karakter spasi putih
- C. Karakter bukan spasi putih (*whitespace*)

Scarf ("f", &radius) berarti bagi komputer adalah sebuah bilangan real (%f) ditempatkan ke alamat dari radius (&radius). Untuk menentukan panjang medan yang disediakan bagi tampilan data, maka sesudah tanda % dalam penentuan format dapat disisipi dengan bilangan bulat yang menyatakan panjang medan

Untuk data yang berupa bilangan bulat misalnya

```
printf("Abad %4d", 20);
```



Untuk data berupa bilangan real

m.n

m = panjang medan

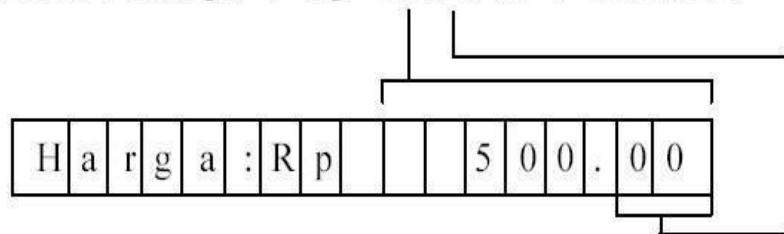
n = jumlah digit pecahan

Contoh pernyataan :

```
printf("Harga : Rp. %8.2f\n", 500.0)
```

Dimana %8.2f menyatakan panjang medan dari bilangan real yang akan ditampilkan adalah 8 karakter dengan digit bilangan pecahan sebanyak 2 buah

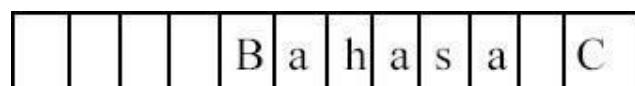
```
printf("Harga : Rp %8.2f\n", 500.0);
```



Atau dapat pula hanya menyertakan banyaknya digit pecahan
Printf("Harga : Rp. %.2f\n", 500.0)

Untuk data string
Printf("%12s", "Bahasa C")

Maka akan ditampilkan sebagai berikut :



Tampak pada gambar tersebut diatas , penentu format yang mengandung panjang medan, secara default akan menampilkan panjang data dalam bentuk rata kanan terhadap panjang medan yang diberikan. Untuk data string yang biasa dikehendaki untuk ditampilkan dalam bentuk rata kiri maka sesudah tanda % disisipi tanda minus (-).

```
printf("%12s", "Bahasa C")
```

Maka akan tampak

B	a	h	a	s	a		C			
---	---	---	---	---	---	--	---	--	--	--

B. Python

Python sudah menyediakan fungsi `input()` dan `raw_input()` untuk mengambil inputan dari keyboard. Cara pakainya:

```
nama_variabel = input("Sebuah Teks")
```

Artinya, teks yang kita inputkan dari keyboard akan disimpan ke dalam ***nama_variabel***. Apa perbedaan fungsi `input()` dengan `raw_input()`?

Fungsi `input()` digunakan untuk mengambil data angka. Sedangkan `raw_input()` untuk mengambil teks. Pada Python3 cukup menggunakan fungsi `input()` saja, karena fungsi `raw_input()` sudah digabungkan di sana.

C. Matlab

Dengan menggunakan fungsi `input`, kita perlu mendefinisikan variabel dengan nilai sebagai fungsi `input`. Sehingga nilai variabel tersebut akan didefinisikan melalui command windows setelah program script file kita dijalankan. Fungsi Input dapat dibuat dengan bentuk :

```
nama_variabel = input('komentar yang ditampilkan  
perintah input')
```

D. Bahasa R

Seperti bahasa pemrograman lain di R, juga dimungkinkan untuk mengambil input dari pengguna. Untuk melakukannya, ada dua metode di R :

- Menggunakan metode ***readline()***

Dalam bahasa R metode `readline()` mengambil input dalam format string. Jika ada yang menginput integer kemudian diinput sebagai string, misalkan ada yang ingin input 255, maka akan diinput sebagai "255", seperti string. Jadi seseorang perlu mengkonversi nilai yang dimasukkan ke format yang dia butuhkan. Dalam hal ini, string "255" diubah menjadi integer 255. Untuk mengubah nilai yang diinputkan menjadi tipe data yang diinginkan, terdapat beberapa fungsi pada R.

1. ***as.integer(n)***; —> convert to integer
2. ***as.numeric(n)***; —> convert to numeric type (float, double etc)
3. ***as.complex(n)***; —> convert to complex number (i.e 3+2i)
4. ***as.Date(n)*** —> convert to date

- Menggunakan metode ***scan()***

Cara lain untuk mengambil input pengguna dalam bahasa R adalah menggunakan metode, yang disebut metode `scan()`. Metode ini mengambil input dari konsol. Metode ini adalah metode yang sangat berguna sementara input diperlukan untuk diambil dengan cepat untuk perhitungan matematis apa pun atau untuk kumpulan data apa pun. Metode ini membaca data dalam bentuk vektor atau daftar. Metode ini juga digunakan untuk membaca input dari file juga.

```
x = scan()
```

`scan()` metode mengambil input terus menerus, untuk menghentikan proses input, perlu menekan tombol Enter 2 kali pada konsol.

1.2.5. Praprosesor Include

`#include` merupakan salah satu jenis pengarah praprosesor yang dipakai untuk membaca file yang dinamakan file-judul (**header file**). File-judul yaitu file yang diantaranya berisi deklarasi fungsi dan definisi konstanta. File ini mempunyai ciri yaitu nama file-nya diakhiri dengan extensi .h , misalnya = stdio.h. Praprosesor Include **hanya** digunakan pada bahasa C.

1.2.6. Pengenal

Pengenal (*identifier*) merupakan nama yang biasa digunakan untuk variabel, konstanta, fungsi atau obyek yang lain yang didefinisikan oleh pemrogram. Aturan yang berlaku bagi pengenal :

- Pengenal haruslah diawali dengan huruf (A..Z, a..z) atau karakter garis bawah (_). Selanjutnya dapat berupa huruf, digit (0..9) atau karakter garisbawah atau tanda dollar (\$).
- Panjang pengenal boleh terdiri lebih dari 32 karakter, akan tetapi hanya 32 karakter pertama yang akan dianggap berarti.

1.2.7. Kata Kunci (*Reserved Word*)

Untuk menghindari kesalahan, pengenal tidak boleh menggunakan nama yang tergolong sebagai kata kunci. Adapun kata kunci tersebut adalah sebagai berikut :

asm	do	if	signed	volatile
auto	double	int	sizeof	while
break	else	interrupt	static	_cs
case	enum	long	struct	_ds
cdecl	extern	near	switch	_es
char	far	pascal	typedef	_ss
const	float	register	union	
continue	for	return	unsigned	
default	goto	short	void	
	huge			

1.2.8. Tipe Data Dasar

A. Bilangan bulat

1. Merupakan bilangan yang tidak mengandung bilangan pecahan desimal. Tipe untuk bilangan ini adalah Integer.
2. Terbagi atas beberapa kategori seperti terlihat dalam tabel yang menunjukkan jenis data, ukuran dalam memori dan rentang nilainya.

Tipe Data	Ukuran	Jangkauan
Byte 1 byte 0 s/d +255	Byte 1 byte 0 s/d +255	Byte 1 byte 0 s/d +255
Shortint 1 byte	28 s/d +127	Shortint 1 byte
Integer 2 bytes	32768 s/d 32767	Integer 2 bytes
Word 2 bytes 0 s/d 65535	Word 2 bytes 0 s/d 65535	Word 2 bytes 0 s/d 65535
Longint	Longint 4 bytes 2147483648 s/d 2147483647	Longint 4 bytes 2147483648 s/d 2147483647

Penggolongan tipe data integer tersebut dimaksudkan untuk membatasi alokasi memori yang dibutuhkan misalkan untuk suatu perhitungan dari suatu variabel bilangan diperkirakan nilai maksimumnya 32767 kita cukup mendeklarasikan variabel bilangan sebagai integer (2 byte), daripada sebagai longint (4 byte). Tipe bilangan bulat adalah tipe yang memiliki keterurutan.

Ini berarti, jika sebuah nilai bilangan bulat diketahui, nilai sebelumnya (*predecessor*) dan nilai sesudahnya (*successor*) dapat ditentukan. Contoh predecessor dari 8 adalah 7 sedangkan successornya adalah 9.

B. Bilangan Riil

Merupakan bilangan yang mengandung pecahan desimal. Harus mengandung titik. Tipe bilangan riil ini adalah real. Dalam Turbo Pascal tipe real direpresentasikan ke dalam 4 macam tipe yaitu real, single, double dan extended. Rentang nilai positif untuk keempat macam tipe tersebut adalah :

Tipe Data	Ukuran	Jangkauan
Real	6 bytes	2.9×10^{-39} s/d 1.7×10^{38}
Single	4 bytes	1.5×10^{-45} s/d 3.4×10^{38}
Double	8 bytes	5.0×10^{-324} s/d 1.7×10^{308}
Extended	10 bytes	3.4×10^{-4932} s/d 1.1×10^{4932}
Comp	8 bytes	-9.2×10^{18} s/d 9.2×10^{18}

C. Karakter

Tipe data ini menyimpan karakter yang diketikkan dari keyboard, memiliki 255 macam yang terdapat dalam tabel ASCII (*American Standard Code for Information Interchange*). Untuk tipe data karakter penulisannya harus diapit oleh tanda petik tunggal.

D. Bilangan Logika

Tipe data logika hanya mengenal dua buah nilai yaitu : **benar** (true) atau **salah** (false). Tipe boolean memakai memori paling kecil, sedangkan WordBool dan LongBool dipakai untuk menulis program yang sesuai dengan lingkungan Windows.

Tipe Data	Ukuran Tempat
Boolean	1 byte
WordBool	2 byte
Longbool	3 byte
Boolean	1 byte
WordBool	2 byte

Operasi-operasi yang dapat dilakukan untuk terhadap bilangan boolean dikenal dengan operasi logika operator yang dapat digunakan untuk operasi logika adalah NOT, AND, OR dan XOR.

E. Tipe data struktur dan tipe bentukan String

Merupakan suatu data yang menyimpan array (larik), sebagai contoh 'ABCDEF' merupakan sebuah konstanta string yang berisikan 6 byte karakter. Ukuran Tempat untuk tipe data ini adalah 2 s/d 256 byte, dengan jumlah elemen 1 s/d 255. Pada Pascal string dideklarasikan dengan string [konstanta] atau string. Bila ukuran string tidak didefinisikan maka akan banyak memakan ruang, karena ukuran string menyesuaikan dengan defaultnya.

Tuple

Tuple dalam Python adalah struktur data yang digunakan untuk menyimpan sekumpulan data. Tuple bersifat immutable, artinya isi tuple tidak bisa kita ubah dan hapus. Namun, dapat kita isi dengan berbagai macam nilai dan objek.

Set

Set adalah kumpulan tipe data tidak berurutan yang dapat diubah, berubah, dan tidak memiliki elemen duplikat. Urutan elemen dalam suatu himpunan tidak terdefinisi meskipun dapat terdiri dari berbagai elemen. Keuntungan utama menggunakan set adalah bahwa ia memiliki metode yang sangat dioptimalkan untuk memeriksa apakah elemen tertentu terkandung dalam set.

Dictionary

Dictionary di Python adalah kumpulan nilai data yang tidak berurutan, digunakan untuk menyimpan nilai data seperti peta, yang, tidak seperti Tipe Data lain yang hanya menyimpan satu nilai sebagai elemen, Dictionary menyimpan pasangan **key:value**. Nilai kunci disediakan dalam Dictionary untuk membuatnya lebih optimal.

Pointer

Pointer merupakan variabel khusus yang berisi suatu address (alamat) di lokasi lain di dalam memori. Suatu variabel yang points(menunjuk) ke sesuatu sehingga disebut pointer. Ada dua macam pointer :

- Typed (tertentu) merupakan pointer yang menunjuk pada tipe data tertentu pada variable.
- Generic(umum) merupakan pointer yang tidak menunjuk pada tipe data tertentu pada variable.

Tipe-tipe data diatas bisa dioperasikan dengan sejumlah operasi yaitu :

o Operasi Aritmatika

Operasi aritmatika yang berlaku pada bilangan bulat adalah

Operator	Arti	Prioritas
$^$, **	Pangkat	1
*	Kali	2
/	Bagi (bilangan Real)	2
\	Bagi (bilangan Bulat)	3
mod	Sisa hasil bagi	4
+	Tambah	5
-	Kurang	6
Operator	Arti	Prioritas

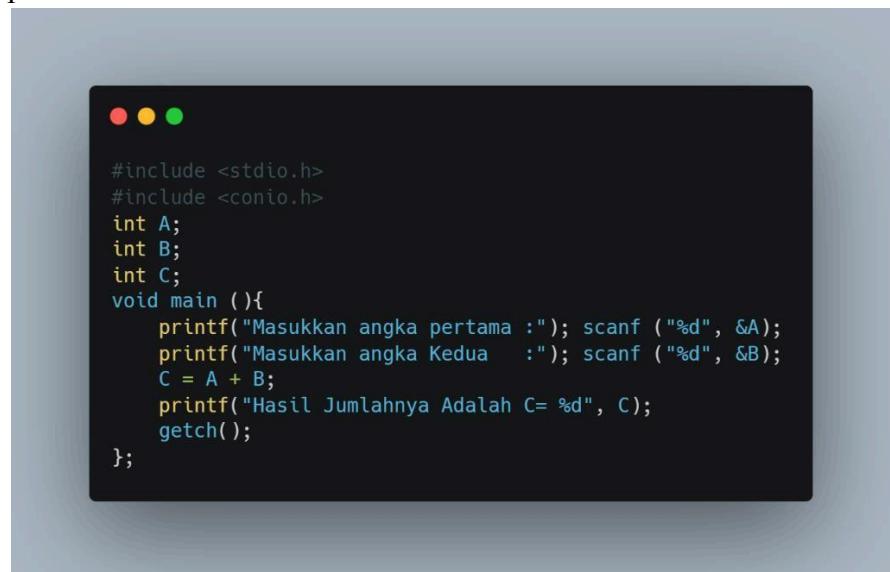
1.3. Alat dan Bahan

1. Seperangkat Komputer
2. Interpreter Python 3
3. Interpreter bahasa R
4. Software Compiler C (ex : Borland C/C++ atau Dev C/C++)
5. Software Compiler Python (ex: VSCode)
6. Software Compiler Matlab (ex: Octave atau Matlab)
7. Software RStudio

1.4. Percobaan dan Pengamatan

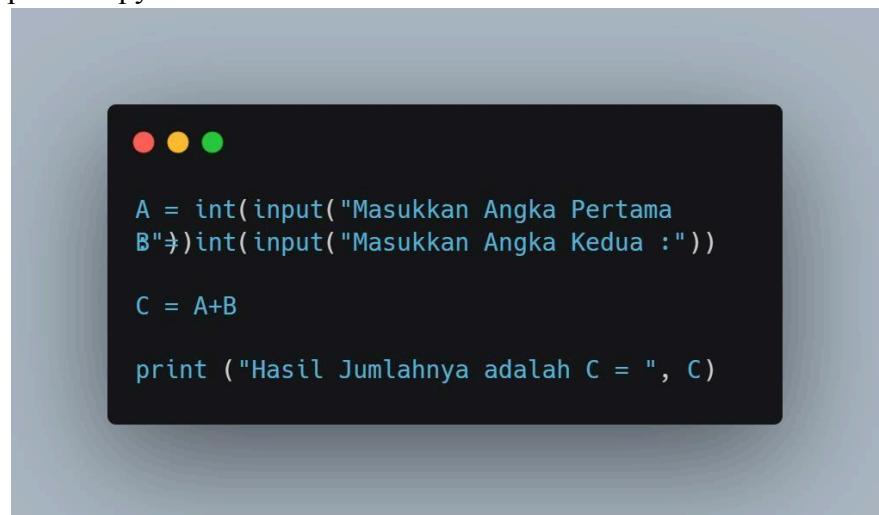
1.4.1. Menghitung masukan

1. Nyalakan komputer yang ada dihadapan anda, biarkan sampai masuk kedalam sistem operasi windows
2. Bukalah program kompiler bahasa C, Python, Matlab dan R yang ada dalam komputer tersebut.
3. Ketiklah kode sumber berikut ini dengan benar dan tepat
 - pertama.c



```
#include <stdio.h>
#include <conio.h>
int A;
int B;
int C;
void main (){
    printf("Masukkan angka pertama :"); scanf ("%d", &A);
    printf("Masukkan angka Kedua   :"); scanf ("%d", &B);
    C = A + B;
    printf("Hasil Jumlahnya Adalah C= %d", C);
    getch();
};
```

- pertama.py



```
A = int(input("Masukkan Angka Pertama\nB"))int(input("Masukkan Angka Kedua :"))

C = A+B

print ("Hasil Jumlahnya adalah C = ", C)
```

- pertama.m

```

1 A = input('Masukkan angka pertama A : ');
2 B = input('Masukkan angka kedua B : ');
3 C = A+B;
4 fprintf('Hasil jumlahnya adalah C = %d\n',C);

```

- pertama.r

```

1 A = input('Masukkan angka pertama A : ');
2 B = input('Masukkan angka kedua B : ');
3 C = A+B;
4 printf('Hasil jumlahnya adalah C = %d\n',C);

```

4. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera di atas, kemudian simpanlah dalam folder yang diberi nama sesuai dengan NIM anda
5. Kompilasilah kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
6. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
7. Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error

1.4.2. Menghitung nilai tegangan

1. Dengan kondisi program kompiler bahasa C, Python, Matlab dan R yang masih terbuka tersebut
2. Ketiklah kode sumber berikut ini dengan benar dan tepat
 - tegangan.c

```
● ● ●  
1 #include <stdio.h>  
2 int main (){  
3     int arus, hambatan, tegangan;  
4     arus = 10;  
5     hambatan = 500;  
6     tegangan = arus*hambatan;  
7     printf("Voltase = %d", tegangan);  
8     getch();  
9     return 0;  
10 }
```

- tegangan.py

```
● ● ●  
1 arus = 10;  
2 hambatan = 500;  
3 tegangan = arus * hambatan;  
4 fprintf('Voltase = %d',tegangan);
```

- tegangan.m

```
● ● ●  
1 arus = 10;  
2 hambatan = 500;  
3 tegangan = arus * hambatan;  
4 fprintf('Voltase = %d',tegangan);
```

- tegangan.r



```

● tegangan.r
● ● ●

1 arus = 10;
2 hambatan = 500;
3 tegangan = arus * hambatan;
4 message(sprintf("Voltase = %d",tegangan));

```

3. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera di atas, kemudian simpanlah dalam folder yang diberi nama sesuai dengan NIM anda
4. Kompilasilah kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
5. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
6. Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error
7. Buatlah program dalam bahasa C, Python, Matlab dan R untuk menghitung kapasitansi, C dalam farad.
INPUT = Banyaknya muatan(Q), Tegangan(V)
PROSES = Kapasitas = Muatan / tegangan antara keping dalam volt
8. Diketahui sebuah program berikut ini :

```

main ()
{
    int jawab, hasil;
    jawab = 100;
    hasil = jawab - 10;
    printf ("jawabannya adalah %d\n", hasil + 6);
}

```

9. Berdasarkan pada langkah 9 berapakah nilai jawabannya

1.4.3. Menghitung Keliling dan Luas Lingkaran

1. Dengan kondisi program kompiler bahasa C, Python, Matlab dan R yang masih terbuka tersebut.
2. Ketiklah kode sumber berikut ini dengan benar dan tepat

- keluas.c

```
#include <stdio.h>

int main()
{
    float phi=3.14;
    float jari_jari, luas, keliling;

    jari_jari = 7.0;
    luas = phi * jari_jari * jari_jari;
    keliling = 2 * phi * jari_jari;

    printf("luas lingkaran %f\n", luas);
    printf("keliling lingkaran %f", keliling);
}
```

- keluas.py

```
phi = 3.14
jari_jari = 7.0
luas = phi * jari_jari * jari_jari
keliling = 2 * phi * jari_jari
print ("luas lingkaran ", luas)
print ("keliling lingkaran ", keliling)
```

- keluas.m

```
phi = 3.14;
jari_jari = 7.0;
luas = phi * jari_jari * jari_jari;
keliling = 2 * phi * jari_jari;
fprintf("luas lingkaran %d\n", luas);
fprintf("keliling lingkaran %d", keliling);
```

- keluas.r



```
phi = 3.14;
jari_jari = 7.0;
luas = phi * jari_jari * jari_jari;
keliling = 2 * phi * jari_jari;
message(sprintf("Luas lingkaran = %f", luas));
message(sprintf("keliling lingkaran = %f", keliling))
```

3. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera di atas, kemudian simpanlah dalam folder yang diberi nama sesuai dengan NIM anda
4. Kompilasilah kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
5. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
6. Lakukanlah langkah 5 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error

1.4.4. Tipe Data

1. Dengan kondisi program kompiler bahasa C, Python, Matlab dan R yang masih terbuka tersebut.
2. Ketiklah kode sumber berikut ini dengan benar dan tepat

- tipedata.c



```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

int main(){

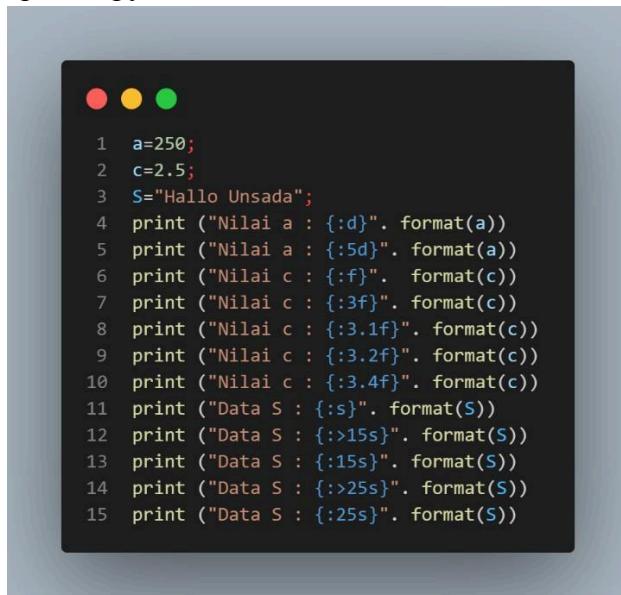
    int a;
    float c;
    char *S;

    a = 250;
    c = 2.5;
    S = "Hallo Unsada";

    printf("\nNilai a : %d",a);
    printf("\nNilai a : %5d",a);
    printf("\nNilai c : %f",c);
    printf("\nNilai c : %3f",c);
    printf("\nNilai c : %3.1f",c);
    printf("\nNilai c : %3.2f",c);
    printf("\nNilai c : %3.4f",c);
    printf("\nNilai S : %s",S);
    printf("\nData S : %15s",S);
    printf("\nData S : %-15s",S);
    printf("\nData S : %25s",S);
    printf("\nData S : %-25s",S);

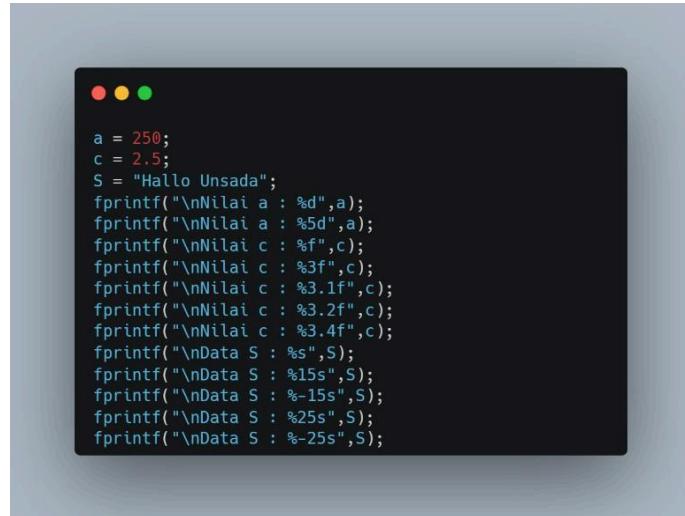
    system("pause");
    return 0;
}
```

- tipedata.py



```
1 a=250;
2 c=2.5;
3 S="Hallo Unsada";
4 print ("Nilai a : {:d}". format(a))
5 print ("Nilai a : {:5d}". format(a))
6 print ("Nilai c : {:f}". format(c))
7 print ("Nilai c : {:3f}". format(c))
8 print ("Nilai c : {:3.1f}". format(c))
9 print ("Nilai c : {:3.2f}". format(c))
10 print ("Nilai c : {:3.4f}". format(c))
11 print ("Data S : {}". format(S))
12 print ("Data S : {:>15s}". format(S))
13 print ("Data S : {:>15s}". format(S))
14 print ("Data S : {:>25s}". format(S))
15 print ("Data S : {:>25s}". format(S))
```

- tipedata.m



```
a = 250;
c = 2.5;
S = "Hallo Unsada";
fprintf("\nNilai a : %d",a);
fprintf("\nNilai a : %5d",a);
fprintf("\nNilai c : %f",c);
fprintf("\nNilai c : %3f",c);
fprintf("\nNilai c : %3.1f",c);
fprintf("\nNilai c : %3.2f",c);
fprintf("\nNilai c : %3.4f",c);
fprintf("\nData S : %s",S);
fprintf("\nData S : %15s",S);
fprintf("\nData S : %-15s",S);
fprintf("\nData S : %25s",S);
fprintf("\nData S : %-25s",S);
```

- tipedata.r



```
a = 250;
c = 2.5;
S = "Hallo Unsada";

message(sprintf("Nilai a :%d", a));
message(sprintf("Nilai a :%5d", a));
message(sprintf("Nilai c :%f", c));
message(sprintf("Nilai c :%3f", c));
message(sprintf("Nilai c :%3.1f", c));
message(sprintf("Nilai c :%3.2f", c));
message(sprintf("Nilai c :%3.4f", c));
message(sprintf("Data S :%s", S));
message(sprintf("Data S :%15s", S));
message(sprintf("Data S :%-15s", S));
message(sprintf("Data S :%25s", S));
message(sprintf("Data S :%-25s", S));
```

3. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera di atas, kemudian simpanlah dalam folder yang diberi nama sesuai dengan NIM anda
4. Kompilasi lah kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
5. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut

6. Lakukanlah langkah 5 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error
7. Modifikasi program ini, agar dapat digunakan untuk menghitung panjang sisi miring segitiga dalam bahasa C, python, matlab dan R.

1.5. Pertanyaan

1. Sebutkan dan jelaskan 3 tipe data dasar dalam pemrograman!
2. Jelaskan perbedaan syntax antara bahasa pemrograman C dan python !
3. Buatlah sebuah program bahasa C untuk menghitung energi potensial jika diketahui sebuah kelapa memiliki massa 2 kg dan berada pada ketinggian 20 meter dari permukaan tanah. Jika diketahui gravitasi bumi di tempat itu $9,81 \text{ m/s}^2$, berapa energi potensial yang dimiliki kelapa pada ketinggian itu? Dalam menjalankan programnya user harus menginputkan nilai massa dan tinggi secara manual sedangkan nilai gravitasi sudah diketahui!
4. Buatlah flowchart menghitung volume tabung di mana nilai phi = 3.14 dan variabel lainnya (jari-jari dan tinggi tabung) diinputkan oleh pengguna !
5. Buatlah flowchartnya untuk mencari sisi miring segitiga dan menampilkan karakter yang dimasukan melalui keyboard ke layar.
6. Tulis desain algoritma dan flowchartnya
 - a. Mencetak tulisan “Hello word”
 - b. Menghitung penjumlahan $144 + 256$
7. Belajar menemukan kesalahan

```

main ( )
{
    INT jumlah;

    /* Merupakan isi dari
    Jumlah = 10;
    printf ("Hasil penjumlahannya adalah");
    printf ("jawabannya adalah %d\n" Jumlah);
}

```

UNIT II PENGAMBILAN KEPUTUSAN

2.1 Tujuan Praktikum

1. Mahasiswa mengenal struktur sequence if-then Else
2. Mahasiswa mengkonstruksi algoritma sequence if-then-else
3. Mahasiswa menyusun program C/C++ yang memiliki struktur sequence, ifthen- else
4. Menjelaskan tentang pernyataan switch dalam Bahasa C dan Python.

2.2. Dasar Teori

Untuk keperluan pengambilan keputusan C dan Python menyediakan beberapa jenis pernyataan yaitu :

1. Pernyataan if
2. Pernyataan if-else
3. Pernyataan if-else if
4. Pernyataan switch

Pernyataan-pernyataan tersebut memerlukan suatu kondisi sebagai basis dalam pengambilan keputusan. Kondisi umum yang dipakai adalah benar dan salah.

2.2.1. Operator Relasi

Operator relasi biasa dipakai untuk membandingkan dua buah nilai. Hasil perbandingan berupa keadaan benar atau salah. Keseluruhan operator relasi pada C maupun Python sama, seperti yang ditunjukkan pada tabel berikut ini :

Tabel 2-0 Daftar operator relasi

Operator	Makna
>	Lebih dari
>=	Lebih dari sama dengan
<=	Kurang dari sama dengan
==	Sama dengan
!=	Tidak sama dengan
<	Kurang dari

2.2.2. Operator Logika

Operator logika biasa dipakai untuk menghubungkan ekspresi relasi. Operator logika pada C dan Python memiliki makna yang sama, tetapi menggunakan kode yang berbeda. Keseluruhan operator logika tampak pada tabel berikut ini :

Tabel 2-1 Daftar operator relasi Bahasa C

Operator	Makna
&&	Dan (AND)
	Atau (OR)
!	Tidak (NOT)

Tabel 2-2 Daftar operator relasi Python

Operator	Makna
and	Dan (AND)
or	Atau (OR)
not	Tidak (NOT)

2.2.3. Prioritas Operator Logika dan Relasi

Tertinggi	! not
	> >= < <=
	= !=
	&& and
Terendah	or

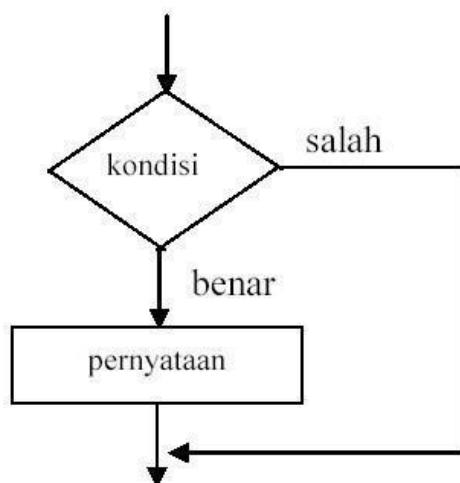
2.2.4. Pernyataan Percabangan dalam Bahasa C Pernyataan if

Bentuk umum pernyataan ini dalam Bahasa C adalah

```
if (kondisi) {
    pernyataan
}
```

Bentuk ini menyatakan :

Jika kondisi yang diseleksi bernilai logika = 1, maka pernyataan yang mengikutinya akan diproses. Sebaliknya jika yang diseleksi adalah bernilai logika = 0, maka pernyataan yang mengikutinya tidak adakan di proses. Mengenai kondisi harus ditulis diantara tanda kurung, sedangkan pernyataan dapat berupa sebuah pernyataan tunggal, pernyataan majemuk atau pernyataan kosong, bentuk diagram alirnya tampak seperti gambar 3-0 di bawah ini :



gambar 2-0 Bentuk umum kondisi if

Pernyataan if bertingkat

Dalam bahasa C Bentuk pengambilan keputusan dapat pula anda modifikasi, dengan menggunakan if, seperti tampak pada pernyataan dibawah ini :

```
If (kondisi 1)
    If (kondisi 2)
    .
    .
        If (kondisi N)
            Pernyataan
        Else
            Pernyataan
        .
        .
    Else
        Pernyataan;
Else
    Pernyataan;
```

Dimana dari pernyataan tersebut maka kondisi yang akan dicek pertama kali adalah kondisi yang paling luar yaitu kondisi 1. Jika kondisi satu bernilai salah maka statemen else yang terluarlah yang menjadi pasangan if tersebut yang akan dijalankan, ketika kondisi 1 bernilai benar maka yang dijalankan pengecekan kondisi 2 dan begitu seterusnya sampai semua kondisi terpenuhi, dan ketika salah satu kondisi if tidak terpenuhi maka yang akan dijalankan adalah else dimana pasangan if tersebut tidak terpenuhi kondisinya

Beberapa bentuk pernyataan dalam Bahasa C

Berikut ini disajikan beberapa bentuk pernyataan dari if-else-then, seperti tampak pada gambar 2-

If Tunggal	If - Jamak	Nested - If
if (kondisi) pernyataan	if (kondisi) { pernyataan-1 pernyataan-2 } else if { pernyataan-3 pernyataan-4 }	if (kondisi-1) if (kondisi-2) { pernyataan-1 } else if (kondisi-3) pernyataan-2
If - Else		
if (kondisi) pernyataan-1 else pernyataan-2		

gambar 2-1 Bentuk kondisi if-then-else

2.2.5. Pernyataan Percabangan dalam Python

Pernyataan if

Dalam percabangan, IF akan terdiri dari ekspresi boolean yang diikuti oleh satu baris atau lebih kondisi atau pernyataan. Kondisi ini bermakna jika hasilnya benar maka kondisi di dalam blok if tersebut akan dieksekusi, tetapi jika salah, maka kondisi tidak dieksekusi

Untuk bentuk pernyataan ini dalam Python adalah

```
if (kondisi):  
    pernyataan;
```

Pernyataan if-else

Pengambilan keputusan (kondisi if else) tidak hanya digunakan untuk menentukan tindakan apa yang akan diambil sesuai dengan kondisi, tetapi juga digunakan untuk menentukan tindakan apa yang akan diambil/dijalankan jika kondisi tidak sesuai. Kondisi if else adalah kondisi dimana jika pernyataan benar True maka kode dalam if akan dieksekusi, tetapi jika bernilai salah False maka akan mengeksekusi kode di dalam else.

```
if (kondisi 1):  
    Pernyataan  
else:  
    Pernyataan
```

Pernyataan if-elif

Bentuk pernyataan ini hanya dapat digunakan pada Python. Elif jika di bahasa pemrograman lain adalah else if, di python cukup dengan elif. Pengambilan keputusan (kondisi if elif) merupakan lanjutan/percabangan logika dari “kondisi if”. Dengan elif kita bisa membuat kode program yang akan menyeleksi beberapa kemungkinan yang bisa terjadi. Hampir sama dengan kondisi “else”, bedanya kondisi “elif” bisa banyak dan tidak hanya satu.

```
if (kondisi 1):
    Pernyataan
elif (kondisi 2):
    Pernyataan
```

Pernyataan if-elif-else

Percabangan tidak hanya bisa untuk dua kondisi, namun bisa untuk tiga, empat, bahkan lebih kondisi percabangan. Fungsi if-else-elif digunakan untuk lebih dari dua logika percabangan.

```
if (kondisi 1):
    Pernyataan
elif (kondisi 2):
    Pernyataan
elif (kondisi 3):
    Pernyataan
else:
    Pernyataan
```

2.2.6. Pernyataan Switch

A. Bahasa C

Pernyataan **switch** merupakan pernyataan yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan sejumlah alternatif. Bentuk umum pernyataan **switch**.

```
switch(ekspresi) {
    case konstanta-1:
        pernyataan-1
        break;
    case konstanta-2:
        pernyataan-2
        break;
    case konstanta-x:
```

```
        pernyataan-x
        break;
    default:
        pernyataan-n
}
```

B. Python

Tidak seperti bahasa pemrograman lain yang pernah kita gunakan sebelumnya, Python tidak memiliki pernyataan switch atau case. Untuk menyesati fakta ini, kami menggunakan **dictionary mapping**.

```
def
    nama_switch(ekspresi):
    switcher = {
        konstanta-1: pernyataan-1,
        konstanta-2: pernyataan-2,
        konstanta-x: pernyataan-x,
    }
    return switcher.get(ekspresi, pernyataan-n)
```

2.3. Alat dan Bahan

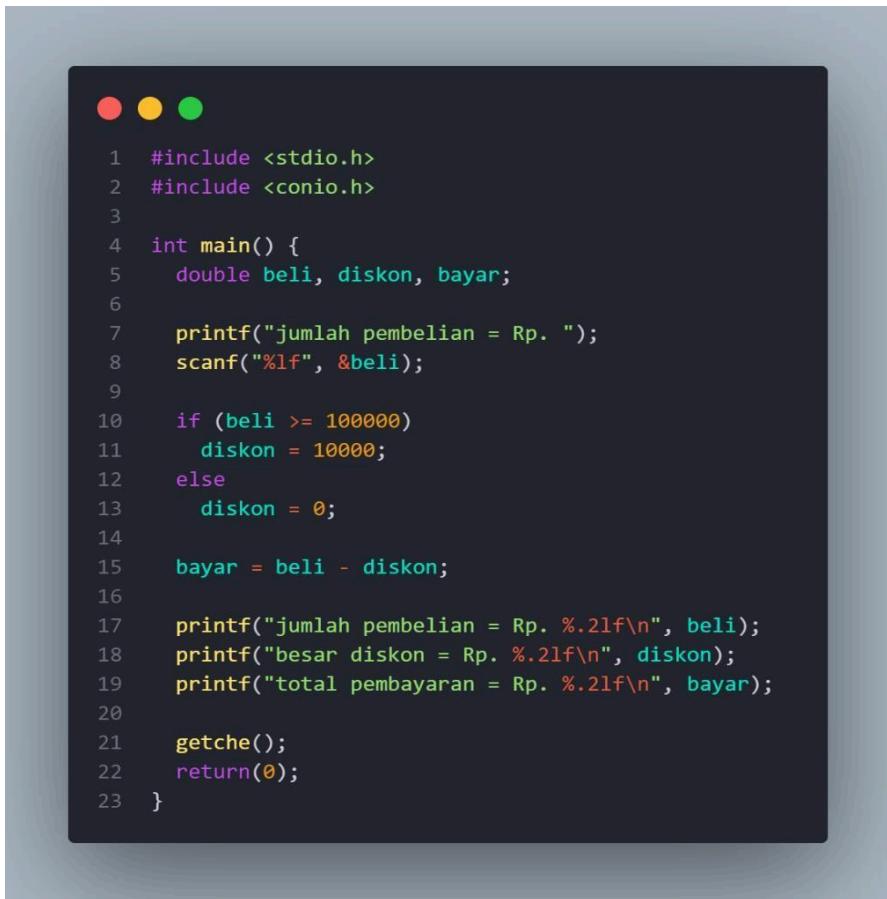
1. Seperangkat Komputer
2. Interpreter Python 3
3. Software Compiler C (ex : Borland C/C++ atau Dev C/C++)
4. Software Compiler Python (ex: VSCode)

2.4. Pengamatan dan Percobaan

2.4.1. Pernyataan tunggal

1. Nyalakan komputer yang ada dihadapan anda, biarkan sampai masuk kedalam sistem operasi windows
2. Bukalah program kompiler bahasa C dan Python yang ada dalam komputer tersebut.
3. Ketiklah kode sumber berikut ini dengan benar dan tepat

- diskon.c



```
● ● ●

1 #include <stdio.h>
2 #include <conio.h>
3
4 int main() {
5     double beli, diskon, bayar;
6
7     printf("jumlah pembelian = Rp. ");
8     scanf("%lf", &beli);
9
10    if (beli >= 100000)
11        diskon = 10000;
12    else
13        diskon = 0;
14
15    bayar = beli - diskon;
16
17    printf("jumlah pembelian = Rp. %.2lf\n", beli);
18    printf("besar diskon = Rp. %.2lf\n", diskon);
19    printf("total pembayaran = Rp. %.2lf\n", bayar);
20
21    getch();
22    return(0);
23 }
```

- diskon.py



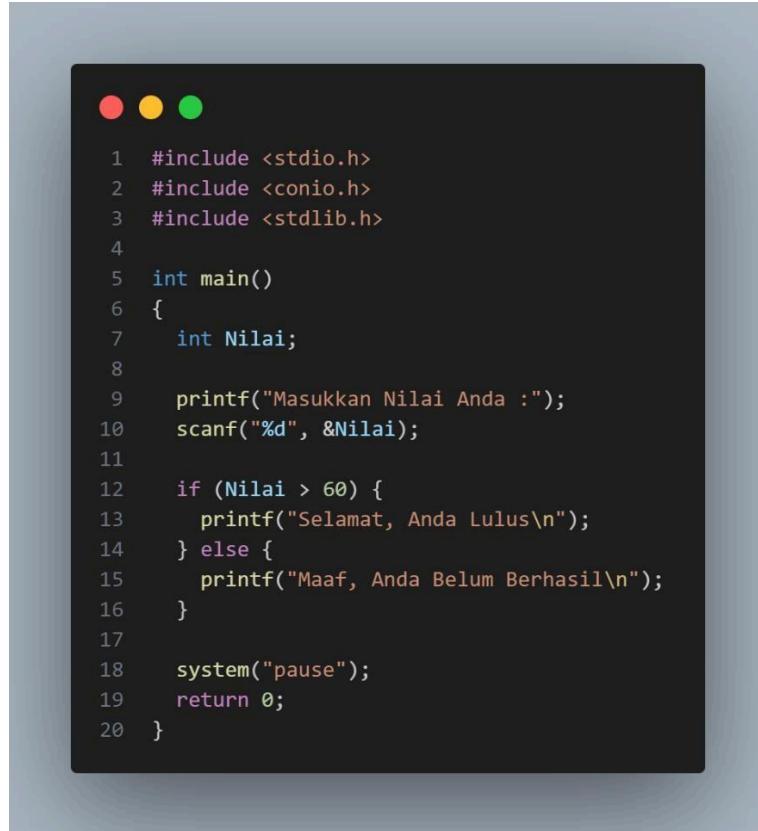
```
● ● ●

1 beli = int(input("jumlah pembelian = Rp. "))
2 if beli >= 100000:
3     diskon = 10000
4 else:
5     diskon = 0
6 bayar = beli - diskon
7
8 print("jumlah pembelian = Rp. ", beli)
9 print("jumlah diskon = Rp. ", diskon)
10 print("total pembayaran = Rp. ", bayar)
```

4. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera di atas, kemudian simpanlah dalam folder yang diberi nama sesuai dengan NIM anda
5. Kompilasi kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
6. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
7. Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error

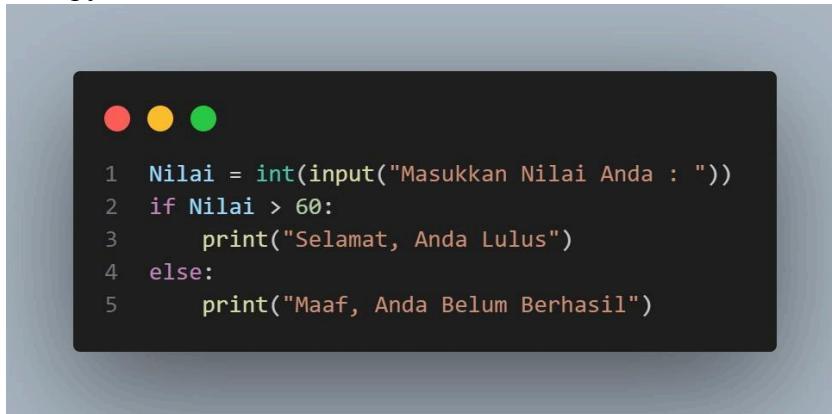
2.4.2. Pernyataan bertingkat

1. Dengan kondisi program kompiler bahasa C dan Python yang masih terbuka tersebut.
2. Ketiklah kode sumber berikut ini dengan benar dan tepat
 - nilai.c



```
1 #include <stdio.h>
2 #include <conio.h>
3 #include <stdlib.h>
4
5 int main()
6 {
7     int Nilai;
8
9     printf("Masukkan Nilai Anda :");
10    scanf("%d", &Nilai);
11
12    if (Nilai > 60) {
13        printf("Selamat, Anda Lulus\n");
14    } else {
15        printf("Maaf, Anda Belum Berhasil\n");
16    }
17
18    system("pause");
19    return 0;
20 }
```

- nilai.py



```
1 Nilai = int(input("Masukkan Nilai Anda : "))
2 if Nilai > 60:
3     print("Selamat, Anda Lulus")
4 else:
5     print("Maaf, Anda Belum Berhasil")
```

3. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera di atas, kemudian simpanlah dalam folder yang diberi nama sesuai dengan NIM anda
4. Kompilasi kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
5. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
6. Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error

2.4.3. Nested if (if bersarang)

1. Dengan kondisi program kompiler bahasa C dan Python yang masih terbuka tersebut.
2. Ketiklah kode sumber berikut ini dengan benar dan tepat

- hariif.c

```
● ● ●
1 #include <stdio.h>
2
3 int main() {
4     int code;
5
6     printf("Masukan kode hari ini [1..7]: ");
7     scanf("%d", &code);
8
9     if (code == 1) {
10         printf("Ini adalah kode hari untuk hari SENIN\n");
11     } else if (code == 2) {
12         printf("ini adalah kode hari SELASA\n");
13     } else if (code == 3) {
14         printf("Ini adalah kode hari RABU\n");
15     } else if (code == 4) {
16         printf("Ini adalah kode hari KAMIS\n");
17     } else if (code == 5) {
18         printf("Ini adalah kode hari ini JUMAT\n");
19     } else if (code == 6) {
20         printf("Ini adalah kode hari SABTU\n");
21     } else if (code == 7) {
22         printf("Ini adalah kode hari ini MINGGU\n");
23     } else {
24         printf("Kode yang Anda masukkan tidak dikenali\n");
25     }
26
27     return 0;
28 }
```

- hariif.py

```
● ● ●
1 kode = int(input("Masukkan kode hari [1..7] : "))
2
3 if kode == 1:
4     print("Ini kode hari SENIN")
5 elif kode == 2:
6     print("Ini kode hari SELASA")
7 elif kode == 3:
8     print("Ini kode hari RABU")
9 elif kode == 4:
10    print("Ini kode hari KAMIS")
11 elif kode == 5:
12    print("Ini kode hari JUMAT")
13 elif kode == 6:
14    print("Ini kode hari SABTU")
15 elif kode == 7:
16    print("Ini kode hari MINGGU")
17 else:
18    print("Kode yang anda masukkan tidak dikenal")
```

3. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera di atas, kemudian simpanlah dalam folder yang diberi nama sesuai dengan NIM anda
4. Kompilasi kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
5. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
6. Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error

2.4.4. Switch Sederhana

1. Dengan kondisi program kompiler bahasa C dan Python yang masih terbuka tersebut.
2. Ketiklah kode sumber berikut ini dengan benar dan tepat

- kalseder.c

```
1 #include <stdio.h>
2 #include <conio.h>
3 #include <stdlib.h>
4
5 int main() {
6     int kode;
7
8     printf("Masukkan kode hari [1..7] : ");
9     scanf(" %d", &kode);
10
11    switch (kode) {
12        case 1:
13            printf("\nIni kode hari SENIN");
14            break;
15        case 2:
16            printf("\nIni kode hari SELASA");
17            break;
18        case 3:
19            printf("\nIni kode hari RABU");
20            break;
21        case 4:
22            printf("\nIni kode hari KAMIS");
23            break;
24        case 5:
25            printf("\nIni kode hari JUMAT");
26            break;
27        case 6:
28            printf("\nIni kode hari SABTU");
29            break;
30        case 7:
31            printf("\nIni kode hari MINGGU");
32            break;
33        default:
34            printf("\nKode yang anda masukkan tidak dikenal");
35    }
36    system("pause");
37    return 0;
38 }
```

- kalseder.py

```
1 def Hari(i):
2     switcher = {
3         1: 'Ini kode hari SENIN',
4         2: 'Ini kode hari SELASA',
5         3: 'Ini kode hari RABU',
6         4: 'Ini kode hari KAMIS',
7         5: 'Ini kode hari JUMAT',
8         6: 'Ini kode hari SABTU',
9         7: 'Ini kode hari MINGGU'
10    }
11    return switcher.get(i, "Kode yang anda masukkan tidak dikenal")
12
13 print(Hari(7))
```

3. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera diatas, kemudian simpanlah dalam folder yang diberinama sesuai dengan NIM anda
4. Kompilasilah kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
5. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
6. Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error

2.4.5. “Switch” – contoh kasus: kalkulator sederhana

1. Dengan kondisi program kompiler bahasa C dan Python yang masih terbuka tersebut.
2. Ketiklah kode sumber berikut ini dengan benar dan tepat
 - kalseder_bonus.c



```

#include <stdio.h>
#include <stdlib.h>

int main() {
    char operator;
    float operand1, operand2;

    printf("Program kalkulator sederhana.....\n");
    printf("* => perkalian\n");
    printf("/ => pembagian\n");
    printf("+ => penjumlahan\n");
    printf("- => pengurangan\n");
    printf("Contoh Menghitung => 5/2 Lalu tekan ENTER\n\n");

    printf("Silahkan Masukkan Operasi : ");
    if (scanf("%c %f", &operator, &operand1) != 3) {
        printf("\nInput tidak valid.\n");
        return 1;
    }

    switch (operator) {
        case '*':
            printf("Hasil = %.2f\n", operand1 * operand2);
            break;
        case '/':
            if (operand2 == 0) {
                printf("\nError: Pembagian dengan nol tidak diizinkan.\n");
                return 1;
            }
            printf("Hasil = %.2f\n", operand1 / operand2);
            break;
        case '+':
            printf("Hasil = %.2f\n", operand1 + operand2);
            break;
        case '-':
            printf("Hasil = %.2f\n", operand1 - operand2);
            break;
        default:
            printf("\nOperator tidak valid.\n");
            return 1;
    }

    return 0;
}

```

- kalseder_bonus.py



```
1 print("Program kalkulator sederhana")
2 print("* => perkalian")
3 print("/ => pembagian")
4 print("+ => penjumlahan")
5 print("- => pengurangan")
6 print("Contoh Menghitung => 5 / 2 Lalu tekan ENTER")
7
8 operand1, operatore, operand2 = input("Silahkan Menghitung Angka : ").split()
9
10 def kalkulator(operatore):
11     switcher = {
12         '*': lambda: float(operand1) * float(operand2),
13         '/': lambda: float(operand1) / float(operand2),
14         '+': lambda: int(operand1) + int(operand2),
15         '-': lambda: int(operand1) - int(operand2),
16     }
17     return switcher.get(operatore, "Perintah Anda Salah. Silahkan Ulangi")()
18
19 print(kalkulator(operatore))
```

3. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera diatas, kemudian simpanlah dalam folder yang diberinama sesuai dengan NIM anda
4. Kompilasilah kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
5. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
6. Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error

2.5. Pertanyaan

1. Bagaimana cara untuk malakukan pengambilan keputusan yang menangani beberapa akemungkinan kondisi dalam suatu program?
2. Pada bahasa pemrograman C apakah pernyataan switch dapat digunakan untuk mengevaluasi tipe data selain bilangan bulat atau karakter? Jelaskan!
3. Buatlah program dalam bahasa C dan Python untuk menentukan angka terbesardan terkecil dari tiga buah bilangan yang diinput dari keyboard ?
4. Buatlah program dalam bahasa C dan Python dengan ketentuan sebagai berikut
:
INPUT
Nim, nama mahasiswa, nilai uts, nilai uas, nilai tugas
PROSES

1.	$\text{Nilai Akhir} = 30\% \times \text{nilai uts} + 50\% \times \text{nilai tugas} + 20\% \times \text{nilai tugas}$		
2.	Nilai Akhir	Nilai Huruf	Nilai Akhir
	90 – 100	A	50 – 59
	70 – 89	B	0 – 49
	60 – 69	C	D
			E

OUTPUT

LAPORAN NILAI MAHASISWA

Nim :

Nama :

Nilai Akhir :

Nilai Huruf :

5. Buatlah sebuah program yang akan menerima sebuah bilangan bulat X. Jika X merupakan satuan, maka akan mencetak “Satuan”. Jika X merupakan puluhan, maka akan mencetak “Puluhan”. Jika X merupakan ratusan, maka akan mencetak “Ratusan”. Jika X merupakan ribuan, maka akan mencetak “Ribuan”. Jika X merupakan puluh ribuan, maka akan mencetak “Puluh Ribuan”!

UNIT III: PERULANGAN

3.1. Tujuan Praktikum

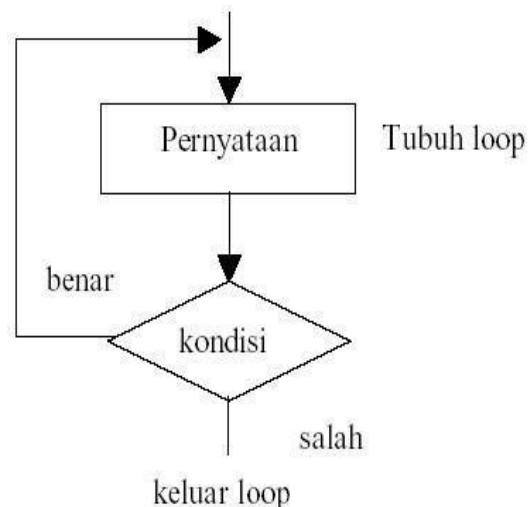
1. Mahasiswa mengenal perulangan dan perulangan bersarang
2. Mahasiswa mengkonstruksi algoritma penrulangan (for, while do, do while)
3. Mahasiswa menyusun program yang memiliki struktur perulangan bersarang
4. Mahasiswa mampu menggunakan proses perulangan menggunakan pada C dan Python 3
5. Mahasiswa mampu menggunakan proses dan variasi perulangan menggunakan goto, continue, dan break pada C
6. Mahasiswa mampu menggunakan proses dan variasi perulangan menggunakan continue dan break pada Python 3

3.2. Dasar Teori

Pada Bahasa C terdapat 3 buah struktur *loop*. Pernyataan yang dipakai berupa **while**, **do-while** dan **for**. Sedangkan untuk Python 3, perulangan terbagi dalam 2 jenis yakni perulang iterasi dan perulangan conditional, yakni **for** dan **while**. Berhubungan dengan penanganan loop ada beberapa pernyataan lain yang berfungsi untuk keluar dari loop yaitu **break** dan **continue**, namun untuk C ada intruksi khusus yakni **goto**.

3.2.1. Pernyataan do-while

Pernyataan ini memiliki bentuk umum seperti tampak pada gambar 3-0 dibawah ini sebagai berikut :

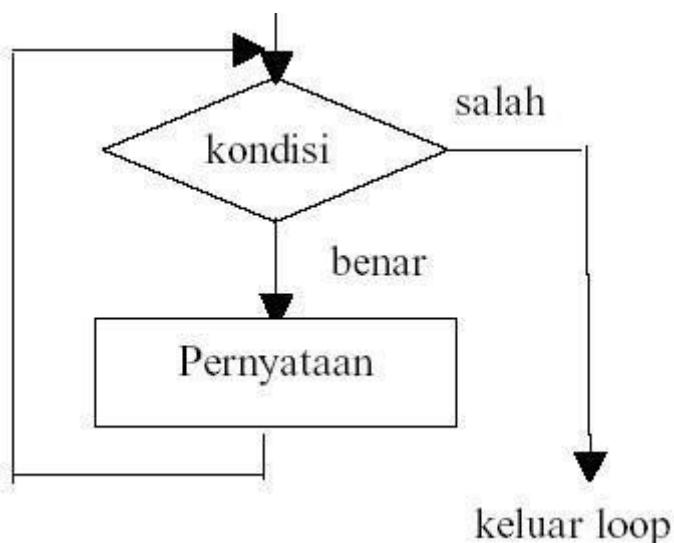


gambar 3-0 Bentuk umum do while

Tahapan proses :	mula-mula pernyataan dijalankan dulu, kemudian kondisi diuji, jika kondisi bernilai benar, maka pernyataan dijalankan lagi dan jika kondisi bernilai salah maka pernyataan tidak dijalankan lagi (berhenti).
Sintaks :	C: do pernyataan while (kondisi) Python: while (kondisi): pernyataan

3.2.2. Pernyataan While

Pernyataan ini memiliki bentuk umum seperti tampak pada gambar 3-1 dibawah ini sebagai berikut :



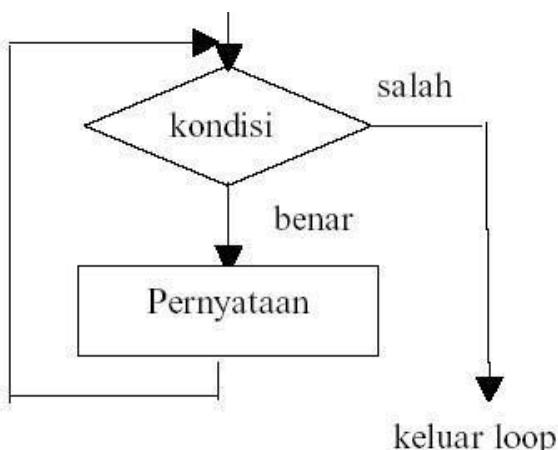
gambar 3-1 Bentuk umum do while

Tahapan proses :	Dari awal langsung dilakukan tes kondisi jika kondisi bernilai benar maka menjalankan pernyataan dan jika kondisi bernilai salah maka keluar dari loop. Jadi pada perulangan menggunakan while memungkinkan sebuah pernyataan tidak dijalankan sama sekali.
Sintaks :	C: while (kondisi) pernyataan Python: while (kondisi): pernyataan

Yang membedakan antara perulangan menggunakan **do-while** dan **while**, biasanya adalah pada pernyataan while harus diberikan terlebih dahulu inisialisasi terhadap variabel yang akan dijadikan sebagai kondisi dari **while**. Dalam python, hanya ada perulangan **while**, sehingga untuk logika **do-while** dilihat dari alur program.

3.2.3. Pernyataan For

Pernyataan ini memiliki bentuk umum seperti tampak pada gambar 3-2 dibawah ini sebagai berikut :



gambar 3-2 Bentuk umum do while

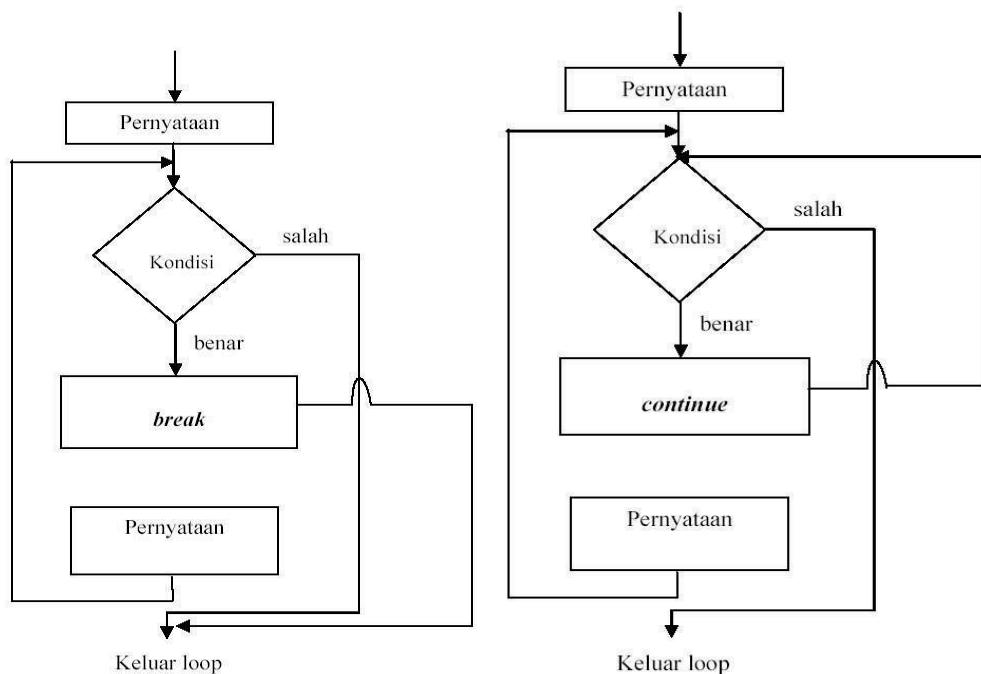
Tahapan proses :	Pernyataan akan dijalankan mulai dari nilai awal (<i>ungkapan1</i>) dan keluar dari loop jika sudah sesuai dengan nilai akhir (<i>ungkapan2</i>).
Sintaks :	<p>Bahasa C :</p> <pre>for (ungkapan1; ungkapan2; ungkapan3) pernyataan</pre> <ul style="list-style-type: none"> - Ungkapan1 → digunakan untuk memberikan inisialisasi terhadap variabel pengendali loop. - Ungkapan2 → digunakan sebagai kondisi untuk keluar dari loop. - Ungkapan3 → digunakan sebagai pengatur kenaikan nilai variabel pengendali loop. <p>Bahasa Python 3 :</p> <pre>for (var) in (iterasi) for (var) in range (inisiasi, stop, increment)</pre> <ul style="list-style-type: none"> - var → variabel untuk perulangan - iterasi → iterasi untuk perulangan - insiasi → nilai untuk memulai perulangan

- stop → kondisi batas akhir perulangan
- increment → digunakan untuk peningkatan variabel yang diulang

3.2.4. Pernyataan Break dan Goto

Untuk keluar dari sebuah loop atau perulangan dapat dilakukan ketika sarat kondisi perulangan telah terpenuhi tetapi kita juga dapat melakukan proses keluar paksa tersebut dengan menggunakan perintah break, setelah menggunakan break kita dapat memanfaatkan perintah continue untuk menjalankan proses lainnya yaitu iterasi pada loop yang sama, sedangkan goto (khusus pada bahasa C) untuk menjalankan perintah yang diawali dengan sebuah label identifier yang dimulai atau diawali dengan tanda (:= titik dua).

Perhatikan flowchartnya berikut ini :



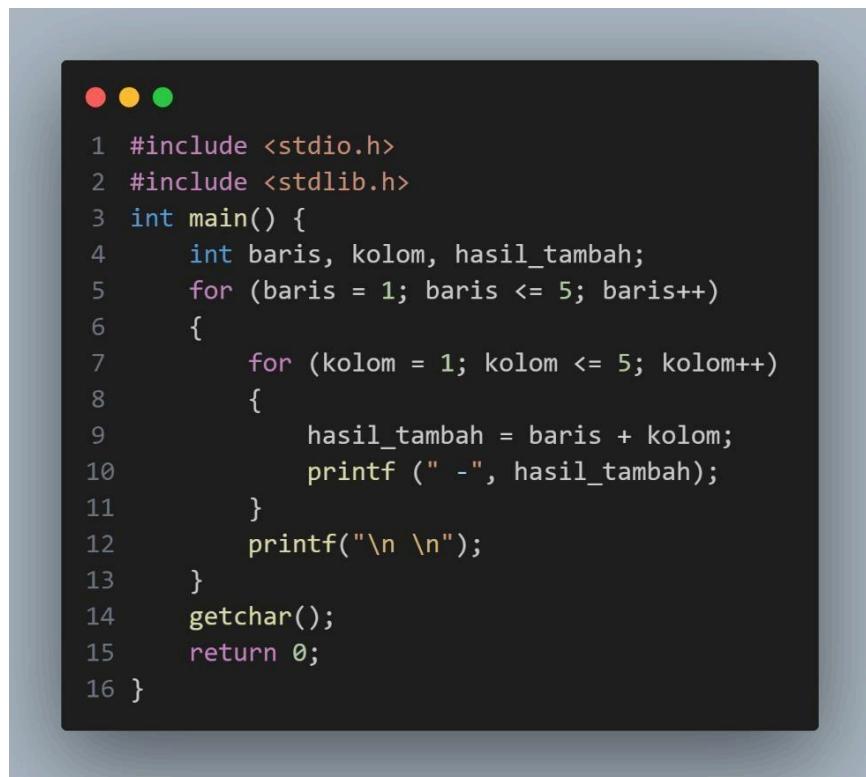
3.3. Alat dan Bahan

1. Seperangkat Komputer
2. Software Compiler C (ex : Borland C/C++ atau Dev C/C++)
3. Software Compiler Python (ex: VSCode)

3.4. Percobaan dan Pengamatan

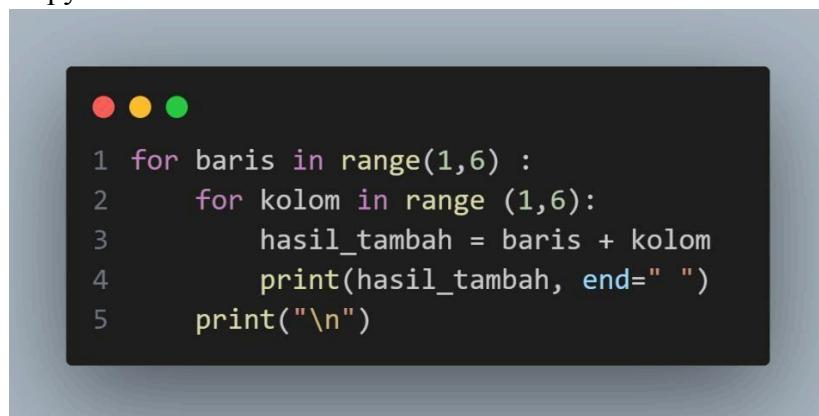
3.4.1. Penjumlahan Beberapa bilangan

1. Nyalakan komputer yang ada dihadapan anda, biarkan sampai masuk kedalam sistem operasi windows
2. Bukalah program kompiler bahasa C dan Python 3 yang ada dalam komputer tersebut.
3. Ketiklah kode sumber berikut ini dengan benar dan tepat
 - bil.c



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4     int baris, kolom, hasil_tambah;
5     for (baris = 1; baris <= 5; baris++) {
6         for (kolom = 1; kolom <= 5; kolom++)
7         {
8             hasil_tambah = baris + kolom;
9             printf (" -", hasil_tambah);
10            printf("\n \n");
11        }
12    }
13    getchar();
14    return 0;
15 }
```

- bil.py



```
1 for baris in range(1,6) :
2     for kolom in range (1,6):
3         hasil_tambah = baris + kolom
4         print(hasil_tambah, end=" ")
5     print("\n")
```

4. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera di atas, kemudian simpanlah dalam folder yang diberi nama sesuai dengan NIM anda
5. Kompilasilah kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
6. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
7. Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error

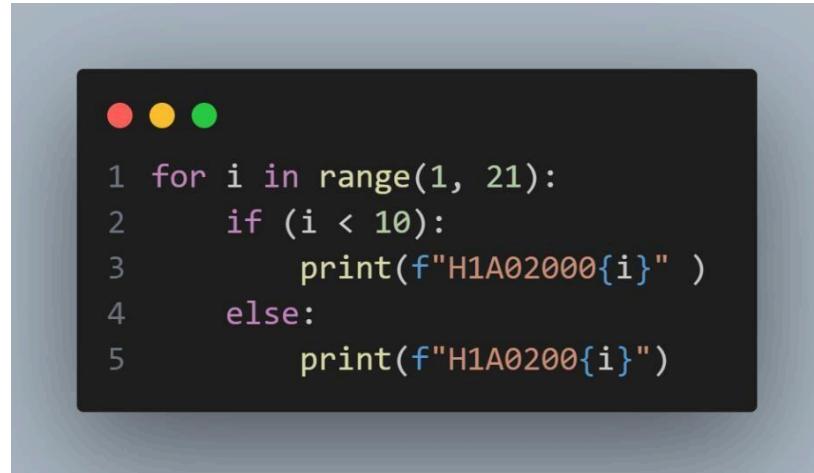
3.4.2. For

1. Dengan kompiler bahasa C dan Python 3 yang ada dalam komputer tersebut maka
2. Ketiklah kode sumber berikut ini dengan benar dan tepat

- for.c

```
1 #include <stdio.h>
2
3 int main (){
4     int i;
5     for(i = 1; i<= 20; i++){
6         if (i < 10){
7             printf("H1A02000");
8             printf("%d", i);
9             printf("\n");
10        } else {
11            printf("H1A0200");
12            printf("%d", i);
13            printf("\n");
14        }
15    }
16    return 0;
17 }
```

- for.py



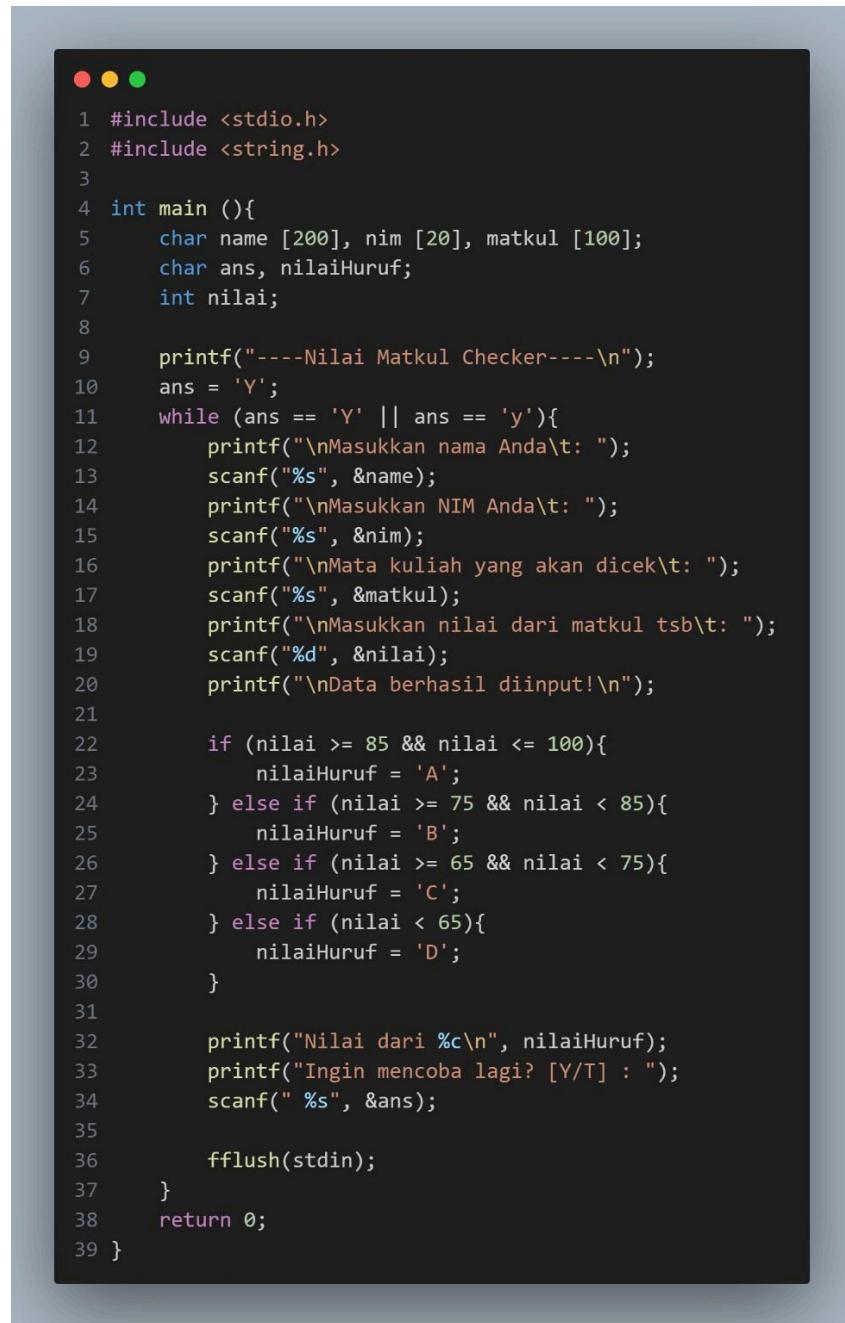
```
● ● ●
1 for i in range(1, 21):
2     if (i < 10):
3         print(f"H1A0200{i}" )
4     else:
5         print(f"H1A0200{i}")
```

3. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera diatas, kemudian simpanlah dalam folder yang diberinama sesuai dengan NIM anda
4. Kompilasilah kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
5. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
6. Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error

3.4.3. While do

- 1. Dengan kompiler bahasa C yang ada dalam komputer tersebut maka
- 2. Ketiklah kode sumber berikut ini dengan benar dan tepat

- nilaiMatkulChecker.c



```
● nilaiMatkulChecker.c
1 #include <stdio.h>
2 #include <string.h>
3
4 int main (){
5     char name [200], nim [20], matkul [100];
6     char ans, nilaiHuruf;
7     int nilai;
8
9     printf("----Nilai Matkul Checker----\n");
10    ans = 'Y';
11    while (ans == 'Y' || ans == 'y'){
12        printf("\nMasukkan nama Anda\t: ");
13        scanf("%s", &name);
14        printf("\nMasukkan NIM Anda\t: ");
15        scanf("%s", &nim);
16        printf("\nMata kuliah yang akan dicek\t: ");
17        scanf("%s", &matkul);
18        printf("\nMasukkan nilai dari matkul tsb\t: ");
19        scanf("%d", &nilai);
20        printf("\nData berhasil diinput!\n");
21
22        if (nilai >= 85 && nilai <= 100){
23            nilaiHuruf = 'A';
24        } else if (nilai >= 75 && nilai < 85){
25            nilaiHuruf = 'B';
26        } else if (nilai >= 65 && nilai < 75){
27            nilaiHuruf = 'C';
28        } else if (nilai < 65){
29            nilaiHuruf = 'D';
30        }
31
32        printf("Nilai dari %c\n", nilaiHuruf);
33        printf("Ingin mencoba lagi? [Y/T] : ");
34        scanf(" %s", &ans);
35
36        fflush(stdin);
37    }
38    return 0;
39 }
```

- nilaiMatkulChecker.py

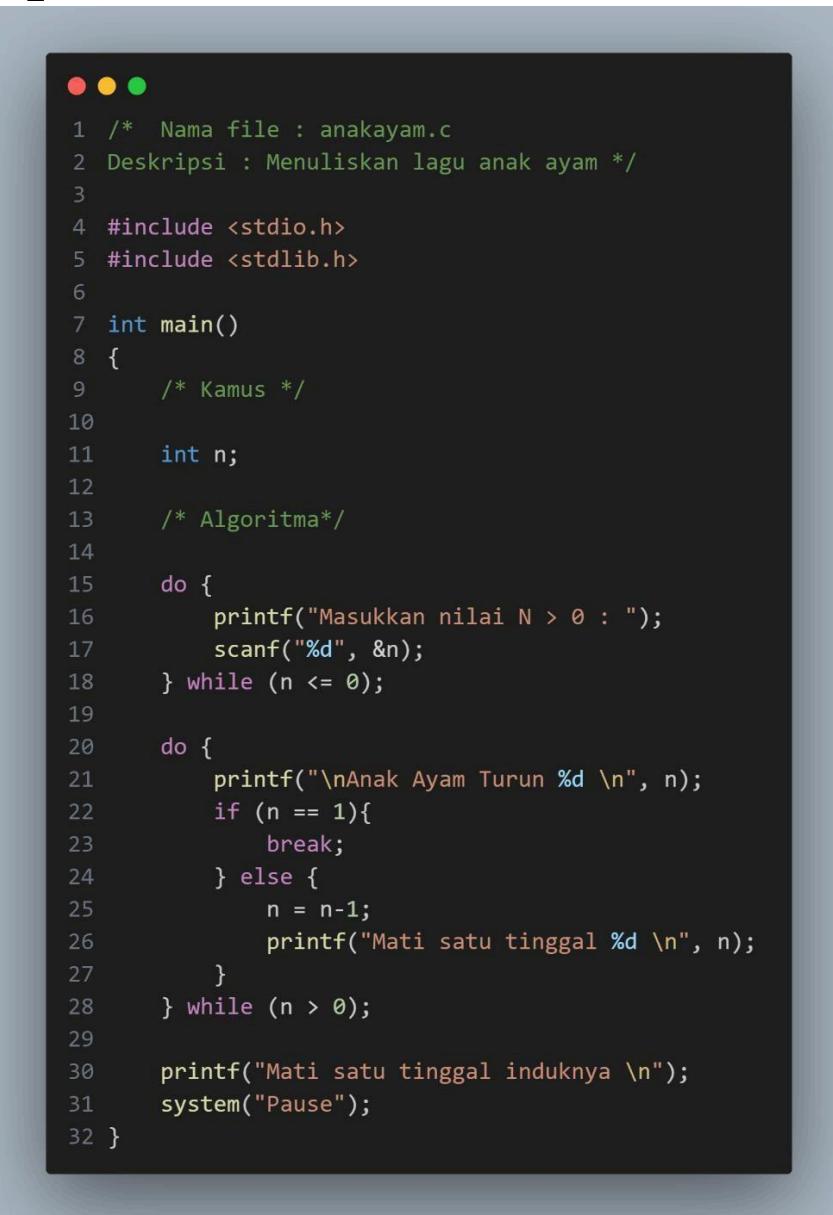
```
1 import time
2
3 print('----Nilai Matkul Checker')
4 ans = 'Y'
5 while ans == 'Y' or ans == 'y':
6     nama = input('Masukkan nama Anda\t: ')
7     nim = input('Masukkan NIM Anda\t: ')
8     matkul = input('Mata kuliah yang akan dicek\t: ')
9     nilai = int(input('Masukkan nilai dari matkul tsb\t: '))
10    print('Data berhasil diinput!')
11
12    if 85 <= nilai <= 100:
13        nilaiHuruf = 'A'
14    elif 75 <= nilai < 85:
15        nilaiHuruf = 'B'
16    elif 65 <= nilai < 75:
17        nilaiHuruf = 'C'
18    elif nilai < 65:
19        nilaiHuruf = 'D'
20
21    time.sleep(1)
22    print('Sedang diproses')
23    time.sleep(0.5)
24    dots = '.....'
25    for dot in dots:
26        print (dot)
27        time.sleep(0.5)
28    print('Nilai dari matkul Anda masuk ke kategori ' + nilaiHuruf + '\n')
29    time.sleep(1)
30    ans = input('Ingin mencoba lagi? [Y/T] : ')
31
32 print('\nProgram berhenti')
```

- 3 Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera diatas, kemudian simpanlah dalam folder yang diberi nama sesuai dengan NIM anda
- 4 Kompilasilah kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
- 5 Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
- 6 Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error

3.4.4. Do While – lagu anak ayam

- 1. Dengan kompiler bahasa C yang ada dalam komputer tersebut maka
- 2. Ketiklah kode sumber berikut ini dengan benar dan tepat

- do_while.c



```
1 /* Nama file : anakayam.c
2 Deskripsi : Menuliskan lagu anak ayam */
3
4 #include <stdio.h>
5 #include <stdlib.h>
6
7 int main()
8 {
9     /* Kamus */
10
11     int n;
12
13     /* Algoritma*/
14
15     do {
16         printf("Masukkan nilai N > 0 : ");
17         scanf("%d", &n);
18     } while (n <= 0);
19
20     do {
21         printf("\nAnak Ayam Turun %d \n", n);
22         if (n == 1){
23             break;
24         } else {
25             n = n-1;
26             printf("Mati satu tinggal %d \n", n);
27         }
28     } while (n > 0);
29
30     printf("Mati satu tinggal induknya \n");
31     system("Pause");
32 }
```

- do_while.py



```
1 n = 0
2
3 while n <= 0:
4     n = int(input('Masukkan nilai n > 0 : '))
5
6 while n > 0:
7     print('\nAnak Ayam Turun ' + str(n))
8
9     if n == 1:
10        break
11    else:
12        n = n - 1
13    print('Mati satu tinggal ' + str(n))
14
15 print('Mati satu tinggal induknya\n')
```

- 3. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera diatas, kemudian simpanlah dalam folder yang diberinama sesuai dengan NIM anda
- 4. Kompilasilah kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
- 5. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
- 6. Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error

3.5. Pertanyaan

1. Buatlah program perulangan yang menampilkan angka dari 1 sampai 100 kecuali bilangan prima! Sertakan penjelasannya!
2. Buatlah program dalam bahasa c dan python untuk menghitung jumlah angka dari suatu bilangan. Contohnya Jumlah angka dari bilangan 3255 adalah $3 + 2 + 5 + 5 = 15$
3. Jelaskan apa yang anda ketahui dengan perulangan while! Kasus seperti apa yang menuntuk kita untuk menggunakan perulangan while atau for?
4. Jelaskan perbedaan perulangan do-while dan perulangan while-do!

UNIT IV: ARRAY DAN STRING

4.1. Tujuan Praktikum

1. Mengenal Struktur Data Array
2. Mendefinisikan Array, Mengakses Array dalam Notasi Algoritmik dan Bahasa C/C++
3. Membuat program yang memiliki struktur data array dan operasi , struktur if-then-else, pengulangan, bersarang, dalam Notasi Algoritmik dan Bahasa C/C++ Mendefinisikan Array, Mengakses Array dalam Notasi Algoritmik dan Bahasa C/C++ serta Python
4. Pengenalan struktur data array, array list, vecor, list linear menggunakan python
5. Membuat program yang memiliki struktur data array dan operasi struktur if-then-else, pengulangan, dan pengulangan bersarang dalam Notasi Algoritmik, Bahasa C/C++ dan Python

4.2. Dasar Teori

Array adalah sekumpulan variabel yang memiliki tipe data yang sama dan dinyatakan dengan nama yang sama. Array merupakan konsep yang penting dalam pemrograman, karena array memungkinkan untuk menyimpan data maupun referensi objek dalam jumlah banyak dan terindeks. Array menggunakan indeks integer untuk menentukan urutan elemenelemennya, dimana elemen pertamanya dimulai dari indeks 0, elemen kedua memiliki indeks 1, dan seterusnya.

4.2.1. Array Satu Dimensi

Array Satu dimensi adalah kumpulan elemen identik yang tersusun dalam satu baris. Elemen- elemen tersebut memiliki tipe data yang sama, tetapi isi dari elemen tersebut boleh berbeda, seperti tampak pada gambar 1.1 dibawah ini :

Elemen ke-	0	1	2	3	4	5	6	7	8	9
Nilai	23	34	32	12	25	14	23	12	11	10

gambar 5-0 Contoh array satu dimensi

4.2.2. Array Dua Dimensi

Array dua dimensi sering digambarkan sebagai sebuah matriks, merupakan perluasan dari array satu dimensi. Jika array satu dimensi hanya terdiri dari sebuah baris dan beberapa kolom elemen, maka array dua dimensi terdiri dari beberapa baris dan beberapa kolom elemen yang bertipe sama sehingga dapat digambarkan pada gambar 1.1 berikut ini :

	0	1	2	3	4	5	6
0	10	21	23	43	45	78	65
1	45	43	65	12	21	12	21
2	32	34	23	56	54	34	45
3	11	12	32	23	56	76	45

gambar 5-1 Contoh array dua dimensi

```
Bentuk umum pada bahasa C:  
<tipe data> NamaArray [m][n];  
Atau  
<tipe data> NamaArray [m][n] = { {a,b,...z},{1,2,...,n-1} };  
Contoh:  
double matrix[4][4];  
bool papan[2][2] = { {true,false},{true,false} };  
  
Bentuk umum pada bahasa Python :  
from array import *  
  
T = [[11, 12, 5, 2], [15, 6, 10, 13], [10, 8, 12, 5], [12, 15,  
     8, 6]]  
for baris in T:  
    for kolom in baris:  
        print(kolom,end= " ")  
    print()
```

4.2.3. Mendeklarasikan variable array

A. Bahasa C

Array harus dideklarasikan seperti layaknya sebuah variabel. Apabila Anda mendeklarasikan array, Anda harus membuat sebuah list dari tipe data, yang diikuti oleh tanda kurung buka dan kurung tutup, yang diikuti oleh nama *identifier*. Mendeklarasikan variabel array dengan tipe data yang diinginkan dengan cara yang hampir sama dengan variabel biasa. Misalnya untuk mendeklarasikan variabel bertipe integer, dapat dilakukan dengan cara :

```
int[ ] bilangan; atau int bilangan[ ];
```

Jadi perbedaan utama pendeklarasian variabel array dengan variabel biasa adalah adanya tanda kurung [] di akhir tipe data atau di akhir nama variabel array. Pada tahap pendeklarasian variabel array ini belum ada alokasi memori untuk menyimpan data.

B. Python

Berbeda dengan bahasa C , pada bahasa Python kita tidak perlu mendeklarasikan tipe datanya. Array pada python dapat didekarasikan sebagai berikut :

```
nama_array = [jumlah_elelen]
```

Contoh pendeklarasian array pada python :

```
angka = [1, 2, 3]
mobil = ["Ford", "Volvo", "BMW"]
```

Kedua pendeklarasian array diatas dapat dijalankan tanpa error sekalipun tidak mendeklarasikan tipe datanya, hanya saja jika kita ingin mendeklarasikan array dengan tipe data string kita harus menambahkan tanda kutip (' ' / " ") di dalam array-nya.

4.2.4. Mendefinisikan Variabel Array

A. Bahasa C

Setelah mendeklarasikan array, kita perlu mendefinisikan array, dalam arti menentukan besar array yang diinginkan. Misalnya dengan cara :

```
Bilangan = new int[5];
```

Array memiliki ukuran yang tetap dalam arti tidak dapat membesar atau mengecil ukurannya setelah didefinisikan. Setelah didefinisikan, maka variabel dengan nama bilangan dapat menyimpan 5 nilai integer. Setelah pendefinisian array, maka memori akan dialokasikan untuk menyimpan data dari array. Besar memori yang dialokasikan tergantung dari tipe data variabel array dan jumlah elemen array yang didefinisikan.

Contoh lain misalkan kita memiliki sekumpulan data ujian seorang siswa, ujian pertama bernilai 90, kemudian 95,78,85. Sekarang kita ingin menyusunnya sebagai suatu data kumpulan ujian seorang siswa. Dalam array kita menyusunnya sebagai berikut :

```
ujian[0] = 90;
ujian[1] = 95;
ujian[2] = 78;
ujian[3] = 85;
```

Empat pernyataan diatas memberikan nilai kepada array ujian. Tetapi sebelum kita memberikan nilai kepada array, kita harus mendeklarasikannya terlebih dahulu :

```
int ujian[4];
```

Perhatikan bahwa nilai 4 yang berada didalam tanda kurung menunjukkan jumlah elemen larik, bukan menunjukkan elemen larik yang ke-4. Jadi elemen larik ujian dimulaidari angka 0 sampai 3. Pemrogram juga dapat menginisialisasi larik sekaligus mendeklarasikannya, sebagai contoh :

```
int ujian[4] = {90, 95, 78, 85};
```

Array multidimensi merupakan array yang terdiri dari array yang tidak terbatas hanya dua dimensi saja. Kita dapat menggunakan kode berikut untuk mendapatkan array tiga dimensi :

```
int[][][]array_dimensi = new int[5][10][5];
```

Dan pada array multidimensi, kita dapat menentukan ukuran array yang berbeda pada tiap array : `int[][][] mdimensi = new int[5][][];`

B. Python

Pada bahasa Python, array dibuat dengan menempatkan semua item (elemen) di dalam tanda kurung siku [] dan dipisahkan oleh koma. Array dapat memiliki sejumlah elemen dan mungkin memiliki tipe yang berbeda (integer, float, string, dll.).

```
#Array kosong
array_saya = []

#Array dengan item integer
array_saya = [1, 2, 3]

#Array dengan tipe data campuran
array_saya = [1, 'Halo', 3.4]
```

Array juga dapat memiliki array lain sebagai item. Bentuk ini disebut dengan array bersarang.

```
array_saya = ["tikus", [1, 3, 5], ['b']]
```

4.2.5. Pengaksesan Elemen Array

A. Bahasa C

Untuk mengakses sebuah elemen dalam array, atau mengakses sebagian dari array, Anda harus menggunakan sebuah nomor atau yang disebut sebagai index atau subscript. Pengaksesan elemen array pada python tidak memerlukan titik koma “;”

Sebuah nomor index atau subscript telah diberikan kepada tiap anggota array, sehingga program dan programmer dapat mengakses setiap value apabila dibutuhkan.

Index selalu dalam integer. Dimulai dari nol, kemudian akan terus bertambah sampai list value dari array tersebut berakhir. Perlu dicatat, bahwa elemen-elemen didalam array dimulai dari 0 sampai dengan (`ukuranArray-1`). Sebagai contoh, pada array yang kita deklarasikan tadi, kita mempunyai,

```

//memberikan nilai 5 kepada elemen pertama array
Bilangan [1][1] = 5;
//mencetak elemen array yang
terakhir printf("%d", bilangan[5]);

```

B. Python

Dalam bahasa Python terdapat berbagai cara di mana kita dapat mengakses elemen pada array.

a. Indeks

Kita dapat menggunakan operator indeks [] untuk mengakses item dalam array. Dalam Python, indeks dimulai dari 0. Jadi, array yang memiliki 5 elemen akan memiliki indeks dari 0 hingga 4. Mencoba mengakses indeks selain ini akan meningkatkan IndexError. Indeks harus berupa bilangan bulat. Kita tidak bisa menggunakan float atau tipe data lainnya, hal ini akan menghasilkan TypeError. array bersarang diakses menggunakan pengindeksan bersarang.

```

array_saya = ['p', 'r', 'o', 'b', 'e']

# Output: p
print(array_saya[0])

# Output: o
print(array_saya[2])

# Output: e
print(array_saya[4])

# array bersarang
n_array = ["Happy", [2, 0, 1, 5]]

# Indeks bersarang
print(n_array[0][1])
print(n_array[1][3])

# Error! indeks hanya dilakukan dengan integer
print(array_saya[4.0])

```

b. Pengindeksan negatif

Python memungkinkan pengindeksan negatif untuk urutan pada array. Indeks -1 mengacu pada item terakhir, -2 untuk item terakhir kedua dan seterusnya.

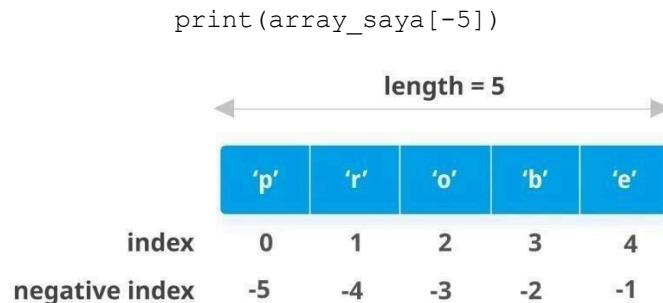
```

array_saya = ['p', 'r', 'o', 'b', 'e']

#output : e
print(array_saya[-1])

#output : p

```



Gambar 5-2 Pengindeksan array pada Python

4.2.6. Struktur

A. Bahasa C

Dalam C, tipe data struktur yang dideklarasikan dengan kata kunci struct, dapat mempunyai komponen dengan sembarang tipe data, baik tipe data dasar maupun tipe data turunan, termasuk fungsi. Misalnya, kita ingin membentuk tipe data struktur yang namanya kotak. Maka dapat dideklarasikan sebagai berikut :

```
struct tkotak
{
    double panjang;
    double lebar;
};
tkotak kotak;
```

Untuk memberi nilai ukuran kotak tersebut, kita dapat menggunakan perintah-perintah ini:

```
kotak.panjang = 10;
kotak.lebar = 7;
```

Untuk memberi nilai panjang dan lebar kotak, salah satu caranya adalah seperti di atas. Cara lain untuk memberi nilai panjang dan lebar adalah dengan membentuk suatu fungsi. Karena fungsi ini hanya digunakan untuk memberi nilai data panjang dan lebar suatu kotak, tentunya fungsi ini khusus milik objek kotak, sehingga harus dianggap sebagai anggota struktur kotak.

C sebagai bahasa pemrograman dapat mendefinisikan anggota tipe struktur yang berupa fungsi. Dengan menambah fungsi tersebut, maka struktur kotak menjadi lebih jelas bentuknya.

```

struct tkotak
{
    double panjang;
    double lebar;
    void SetUkuran(double pj, double lb)
    {
        panjang = pj;
        lebar = lb;
    };
};
tkotak kotak;

```

Dengan tipe struktur kotak seperti itu, untuk memberi nilai panjang dan lebar hanya dengan memanggil fungsi SetUkuran()

```
kotak.SetUkuran(10,7);
```

B. Python

Berbeda dengan bahasa C, struct pada Python merupakan modul yang digunakan untuk mengubah tipe data asli Python menjadi string byte dan sebaliknya. Modul struct terkait dengan bahasa C. Dimana kita harus mengetahui notasi yang digunakan dalam bahasa C untuk mewakili berbagai jenis data untuk bekerja dengan modul struct, diantaranya:

Data Type	Format Character
int	i
char	c
string	s
float	f

Untuk mengubah suatu tipe data asli pada Python ke dalam byte, kita dapat menggunakan fungsi berikut :

```
struct.pack(format, data1, data2, ...)
```

Dimana format merupakan format character dari tipe data yang kita miliki dan data1, data2, dst merupakan nilai dari data yang kita miliki. Untuk dapat menggunakan fungsi pada struct kita perlu meng-import terlebih dahulu modul struct tersebut seperti berikut:

```

import struct

# Format: i adalah integer pada bahasa C
# Format: f adalah float pada bahasa C
# Format 'if' merepresentasikan 'integer float'

var = struct.pack('if',1, 3.5) print(var)

output:
b'\x01\x00\x00\x00\x00\x00'

```

Sebaliknya, jika kita ingin melakukan konversi byte ke dalam tipe data asli, maka kita dapat menggunakan fungsi berikut:

```
struct.unpack(format, buffer)
```

Dimana format merupakan format character dari tipe data yang kita miliki dan buffer mengacu pada variable yang dituju, contoh:

```

import struct

var = struct.pack('if',1, 3.5) print(var)
tup = struct.unpack('if', var) print(tup)

output: b'\x01\x00\x00\x00\x00\x00'
(1, 3.5)

```

4.3. Alat dan Bahan

1. Seperangkat Komputer
2. Software Compiler C (ex : Borland c/c++ atau dev c/c++)
3. Software Interpreter Python (ex : IDLE atau VS Code)

4.4. Percobaan dan Pengamatan

4.4.1. Percobaan 1 ‘Array 1 Dimensi’

1. Nyalakan komputer yang ada dihadapan anda, biarkan sampai masuk kedalam sistem operasi windows
2. Bukalah program kompiler bahasa C dan intrepeter Python yang ada dalam komputer tersebut.
3. Ketiklah kode sumber berikut ini dengan benar dan tepat

- rata-rata.c



```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int i;
    int numbers[10];
    int count = 10;
    long sum = 0L;
    float average = 0.0f;
    printf("Masukan 10 bilangan: \n");
    for (i = 0; i < count; i++){
        printf("Data ke - %2d> ",i+1);
        scanf("%d", &numbers[i]);
        sum += numbers[i];
    }
    average = (float)sum/count;
    printf("\n Nilai rata-ratanya adalah: %f\n", average);

    system("pause");
    return 0;
}
```

- rata-rata.py

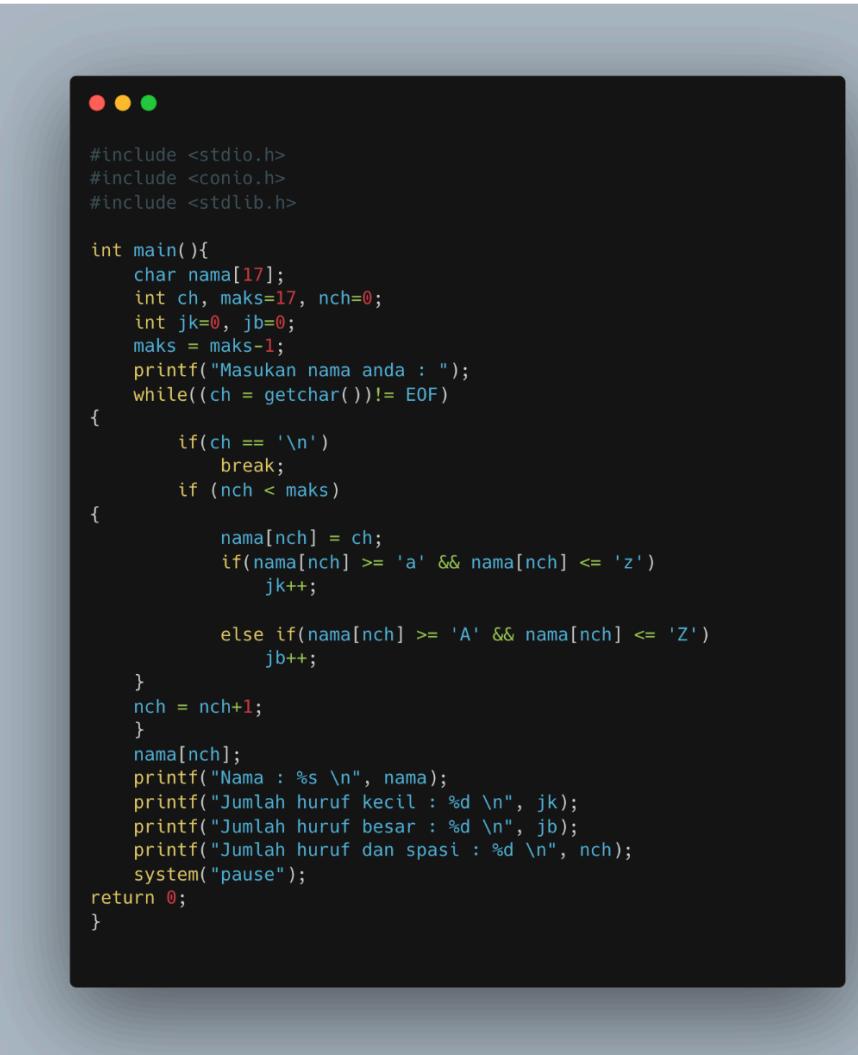


```
numbers = [None for i in range(10)]
count = 10
sum = 0
average = 0.0
print("\nMasukan 10 bilangan : ")

for i in range(0,10):
    numbers[i]=int(input("Data ke - " + str(i+1) + " > "))
    sum+=numbers[i]
average=float(sum)/count

print("\n Nilai rata-ratanya adalah: " ,average,"\\n")
```

4. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera diatas, kemudian simpanlah dalam folder yang diberinama sesuai dengan NIM anda
 5. Kompilasilah kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
 6. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
 7. Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error
 8. Masuk kembali pada editor kompiler C dan interpreter Python milik anda dan buatlah file baru kemudian ketikan kode sumber dibawah ini :
- array_string.c



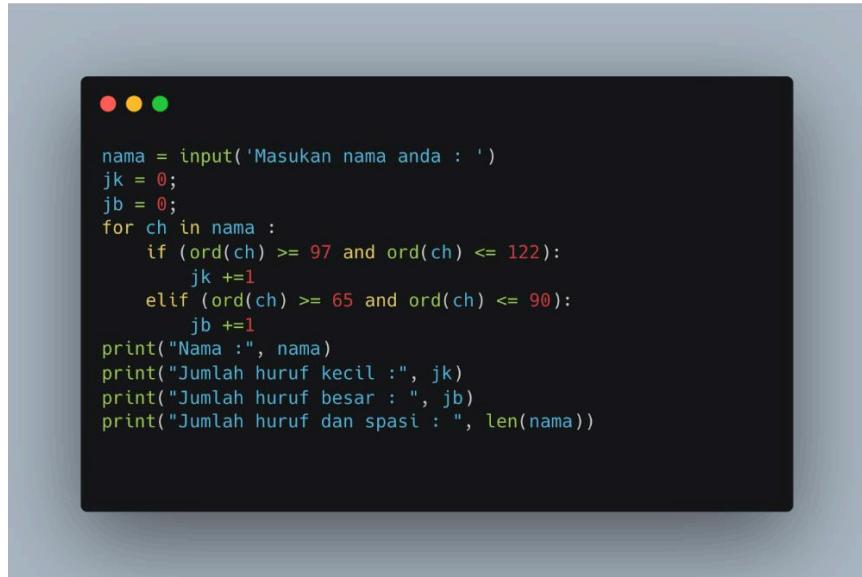
```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

int main(){
    char nama[17];
    int ch, maks=17, nch=0;
    int jk=0, jb=0;
    maks = maks-1;
    printf("Masukan nama anda : ");
    while((ch = getchar())!= EOF)
    {
        if(ch == '\n')
            break;
        if (nch < maks)
        {
            nama[nch] = ch;
            if(nama[nch] >= 'a' && nama[nch] <= 'z')
                jk++;
            else if(nama[nch] >= 'A' && nama[nch] <= 'Z')
                jb++;
        }
        nch = nch+1;
    }
    nama[nch];
    printf("Nama : %s \n", nama);
    printf("Jumlah huruf kecil : %d \n", jk);
    printf("Jumlah huruf besar : %d \n", jb);
    printf("Jumlah huruf dan spasi : %d \n", nch);
    system("pause");
    return 0;
}

```

- array_string.py



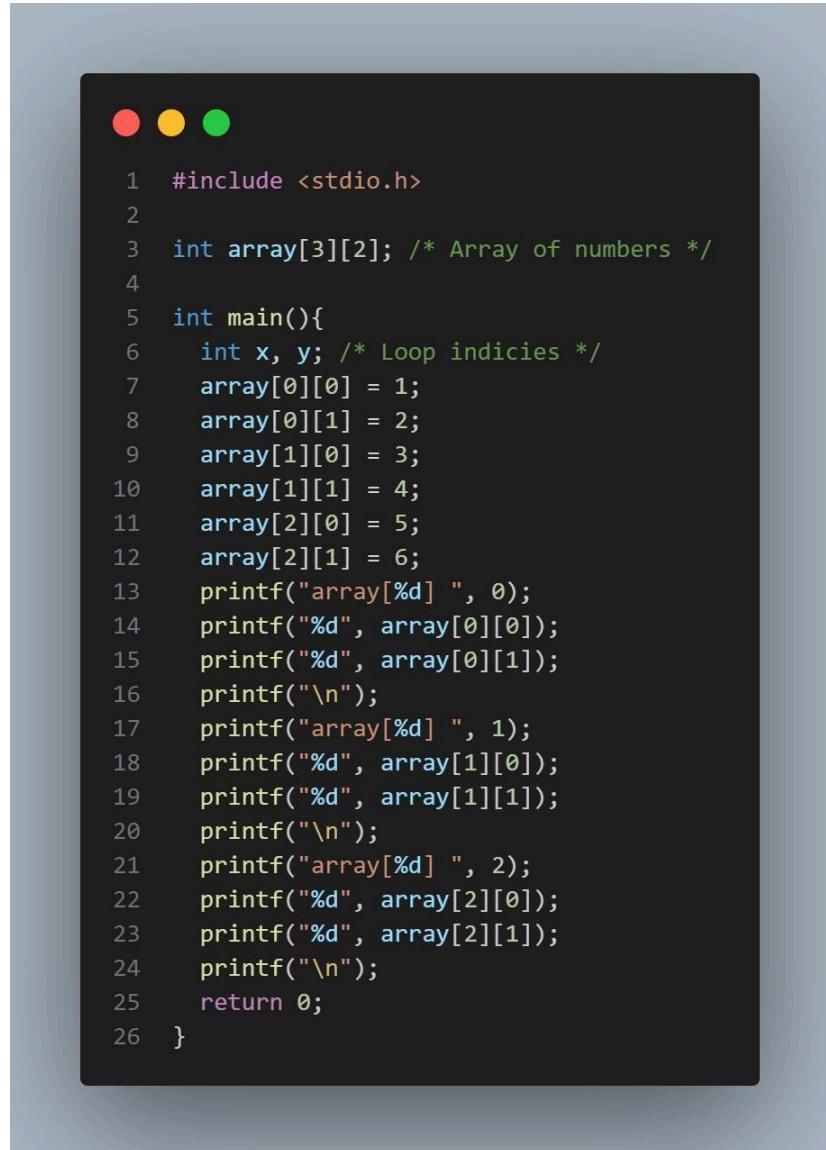
```
nama = input('Masukan nama anda : ')
jk = 0;
jb = 0;
for ch in nama :
    if (ord(ch) >= 97 and ord(ch) <= 122):
        jk +=1
    elif (ord(ch) >= 65 and ord(ch) <= 90):
        jb +=1
print("Nama :", nama)
print("Jumlah huruf kecil :", jk)
print("Jumlah huruf besar : ", jb)
print("Jumlah huruf dan spasi : ", len(nama))
```

9. Lakukan langkah 4 sampai dengan langkah 7 diatas

4.4.2. Percobaan 2 ‘Array 2 Dimensi’

1. Dengan kondisi editor kompiler C dan interpreter Python yang masih terbuka, buatlah file baru kembali dan ketikan kode sumber dibawah ini :

- array_2d.c



```
1 #include <stdio.h>
2
3 int array[3][2]; /* Array of numbers */
4
5 int main(){
6     int x, y; /* Loop indicies */
7     array[0][0] = 1;
8     array[0][1] = 2;
9     array[1][0] = 3;
10    array[1][1] = 4;
11    array[2][0] = 5;
12    array[2][1] = 6;
13    printf("array[%d] ", 0);
14    printf("%d", array[0][0]);
15    printf("%d", array[0][1]);
16    printf("\n");
17    printf("array[%d] ", 1);
18    printf("%d", array[1][0]);
19    printf("%d", array[1][1]);
20    printf("\n");
21    printf("array[%d] ", 2);
22    printf("%d", array[2][0]);
23    printf("%d", array[2][1]);
24    printf("\n");
25    return 0;
26 }
```

- array_2d.py



```
array = [[None]*2 for array in range (3)]
x,y = 0,0

array[0][0] = 1
array[0][1] = 2
array[1][0] = 3
array[1][1] = 4
array[2][0] = 5
array[2][1] = 6

print("array[{}].format(0), end=" ")
print("{}".format(array[0][0]), end =" ")
print("{}".format(array[0][1]))
print("array[{}].format(1),end =" ")
print("{}".format(array[1][0]), end =" ")
print("{}".format(array[1][1]))
print("array[{}].format(2),end =" ")
print("{}".format(array[2][0]), end =" ")
print("{}".format(array[2][1]))
```

2. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera diatas dalam folder yang diberi nama sesuai dengan NIM anda
3. Ulangi langkah 4 sampai dengan 7 (pada bagian pertama diatas)
4. Dengan kondisi editor kompiler c dan interpreter python yang masih terbuka, buatlah file baru kembali dan ketikan kode sumber dibawah ini :
5. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera diatas dalam folder yang diberi nama sesuai dengan NIM anda.

- matriks2d.c

```
● ● ●  
1 #include <stdio.h>  
2 #include <conio.h>  
3  
4 int main(){  
5     /*Kamus Lokal*/  
6     int baris, kolom, matriks[3][4];  
7     /*Algoritma*/  
8     printf("Program Output Matriks\n\n");  
9     printf("Input elemen Array : \n");  
10    for(baris = 0; baris < 3; baris++){  
11        for(kolom = 0; kolom < 4; kolom++){  
12            printf("matriks[%i][%i]>>> ", baris + 1, kolom + 1);  
13            scanf("%i", &matriks[baris][kolom]);  
14        }  
15        printf("\n");  
16    }  
17    printf("matriks :\n");  
18    for(baris = 0; baris < 3; baris++){  
19        for(kolom = 0; kolom < 4; kolom++){  
20            printf("%i", matriks[baris][kolom]);  
21        }  
22        printf("\n\n");  
23    }  
24    getch();  
25    return 0;  
26 }
```

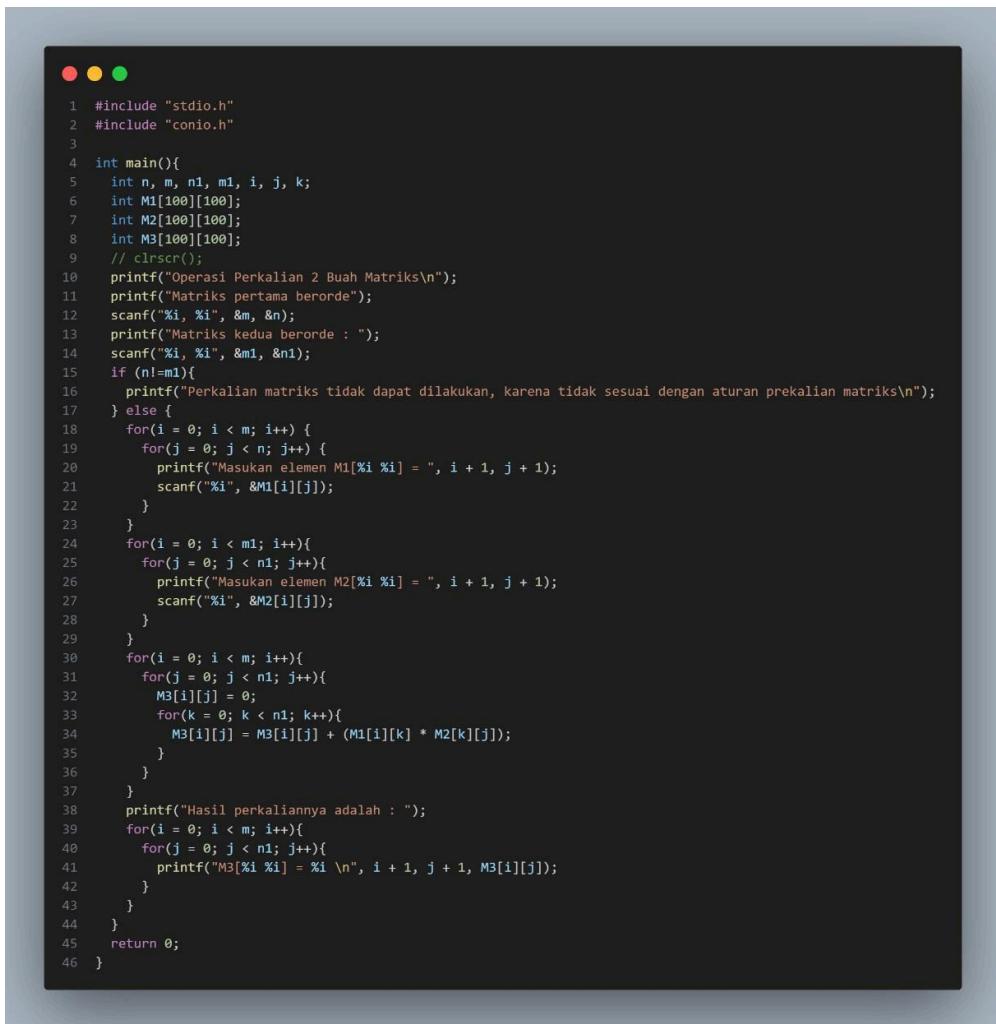
- matriks2d.py

```
● ● ●  
1 def main():  
2     matriks = [[None for i in range(4)] for j in range(3)]  
3     print("Input elemen array : \n")  
4     for baris in range(0, 3):  
5         for kolom in range(0, 4):  
6             print("matriks[{one}][{two}]".format(one = baris + 1, two = kolom + 1))  
7             matriks[baris][kolom] = int(input())  
8             print()  
9             print("Isi array : \n")  
10            for baris in range(3):  
11                for kolom in range(4):  
12                    print(matriks[baris][kolom], end="")  
13                    print()  
14 main()
```

6. Ulangi langkah 4 sampai dengan 7 (pada bagian pertama diatas)

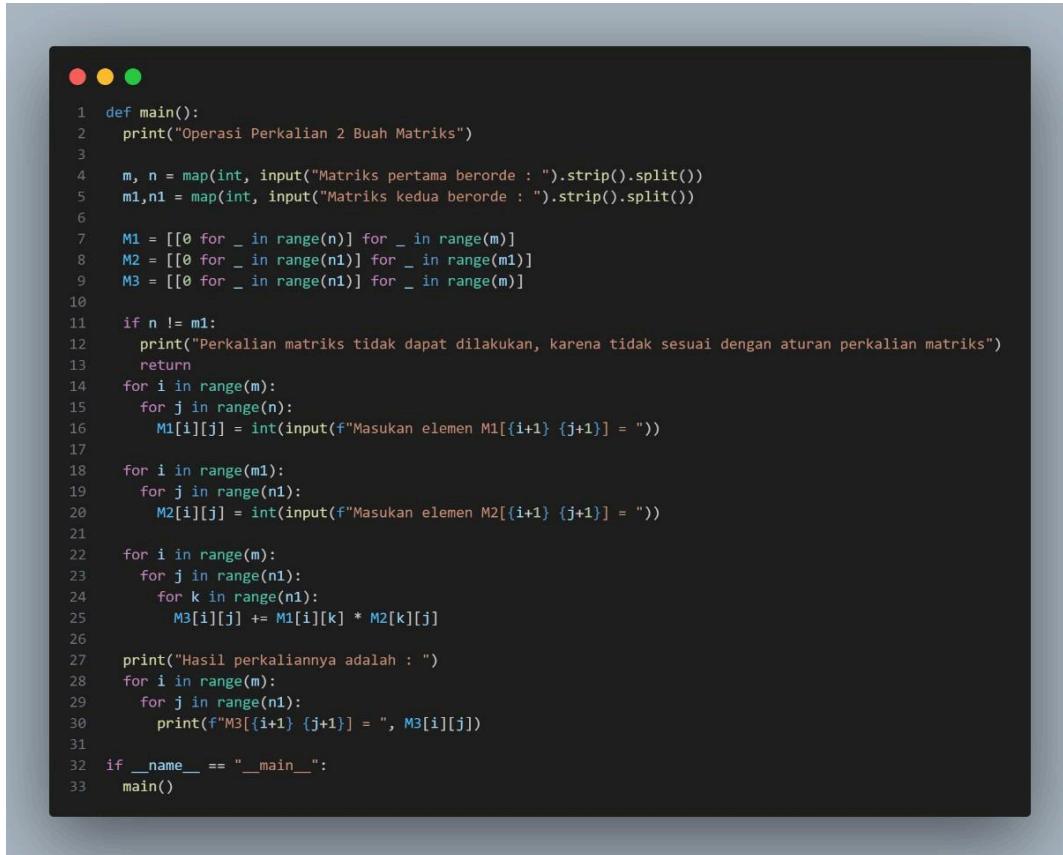
4.4.3. Percobaan 3 ‘Matriks’

1. Dengan kondisi editor kompiler c dan interpreter python yang masih terbuka, buatlah file baru kembali dan ketikan kode sumber dibawah ini :
 - matriks.c



```
1 #include "stdio.h"
2 #include "conio.h"
3
4 int main(){
5     int n, m, n1, m1, i, j, k;
6     int M1[100][100];
7     int M2[100][100];
8     int M3[100][100];
9     // clrscr();
10    printf("Operasi Perkalian 2 Buah Matriks\n");
11    printf("Matriks pertama berorde");
12    scanf("%i, %i", &m, &n);
13    printf("Matriks kedua berorde : ");
14    scanf("%i, %i", &m1, &n1);
15    if (n!=m1){
16        printf("Perkalian matriks tidak dapat dilakukan, karena tidak sesuai dengan aturan perkalian matriks\n");
17    } else {
18        for(i = 0; i < m; i++) {
19            for(j = 0; j < n; j++) {
20                printf("Masukan elemen M1[%i %i] = ", i + 1, j + 1);
21                scanf("%i", &M1[i][j]);
22            }
23        }
24        for(i = 0; i < m1; i++){
25            for(j = 0; j < n1; j++){
26                printf("Masukan elemen M2[%i %i] = ", i + 1, j + 1);
27                scanf("%i", &M2[i][j]);
28            }
29        }
30        for(i = 0; i < m; i++){
31            for(j = 0; j < n1; j++){
32                M3[i][j] = 0;
33                for(k = 0; k < m1; k++){
34                    M3[i][j] = M3[i][j] + (M1[i][k] * M2[k][j]);
35                }
36            }
37        }
38        printf("Hasil perkaliannya adalah : ");
39        for(i = 0; i < m; i++){
40            for(j = 0; j < n1; j++){
41                printf("M3[%i %i] = %i \n", i + 1, j + 1, M3[i][j]);
42            }
43        }
44    }
45    return 0;
46 }
```

- matriks.py



```
1 def main():
2     print("Operasi Perkalian 2 Buah Matriks")
3
4     m, n = map(int, input("Matriks pertama berorde : ").strip().split())
5     m1,n1 = map(int, input("Matriks kedua berorde : ").strip().split())
6
7     M1 = [[0 for _ in range(n)] for _ in range(m)]
8     M2 = [[0 for _ in range(n1)] for _ in range(m1)]
9     M3 = [[0 for _ in range(n1)] for _ in range(m)]
10
11    if n != m1:
12        print("Perkalian matriks tidak dapat dilakukan, karena tidak sesuai dengan aturan perkalian matriks")
13        return
14    for i in range(m):
15        for j in range(n):
16            M1[i][j] = int(input(f"Masukan elemen M1[{i+1} {j+1}] = "))
17
18    for i in range(m1):
19        for j in range(n1):
20            M2[i][j] = int(input(f"Masukan elemen M2[{i+1} {j+1}] = "))
21
22    for i in range(m):
23        for j in range(n1):
24            for k in range(n1):
25                M3[i][j] += M1[i][k] * M2[k][j]
26
27    print("Hasil perkaliannya adalah : ")
28    for i in range(m):
29        for j in range(n1):
30            print(f"M3[{i+1} {j+1}] = ", M3[i][j])
31
32 if __name__ == "__main__":
33     main()
```

2. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera diatas dalam folder yang diberi nama sesuai dengan NIM anda
3. Ulangi langkah 4 sampai dengan 7 (pada bagian pertama diatas)

4.5. Data Pengamatan

4.5.1. Percobaan 1 Array 1 Dimensi

1. Masukan Integer 1, 2, 3, 4, 5, 6, 7, 9, 10, 8 Hasilnya = _____
2. Masukan 10 Integer 2, 7, 31, 41, 6, 7, 9, 12, 5, 0 Hasilnya = _____
3. Ubahlan kode sumber Bahasa C menjadi int numbers[100] dan Python menjadi range(100), ulangi langkah 1 & 2 Hasilnya = _____
4. Masukan 100 Integer 1, 2, 2, 3, 4, 5, 6, ..., 100 Hasilnya = _____
5. Sesuai dengan kode sumber pada langkah 8 diatas, coba anda masukan karakter berikut ini, ABC123assdfasdf
6. Sesuai dengan kode sumber pada langkah 8 diatas, coba anda masukan karakter berikut ini, ABC!@#\$%^&*()_-

4.5.2. Percobaan 2 Array 2 Dimensi

1. Berdasarkan percobaan pertama pada program array_2d.c dan array_2d.py diatas, tampilkan hasil eksekusi programnya
2. Rubahlah kode sumbernya agar menampilkan elemen dari 0,0 sampai 3,2
3. Berdasarkan percobaan pada program matriks2d.c dan matriks2d.py diatas, tampilkan hasil eksekusi programnya
4. Jika inputannya dirubah menjadi karakter, a, b, c, d, e, f, g, h, i, j, k, l, maka hasil _____

4.5.3. Percobaan 3 Matriks

1. Berdasarkan percobaan pada matriks diatas, tampilkan hasil eksekusi programnya
2. Jika orde pertama matrik pertama sama dengan 2×2 dan orde matriks kedua 3×3 , maka hasil
3. Jika order pertama matriks pertama sama dengan 2×2 dan order matriks kedua 2×2 dengan inputan matriks pertama adalah a, b, c, d dan matriks kedua 1, 2, 3, 4 :

4.6. Pertanyaan

1. Buat sebuah algoritma program yang menyertakan proses pendefinisian, pendeklarasian dan pengaksesan variabel array. (Program yang dibuat harus berbeda dengan apa yang ada dalam modul praktikum ini)
2. Dari pertanyaan (1) di atas, buat programnya dalam bahasa C dan Python. Tuliskan kode sumber dan bila diperlukan sertakan *screenshot* tampilan program.
3. Buat sebuah program dimana menerima input sebuah list/array dan value tertentu. Ketika list/array tersebut memiliki value yang dituju maka program akan memberikan output True. Sedangkan akan memberikan output False ketika tidak ada value didalam list/array.

```
● ● ●  
you_only_need_one([1, 2, 3, 4, 5], 8) -> False  
you_only_need_one([2, 4, 6, 8], 8) -> True  
you_only_need_one(["hi", "aku", "ganteng"], "ganteng") -> True  
you_only_need_one([8, "hi", 9, ["aku", 1], "ganteng"], 1) -> False
```

4. Jelaskan perbedaan antara array dan list dalam pemrograman! dan kapan waktu yang tepat untuk menggunakan array ataupun list?
5. Buatlah sebuah program yang menukar posisi dua elemen dalam sebuah array bedasarkan indeks array yang diberikan! Contoh, array masukan adalah 1, 5, 6, 7 lalu indeks array yang ingin ditukar adalah indeks ke 1 dan ke 2 maka array terbaru adalah 1, 6, 5, 7

UNIT V FUNGSI DAN PROSEDUR

5.1. Tujuan Praktikum

1. Memformulasikan abstraksi masalah
2. Pengenalan dan menyusun Solusi Masalah dalam Abstract Data Type (ADT)
3. Membuat Project Prototipe Dasar: Konstruksi ADT, implementasi ADT dalam C/C++
4. Memahami pembentukan fungsi dan prosedur
5. Memahami proses pemanggilan fungsi dan prosedur

5.2. Dasar Teori

5.2.1. Tujuan fungsi

Fungsi adalah kumpulan baris perintah (blok program) yang dibuat untuk melaksanakan tugas tertentu, yang tujuannya adalah :

1. Program menjadi lebih terstruktur, sehingga mudah untuk dipahami, mencari kesalahan kode dan dikembangkan lebih lanjut.
2. Untuk mengurangi penulisan kebalik kode yang telah dibuat.
3. Fungsi memerlukan masukan yang dinamakan argumen atau parameter, yang akan diolah oleh fungsi.
4. Hasil akhir dari sebuah fungsi adalah sebuah nilai berupa nilai keluaran fungsi.

Perhatikan ilustrasinya pada gambar 4-0 berikut ini :

Tipe_fungsi nama_fungsi (parameter) Deklarasi parameter { bagian_fungsi }	Tipe_fungsi = tipe keluaran fungsi Nama_fungsi = nama fungsi Parameter = daftar parameter Deklarasi parameter = deklarasi variabel Bagian_fungsi = tubuh / isi fungsi
---	---

gambar 4-0 Ilustrasi fungsi

5.2.2. Fungsi Dasar

Bahasa C sendiri memiliki fungsi standard yaitu

1. printf() : menampilkan informasi data kelayar
2. scanf() : membaca kode tombol yang dimasukan

Dan bahasa python sendiri memiliki fungsi standard yaitu

3. print() : menampilkan informasi data kelayar
4. input() : membaca kode tombol yang dimasukan

Pada umumnya fungsi memerlukan nilai masukan atau parameter yang disebut sebagai argumen, nilai masukan ini akan diolah oleh fungsi, hasil akhir fungsi

berupa sebuah nilai yang kemudian disebut sebagai return value. Oleh karena itu fungsi sering disebut sebagai kotak hitam, seperti tampak pada gambar dibawah



ini

gambar 4-1 Fungsi sebagai kotak hitam

Penggambaran tersebut sebagai kotak gelap diantaranya menjelaskan bahwa dalam fungsi pribagi bagi fungsi. Tidak ada suatu pernyataan diluar fungsi yang bisa mengakses bagian dalam fungsi, selain melalui parameter atau variabel eksternal dengan memanfaatkan perintah goto kita bisa menjalankan suatu fungsi, bentuk umum fungsi adalah sebagai berikut :

```
tipe-keluaran-fungsi nama-fungsi(deklarasi argumen)
{ tubuh
    fungsi
}
```

Keterangan

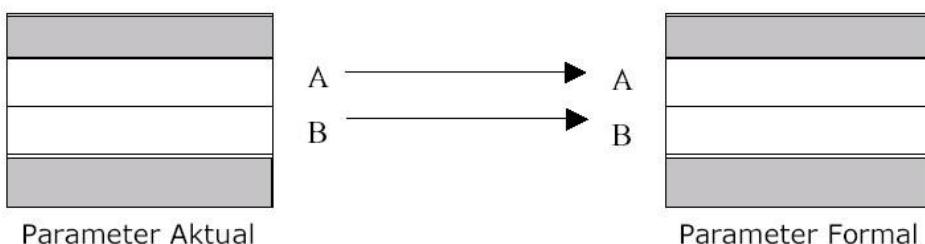
1. tipe-keluaran-fungsi dapat berupa salah satu tipe data dalam bahasa C, misalnya int
2. tubuh fungsi berisi deklarasi variabel dan statemen yang akan melakukan tugas yang akan diberikan kepada fungsi yang bersangkutan yang ditulis dalam tanda kurung kurawal

5.2.3. Cara melewatkkan parameter terhadap fungsi

Ada dua cara yang dapat dilakukan untuk melewatkkan parameter, yaitu :

1. Pemanggilan dengan nilai (call by value)

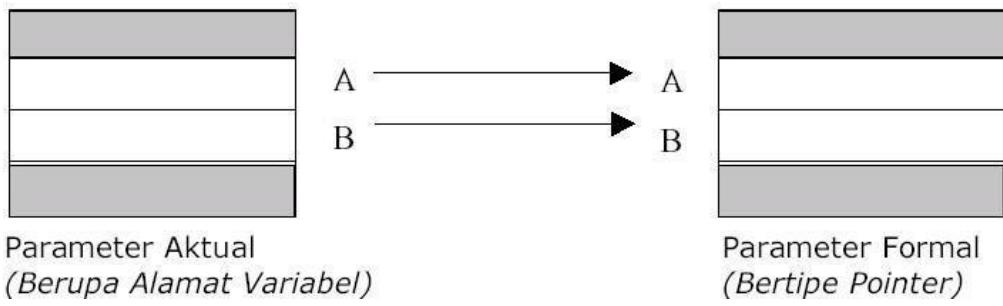
Nilai dari parameter aktual akan disalin dalam parameter formal, sehingga nilai pada parameter aktual tidak bisa berubah sekalipun nilai parameter formalnya mengalami perubahan.



gambar 4-2 Pemanggilan dengan nilai

2. Pemanggilan dengan referensi (call by reference)

Dengan melewatkkan alamat dari suatu variabel ke dalam fungsi. Dengan cara ini isi dari sebuah variabel dapat diubah diluar fungsi dengan proses perubahan didalam fungsi.



gambar 4-3 Pemanggilan dengan referensi

5.3. Alat dan Bahan

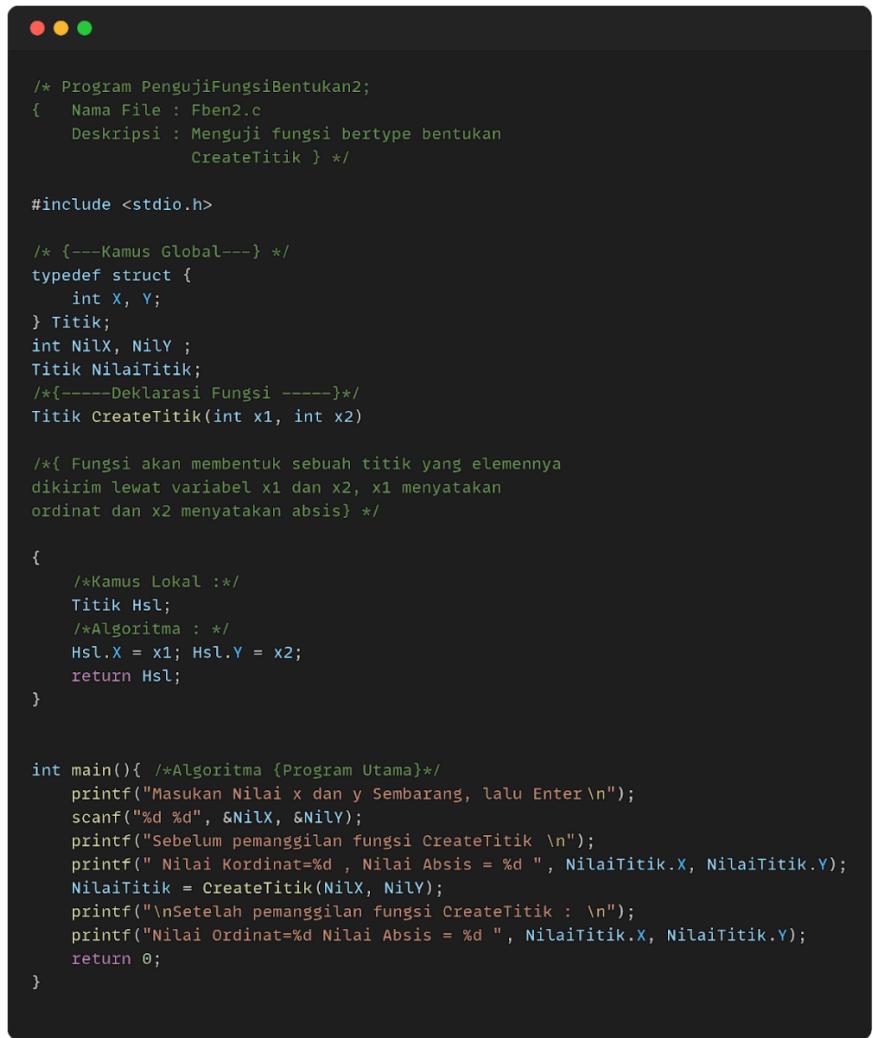
1. Seperangkat Komputer
2. Software Compiler C & Python (ex : Borland c/c++ atau dev c/c++)

5.4. Pengamatan dan Percobaan

5.4.1. Fungsi create titik

1. Nyalakan komputer yang ada dihadapan anda, biarkan sampai masuk kedalam sistem operasi windows
2. Bukalah program kompiler bahasa C & python yang ada dalam komputer tersebut.
3. Ketiklah kode sumber berikut ini dengan benar dan tepat

- titik.c



```
/* Program PengujI Fungsi Bentukan2;
{   Nama File : Fben2.c
    Deskripsi : Menguji fungsi bertipe bentukan
                CreateTitik } */

#include <stdio.h>

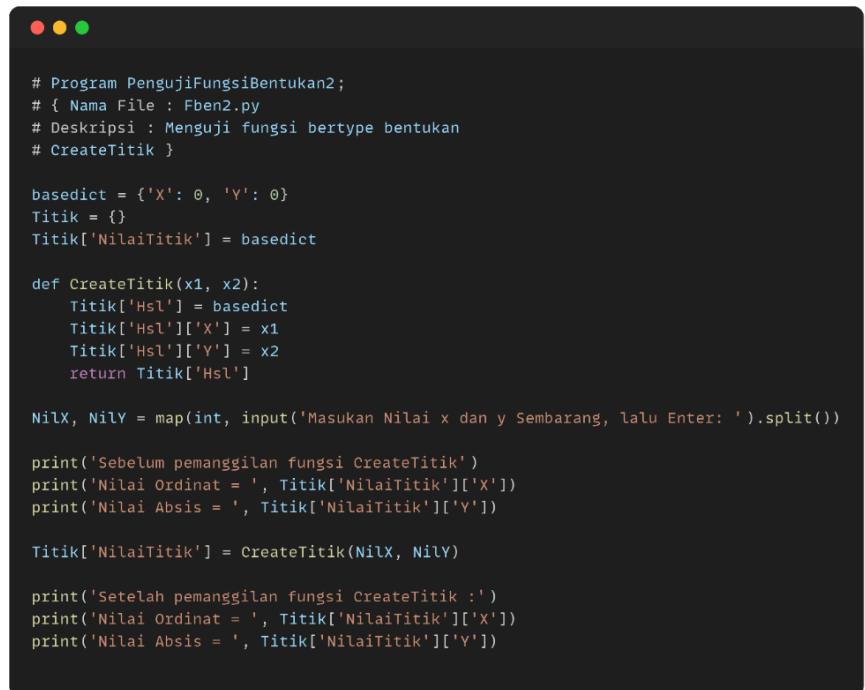
/* {---Kamus Global---} */
typedef struct {
    int X, Y;
} Titik;
int NilX, NilY ;
Titik NilaiTitik;
/*{-----Deklarasi Fungsi -----*/
Titik CreateTitik(int x1, int x2)

/*{ Fungsi akan membentuk sebuah titik yang elemennya
dikirim lewat variabel x1 dan x2, x1 menyatakan
ordinat dan x2 menyatakan absis} */

{
    /*Kamus Lokal :*/
    Titik Hsl;
    /*Algoritma : */
    Hsl.X = x1; Hsl.Y = x2;
    return Hsl;
}

int main(){ /*Algoritma {Program Utama}*/
    printf("Masukan Nilai x dan y Sembarang, lalu Enter\n");
    scanf("%d %d", &NilX, &NilY);
    printf("Sebelum pemanggilan fungsi CreateTitik \n");
    printf(" Nilai Kordinat=%d , Nilai Absis = %d ", NilaiTitik.X, NilaiTitik.Y);
    NilaiTitik = CreateTitik(NilX, NilY);
    printf("\nSetelah pemanggilan fungsi CreateTitik : \n");
    printf("Nilai Ordinat=%d Nilai Absis = %d ", NilaiTitik.X, NilaiTitik.Y);
    return 0;
}
```

- titik.py



```
# Program PengujifungsiBentukan2;
# { Nama File : Fben2.py
# Deskripsi : Menguji fungsi bertipe bentukan
# CreateTitik }

basedict = {'X': 0, 'Y': 0}
Titik = {}
Titik['NilaiTitik'] = basedict

def CreateTitik(x1, x2):
    Titik['Hsl'] = basedict
    Titik['Hsl']['X'] = x1
    Titik['Hsl']['Y'] = x2
    return Titik['Hsl']

NilX, NilY = map(int, input('Masukan Nilai x dan y Sembarang, lalu Enter: ').split())

print('Sebelum pemanggilan fungsi CreateTitik')
print('Nilai Ordinat = ', Titik['NilaiTitik']['X'])
print('Nilai Absis = ', Titik['NilaiTitik']['Y'])

Titik['NilaiTitik'] = CreateTitik(NilX, NilY)

print('Setelah pemanggilan fungsi CreateTitik :')
print('Nilai Ordinat = ', Titik['NilaiTitik']['X'])
print('Nilai Absis = ', Titik['NilaiTitik']['Y'])
```

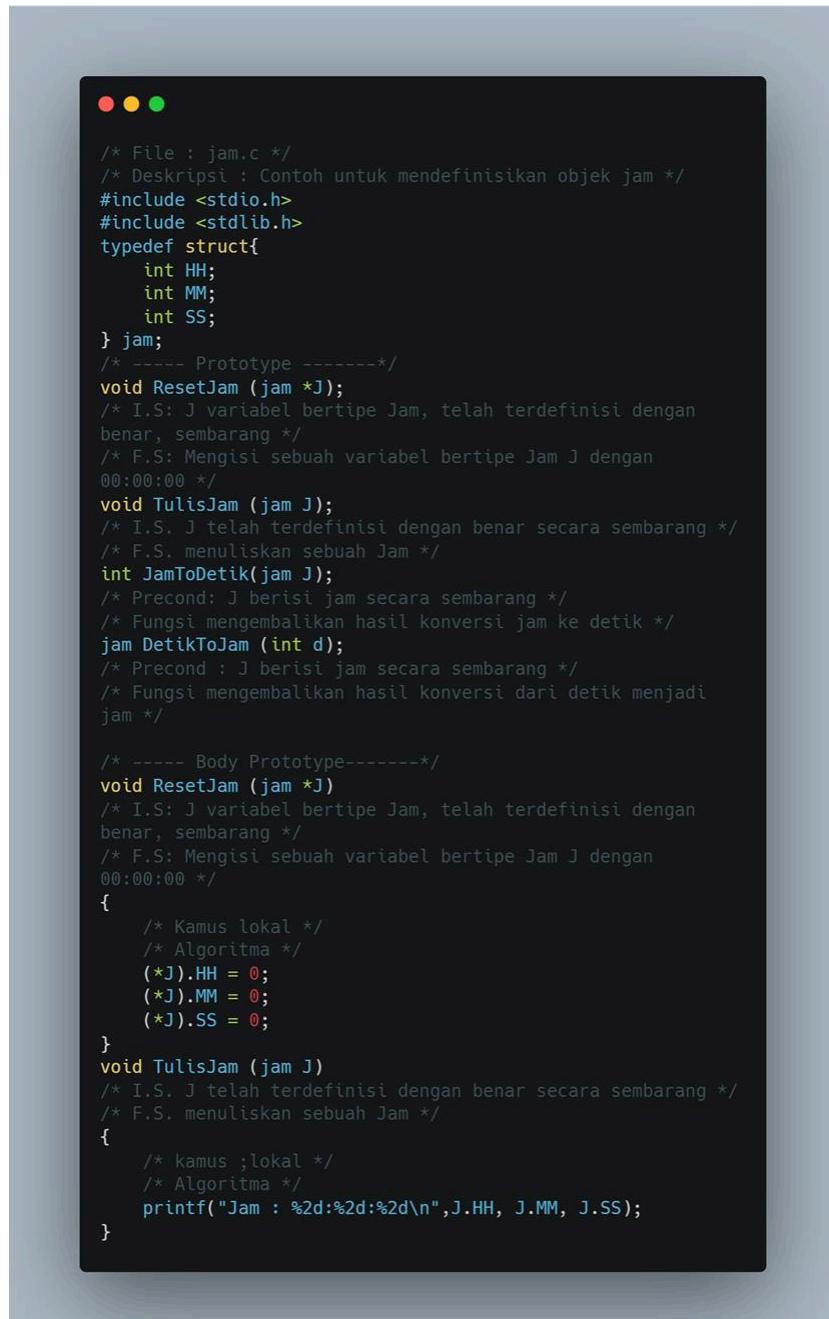
codesnap.dev

4. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera diatas, kemudian simpanlah dalam folder yang diberinama sesuai dengan NIM anda
5. Kompilasilah kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
6. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
7. Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error
8. Amati nilai NilaiTitik sebelum dan sesudah pemanggilan fungsi
9. Realisasikan algoritma dibawah ini

```
Function CreateTitik(x1 : integer; x2 : integer): PTitik;
{ Fungsi akan membentuk sebuah titik yang elemennya dikirim
lewat variabel x1 dan x2, x1 menyatakan ordinat dan x2
menyatakan absis}
Function TambahTitik(Titik1 : Ptitik; Titik2 : PTitik) :
Ptitik;
{Fungsi akan mengembalikan hasil penambahan elemen bersesuaian
antara Titik1 dengan Titik2, Absis Titik1 ditambah absis pada
Titik2, Ordinat Titik1 ditambah Ordinat Titik2 }
```

5.4.2. Implementasi prosedur dan fungsi dalam program jam

1. Dengan kompiler C yang masih terbuka
2. Ketiklah kode sumber berikut ini dengan benar dan tepat
 - titik2.c



```
/* File : jam.c */
/* Deskripsi : Contoh untuk mendefinisikan objek jam */
#include <stdio.h>
#include <stdlib.h>
typedef struct{
    int HH;
    int MM;
    int SS;
} jam;
/* ----- Prototype -----*/
void ResetJam (jam *J);
/* I.S: J variabel bertipe Jam, telah terdefinisi dengan
benar, sembarang */
/* F.S: Mengisi sebuah variabel bertipe Jam J dengan
00:00:00 */
void TulisJam (jam J);
/* I.S. J telah terdefinisi dengan benar secara sembarang */
/* F.S. menuliskan sebuah Jam */
int JamToDetik(jam J);
/* Precond: J berisi jam secara sembarang */
/* Fungsi mengembalikan hasil konversi jam ke detik */
jam DetikToJam (int d);
/* Precond : J berisi jam secara sembarang */
/* Fungsi mengembalikan hasil konversi dari detik menjadi
jam */

/* ----- Body Prototype-----*/
void ResetJam (jam *J)
/* I.S: J variabel bertipe Jam, telah terdefinisi dengan
benar, sembarang */
/* F.S: Mengisi sebuah variabel bertipe Jam J dengan
00:00:00 */
{
    /* Kamus lokal */
    /* Algoritma */
    (*J).HH = 0;
    (*J).MM = 0;
    (*J).SS = 0;
}
void TulisJam (jam J)
/* I.S. J telah terdefinisi dengan benar secara sembarang */
/* F.S. menuliskan sebuah Jam */
{
    /* kamus ;lokal */
    /* Algoritma */
    printf("Jam : %d:%d:%d\n", J.HH, J.MM, J.SS);
}
```

```
● ● ●

int JamToDetik(jam J)
/* Precond: J berisi jam secara sembarang */
/* Fungsi mengembalikan hasil konversi jam ke detik */
{
    /* Kamus Lokal */
    /* Algoritma */
    return (J.HH * 3600 + J.MM * 60 + J.SS);
}

jam DetikToJam (int d)
/* Precond : J berisi jam secara sembarang */
/* Fungsi mengembalikan hasil konversi dari detik menjadi jam */
{
    /* Kamus Lokal */
    jam J;
    int sisa;
    /* Algoritma */
    J.HH = d / 3600;
    sisa = d % 3600;
    J.MM = sisa / 60;
    J.SS = sisa % 60;
    return J;
}
/*-----Driver Jam-----*/
int main()
{
    /* Kamus Lokal */
    jam J1;
    jam J2;
    int dt=30000;
    /* Algoritma */
    printf("Asalamualaikum \n");
    ResetJam(&J1);
    J1.HH=8;
    J1.MM=50;
    TulisJam(J1);
    printf("Konversi jam ke detik : %d\n",JamToDetik(J1));
    printf("\n===== \n");
    printf("%d detik = ", dt);
    J2=DetikToJam(dt);
    TulisJam(J2);
    system ("pause");
    return 0;
}
```

- titik2.py

```
# File : jam.c
# Deskripsi : Contoh untuk mendefinisikan objek jam
basedict = {'HH': 0, 'MM': 0, 'SS': 0}
jam = {}
jam['J'] = basedict
# ----- Prototype -----
# ----- Body Prototype-----
def ResetJam (HH,MM,SS):
    # I.S: J variabel bertipe Jam, telah terdefinisi dengan benar, sembarang
    # F.S: Mengisi sebuah variabel bertipe Jam J dengan 00:00:00
    # Kamus lokal
    # Algoritma
    HH = 0
    MM = 0
    SS = 0

def TulisJam (HH,MM,SS):
    # I.S. J telah terdefinisi dengan benar secara sembarang
    # F.S. menuliskan sebuah Jam
    # kamus ;lokal
    # Algoritma
    print('Jam : ',HH,MM,SS)

def JamToDetik(HH,MM,SS):
    # Precond: J berisi jam secara sembarang
    # Fungsi mengembalikan hasil konversi jam ke detik
    # Kamus Lokal
    # Algoritma
    return (HH * 3600 + MM * 60 + SS);
def DetikToJam (d):
    # Precond : J berisi jam secara sembarang
    # Fungsi mengembalikan hasil konversi dari detik menjadi jam
    # Kamus Lokal
    jam['J'] = basedict
    # Algoritma
    jam['J']['HH'] = d / 3600
    sisa = d % 3600
    jam['J']['MM'] = sisa / 60
    jam['J']['SS'] = sisa % 60
    return jam['J']

#-----Driver Jam-----
def main():
    # Kamus Lokal
    jam['J1'] = basedict
    jam['J2'] = basedict
    dt=30000
    # Algoritma
    print('Asalamualaikum')
    ResetJam(**jam['J1'])
    jam['J1']['HH']=8;
    jam['J1']['MM']=50;
    TulisJam(**jam['J1'])
    print('Konversi jam ke detik:',JamToDetik(**jam['J1']))
    print('=====')
    print(' detik = ', dt)
    jam['J2'] = DetikToJam(dt)
    TulisJam(**jam['J2'])
if __name__ == "__main__":
    main()
```

3. Setelah anda selesai mengetik, simpanlah kode sumber tersebut dengan nama sesuai dengan yang tertera diatas, kemudian simpanlah dalam folder yang diberinama sesuai dengan NIM anda
4. Kompilasilah kode sumber tersebut, jika mengalami kesulitan dalam langkah ini anda dapat meminta bantuan asisten
5. Ketika anda melakukan kompilasi dan jika masih ditemukan kesalahan, periksalah kembali kode sumber yang anda ketikan, dan simpan kembali hasil modifikasi tersebut
6. Lakukanlah langkah 6 ini sampai kode sumber anda dapat dikompilasi dengan benar dan tidak ditemukan kode yang error

5.5. Pertanyaan

1. Jelaskan dan sebutkan tujuan digunakannya sebuah fungsi dan prosedur?
2. Jelaskan dengan bahasa anda sendiri mengenai konsep dan penggunaan dari typedef struct?
3. Apa keuntungan dan kerugian penggunaan fungsi rekursif dibandingkan dengan iterasi dalam pemrograman?
4. Buatlah sebuah fungsi yang digunakan untuk menghitung akar-akar persamaankuadrat dimana fungsi tersebut bersifat float yaitu bilangan yang akan dikuadratkan serta memiliki nilai return value bersifat float
5. Tulislah prototipenya
6. Buatlah fungsi utama untuk memanggil fungsi kuadrat tersebut dengan bilanganx yang akan dicari diinputkan dari user

UNIT VI: DASAR ALGORTIMA DAN STRUKTUR DATA PYTHON

6.1. Tujuan Praktikum

1. Mahasiswa mampu menkonstruksi proses rekursif
2. Mahasiswa mampu menghitung faktorial dengan library Math
3. Mahasiswa mampu menghitung faktorial dengan loop iteratif
4. Mahasiswa mampu menghitung faktorial dengan rekursif
5. Mahasiswa dapat memahami Sorting, Tuples, Sets, List Searching, dan Lambda pada Python 3
6. Mahasiswa dapat memahami konsep Object Oriented Programming pada Python 3
7. Mahasiswa dapat memahami Double Linked List & Object Relation pada Python 3

6.2. Dasar Teori

Bahasa pemrograman Python dalam banyak hal berbeda dengan bahasa pemrograman prosedural, seperti C/C++. Dalam C/C++, file source code harus dikompilasi ke dalam bentuk executable file sebelum dijalankan. Pada Python, tidak terdapat langkah kompilasi, sebagai gantinya source code ditafsirkan secara langsung baris demi baris (interpretasi).

Keunggulan utama dari suatu bahasa pemrograman *Interpreter* seperti Python adalah tidak membutuhkan pendeklarasian variabel, sehingga lebih fleksibel dalam penggunaannya. Namun, terdapat kelemahan yang mencolok, yaitu program – program numerik yang dijalankan pada Python lebih lambat ketimbang dijalankan menggunakan bahasa pemrograman terkompilasi.

Kelemahan ini tentu membuat kita berpikir apakah Python cocok untuk digunakan dalam komputasi ilmiah? Meskipun bekerja lebih lambat dibandingkan C/C++, Python memiliki banyak fungsi – fungsi sederhana yang dapat menjalankan hal – hal yang umumnya dikerjakan dengan subroutine rumit dalam C/C++. Sehingga Python merupakan pilihan tepat dalam komputasi ilmiah dewasa ini.

6.3 Alat dan Bahan

1. Komputer
2. IDE Python 3
3. Interpreter Python 3

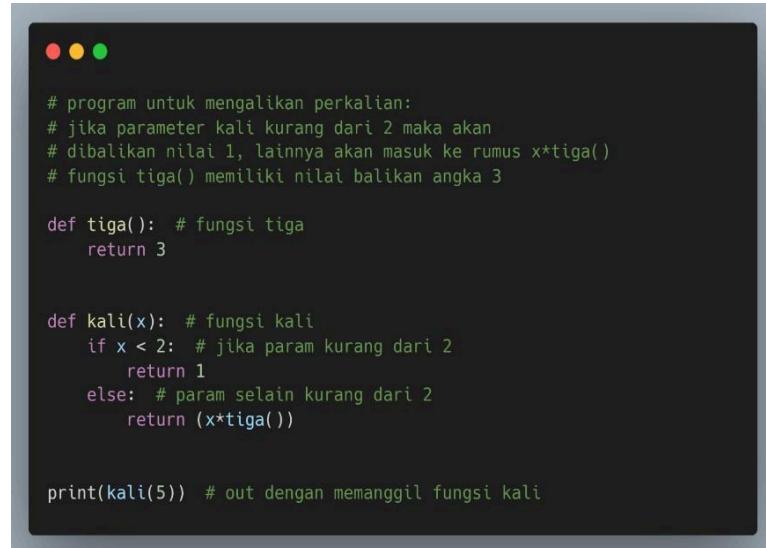
Anda bisa menggunakan IDE apapun untuk Python (Visual Studio Code, Anaconda, Python Online, atau bahkan Notepad & cmd)

6.4 Langkah Kerja

6.4.1 Function

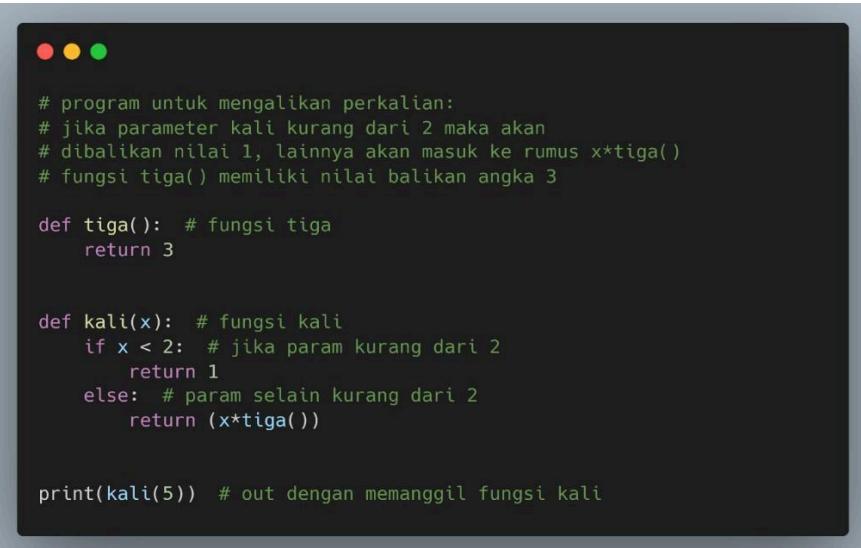
Masukkan Source Code ini pada Python, berikan comment, dan analisa outputnya!

1. Function_A.py



```
# program untuk mengalikan perkalian:  
# jika parameter kali kurang dari 2 maka akan  
# dibalikkan nilai 1, lainnya akan masuk ke rumus x*tiga()  
# fungsi tiga() memiliki nilai balikan angka 3  
  
def tiga(): # fungsi tiga  
    return 3  
  
def kali(x): # fungsi kali  
    if x < 2: # jika param kurang dari 2  
        return 1  
    else: # param selain kurang dari 2  
        return (x*tiga())  
  
print(kali(5)) # out dengan memanggil fungsi kali
```

2. Function_B.py

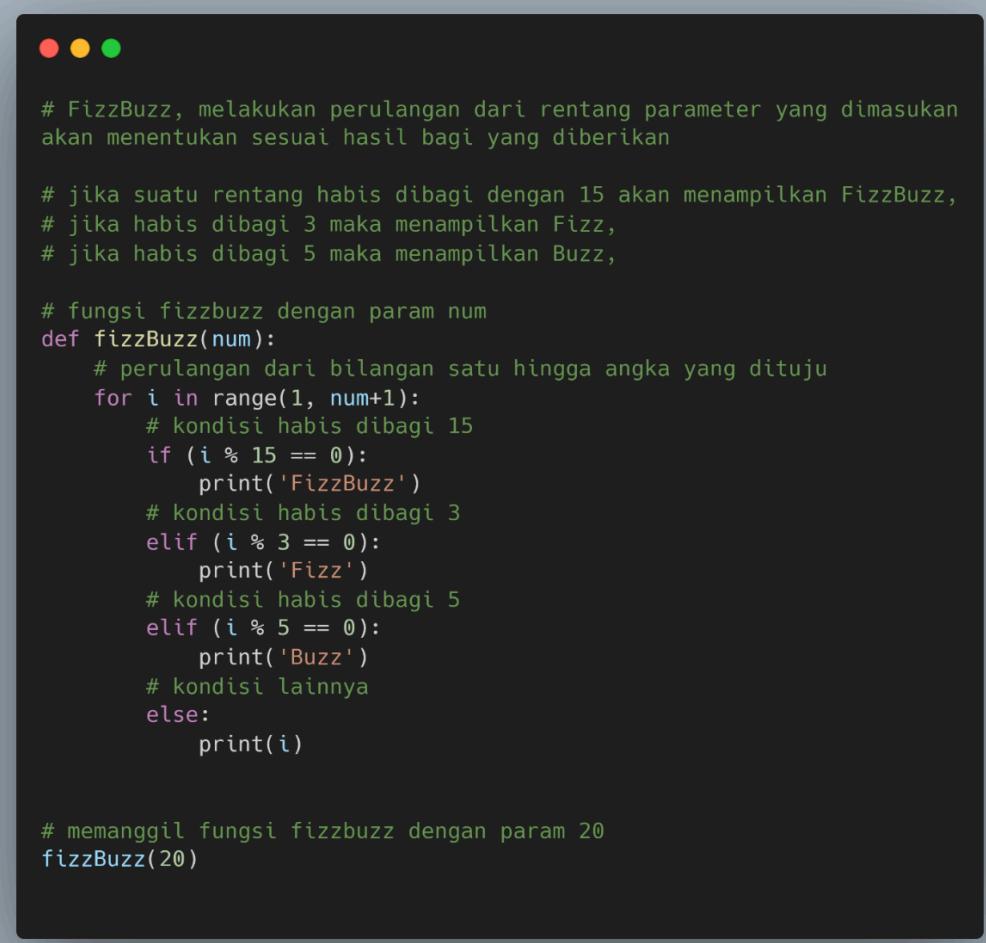


```
# program untuk mengalikan perkalian:  
# jika parameter kali kurang dari 2 maka akan  
# dibalikkan nilai 1, lainnya akan masuk ke rumus x*tiga()  
# fungsi tiga() memiliki nilai balikan angka 3  
  
def tiga(): # fungsi tiga  
    return 3  
  
def kali(x): # fungsi kali  
    if x < 2: # jika param kurang dari 2  
        return 1  
    else: # param selain kurang dari 2  
        return (x*tiga())  
  
print(kali(5)) # out dengan memanggil fungsi kali
```

3. Function_A_Modif.py

Modifikasi source code dari Function_A agar dapat menghitung luas segitiga dari input data 3 sisi segitiga (gunakan function).

4. Fizz_Buzz.py



```
# FizzBuzz, melakukan perulangan dari rentang parameter yang dimasukan akan menentukan sesuai hasil bagi yang diberikan

# jika suatu rentang habis dibagi dengan 15 akan menampilkan FizzBuzz,
# jika habis dibagi 3 maka menampilkan Fizz,
# jika habis dibagi 5 maka menampilkan Buzz,

# fungsi fizzbuzz dengan param num
def fizzBuzz(num):
    # perulangan dari bilangan satu hingga angka yang dituju
    for i in range(1, num+1):
        # kondisi habis dibagi 15
        if (i % 15 == 0):
            print('FizzBuzz')
        # kondisi habis dibagi 3
        elif (i % 3 == 0):
            print('Fizz')
        # kondisi habis dibagi 5
        elif (i % 5 == 0):
            print('Buzz')
        # kondisi lainnya
        else:
            print(i)

# memanggil fungsi fizzbuzz dengan param 20
fizzBuzz(20)
```

6.4.2 List

Masukkan Source Code ini pada Python, berikan comment, dan analisa outputnya!

1. List.py

```
● ● ●

# program untuk membuat list dan operasi pada list
# membuat data dan datalist
Laptop1 = 'Acer'
Laptop2 = 'Dell'
Laptop3 = 'Lenovo'
Laptop = ['Acer', 'Dell', 'Lenovo']
Laptop = [
    'Acer',
    'Dell',
    'Lenovo']
print('-----')
# menampilkan elemen list laptop
print(Laptop)
# menampilkan elemen list laptop sesuai indeks
print(Laptop[0])
print(Laptop[1])
print(Laptop[2])
print('-----')
# menampilkan elemen list laptop secara berurutan
for merk in Laptop:
    print(merk)
print('-----')
# menampilkan elemen list laptop indeks lebih dari indeks pertama
print(Laptop[1:])
# menampilkan elemen list laptop indeks kurang dari indeks kedua
print(Laptop[:2])
# menampilkan elemen list laptop indeks diantara indeks pertama dan
# ketiga
print(Laptop[1:3])
# menampilkan elemen list laptop seluruh indeks
print(Laptop[:])
print('-----')
# menimpa elemen indeks pertama dengan data baru
Laptop[1] = 'Razer'
print(Laptop)
```

2. List_Functions.py

```
# Program list untuk menggabungkan tipedata list dengan berbagai fungsi

# membuat tipedata list
Handphone = ['Xiaomi', 'Realme', 'Xiaomi', 'Huawei']

print('-----copy-----')
# list Handphone menggunakan fungsi copy untuk list HP_Cina
print(Handphone)
HP_Cina = Handphone.copy()
print(HP_Cina)

print('-----clear-----')
# list Handphone menggunakan fungsi clear untuk membersihkan list
Handphone.clear()
print(Handphone)

print('-----append-----')
# list Handphone menggunakan fungsi append() untuk menambahkan
# 'Samsung' pada list
Handphone.append('Samsung')
print(Handphone)

print('-----pop-----')
# list HP_Cina menggunakan fungsi pop() (default pop di indeks
terakhir)
HP_Cina.pop()
print(HP_Cina)

print('-----extend-----')
# menggabung tipedata list Handphone dengan fungsi ekstend list HP_Cina
Handphone.extend(HP_Cina)
print(Handphone)

print('-----count-----')
# list Handphone menggunakan fungsi count() untuk menghitung data
"Xiaomi"
mi = Handphone.count("Xiaomi")
print(mi)

print('-----index-----')
# list Handphone menggunakan fungsi index() untuk mencari posisi elemen
# yang dicari
ei = Handphone.index("Samsung")
print(ei)

print('-----insert-----')
# list Handphone dengan fungsi insert() untuk memasukan elemen iPhone
di
# indeks ke-1
# insert(indeks,objek -> str)
Handphone.insert(1, "iPhone")
print(Handphone)

print('-----remove-----')
# list Handphone menggunakan fungsi remove() untuk elemen yang akan
dihapus
# yang dihapus elemen pertama yang terkena
Handphone.remove("Xiaomi")
print(Handphone)

print('-----reverse-----')
# list Handphone menggunakan fungsi reverse() memutar semua elemen
terbalik
Handphone.reverse()
print(Handphone)

print('-----sort-----')
# list handpone menggunakan fungsi sort() sortir elemen
# secara default sortir secara ascending, disini berdasarkan abjad
Handphone.sort()
print(Handphone)
```

3. Dictionaries.py

```
● ● ●

# program untuk program kamus yang merupakan kumpulan pasangan kunci
OS = {"Microsoft": ["Windows XP", 7, "Windows Vista", 8, 10],
       "Apple": ["Yosemite", "Mavericks", "Panther", "Kodiak"],
       "Linux": ["Ubuntu", "Mint", "Kali", "ElementaryOS"]}
print(OS["Microsoft"])
print(OS["Apple"])
print(OS["Linux"])
print(OS["Apple"][1])
```

4. Fibonacci_Series.py

```
● ● ●

# program untuk menampilkan data fibonacci series urutan terakhir
# rumus dari fibonacci series yaitu f(n-1)+f(n-2)
# dengan f0 dan f1 secara urut adalah 0 dan 1
# fungsi fibo dengan param urut -> any
def fibo(urut):
    # membuat data list
    listData = [1, 1]
    # perulangan range indeks 2 hingga param urut
    # indeks list[0,1,2,dst.]
    for i in range(2, urut):
        # penambahan hasil rumus fibonacci ke listData
        listData.append(listData[i-2]+listData[i-1])
    # mengembalikan nilai fibonacci terakhir
    return listData[urut-1]

# menampilkan fungsi fibo param 8
print(fibo(8))
```

5. Reverse_List.py

```
● ● ●

1 import math
2 def reverseList(theList) :
3     for i in range(0, math.floor(len(theList)/2)) :
4         tempList = theList[i]
5         theList[i] = theList[len(theList) - 1 - i]
6         theList[len(theList) - 1 - i] = tempList
7     return theList
8 print(reverseList([1,2,3,4,5,6,7,8]))
```

6. Bubble_Sort.py

```
● ● ●

# program melakukan sorting menggunakan metode bubble

# membuat data list
x = [6000, 34, 203, 3, 746, 200, 984, 198, 764, 9, 1]

def bubbleSort(list): # fungsi bubblesort dengan param list
    # perulangan untuk iterasi
    for i in range(len(list), 0, -1):
        # perulangan untuk pasangan elemen list
        for j in range(0, i-1):
            # komparasi elemen pertama dan kedua
            if (list[j] > list[j+1]):
                # melakukan swap jika elemen pertama lebih dari elemen
                # kedua
                temp = list[j]
                list[j] = list[j + 1]
                list[j + 1] = temp
            # mengembalikan nilai list
    return list

# menampilkan nilai list dengan fungsi bubbleSort dengan param list x
print(bubbleSort(x))
```

6.4.3 Tuples, Sets, Lambda, & List Searching

Masukkan Source Code ini pada Python, berikan comment, dan analisa outputnya! [Jika terdapat error, maka jelaskan mengapa terjadi error tersebut dan perbaikilah source code-nya!]

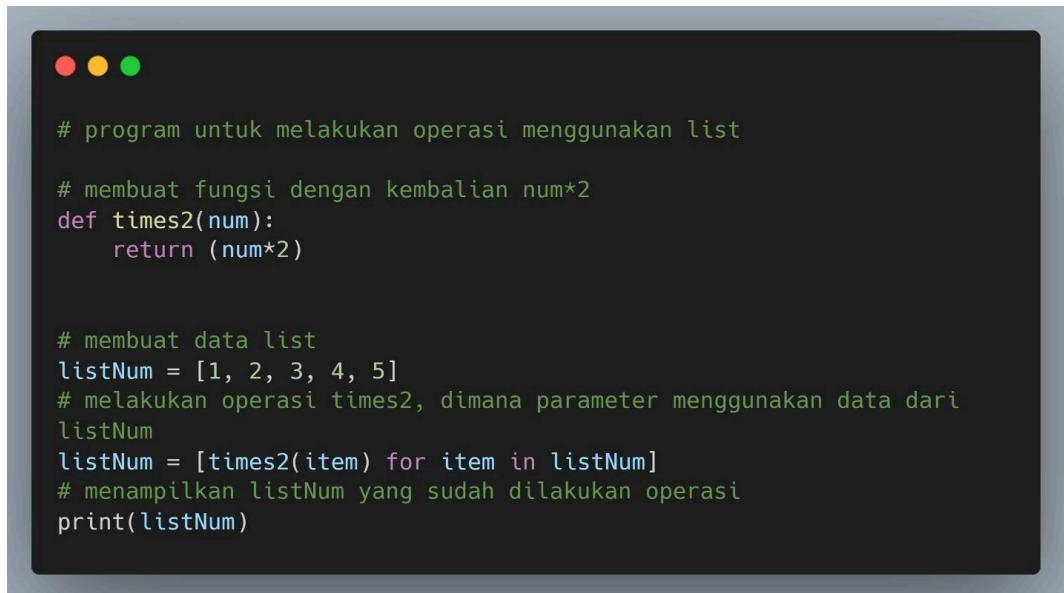
1. Tuples.py

```
# program tipedata tuples, tuples memiliki sifat berurutan, immutable,  
# dan dapat diduplikat  
t = (1, [0, "test"], {"a1": True})  
print(t[2]["a1"])  
print(t[1][1])  
t[1][1] = "akan"  
print(t[1][1])  
t[1][0] = "siti"  
print(t[1])  
  
t = (1, [0, "test"], {"a1": True}, (0, {"test": 5}, 2))  
print(t[3][1]["test"])
```

2. Sets.py

```
# program latihan tipedata sets, memiliki sifat tidak berurutan,  
# tidak dapat diduplikat (unik), dapat diubah setnya,  
# tetapi tidak dapat diubah elemennya  
s = {1, 3, 1, 2, 2, 3, 1, 2, 3, 1, 2, 3}  
print(s)  
print(list(s)[2])  
newList = [1, 3, "test1", "test2", 2, 3, "test1"]  
s = set(newList)  
print(s)  
print(list(s)[2])
```

3. List_Operation.py

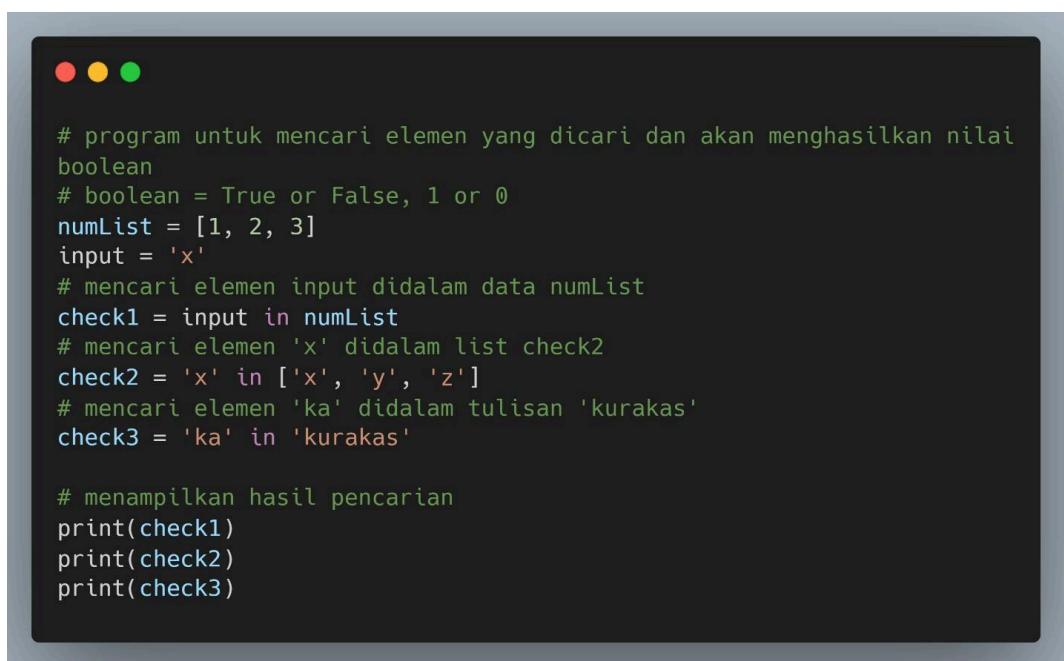


```
# program untuk melakukan operasi menggunakan list

# membuat fungsi dengan kembalian num*2
def times2(num):
    return (num*2)

# membuat data list
listNum = [1, 2, 3, 4, 5]
# melakukan operasi times2, dimana parameter menggunakan data dari
listNum
listNum = [times2(item) for item in listNum]
# menampilkan listNum yang sudah dilakukan operasi
print(listNum)
```

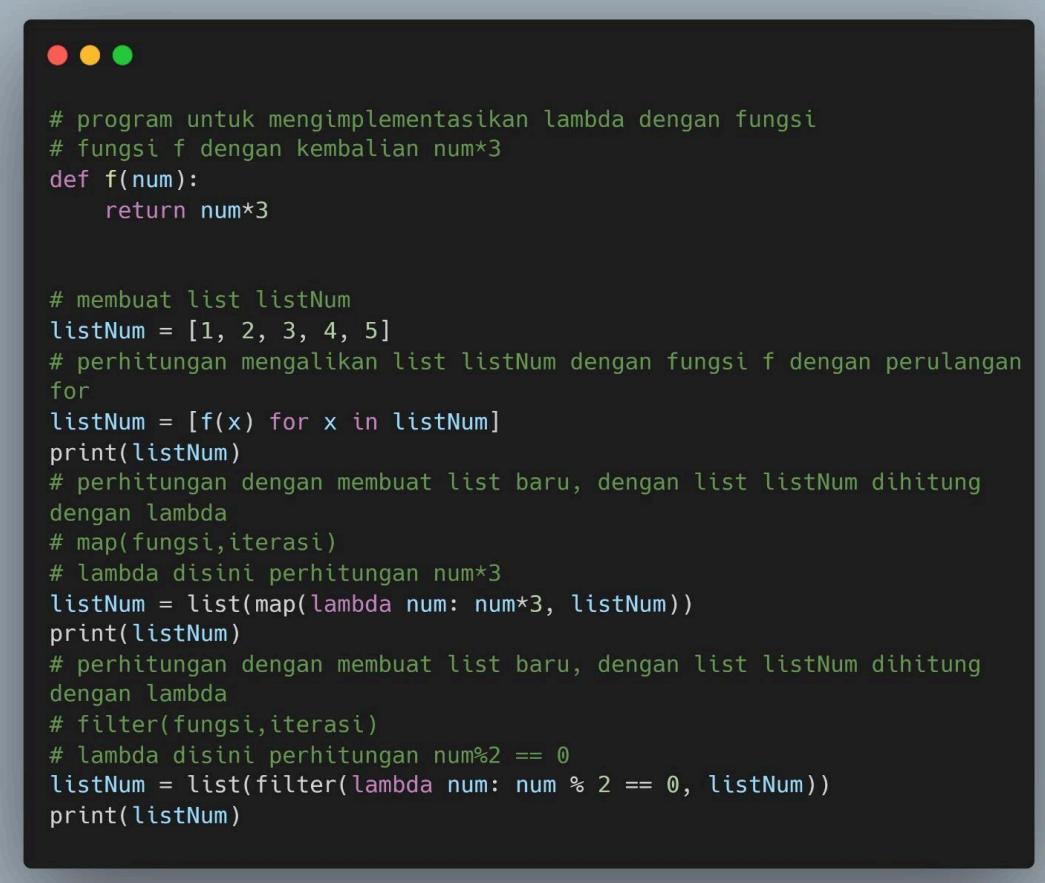
4. List Searching.py



```
# program untuk mencari elemen yang dicari dan akan menghasilkan nilai
boolean
# boolean = True or False, 1 or 0
numList = [1, 2, 3]
input = 'x'
# mencari elemen input didalam data numList
check1 = input in numList
# mencari elemen 'x' didalam list check2
check2 = 'x' in ['x', 'y', 'z']
# mencari elemen 'ka' didalam tulisan 'kurakas'
check3 = 'ka' in 'kurakas'

# menampilkan hasil pencarian
print(check1)
print(check2)
print(check3)
```

5. Lambda.py

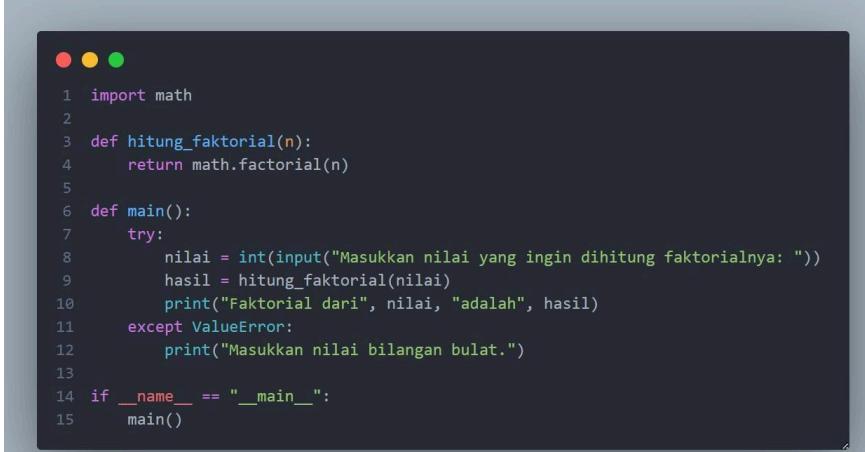


```
# program untuk mengimplementasikan lambda dengan fungsi
# fungsi f dengan kembalian num*3
def f(num):
    return num*3

# membuat list listNum
listNum = [1, 2, 3, 4, 5]
# perhitungan mengalikan list listNum dengan fungsi f dengan perulangan
for
listNum = [f(x) for x in listNum]
print(listNum)
# perhitungan dengan membuat list baru, dengan list listNum dihitung
dengan lambda
# map(fungsi,iterasi)
# lambda disini perhitungan num*3
listNum = list(map(lambda num: num*3, listNum))
print(listNum)
# perhitungan dengan membuat list baru, dengan list listNum dihitung
dengan lambda
# filter(fungsi,iterasi)
# lambda disini perhitungan num%2 == 0
listNum = list(filter(lambda num: num % 2 == 0, listNum))
print(listNum)
```

6.4.4 Recursive dengan Python

1. faktorial_math.py



```
1 import math
2
3 def hitung_faktorial(n):
4     return math.factorial(n)
5
6 def main():
7     try:
8         nilai = int(input("Masukkan nilai yang ingin dihitung faktorialnya: "))
9         hasil = hitung_faktorial(nilai)
10        print("Faktorial dari", nilai, "adalah", hasil)
11    except ValueError:
12        print("Masukkan nilai bilangan bulat.")
13
14 if __name__ == "__main__":
15     main()
```

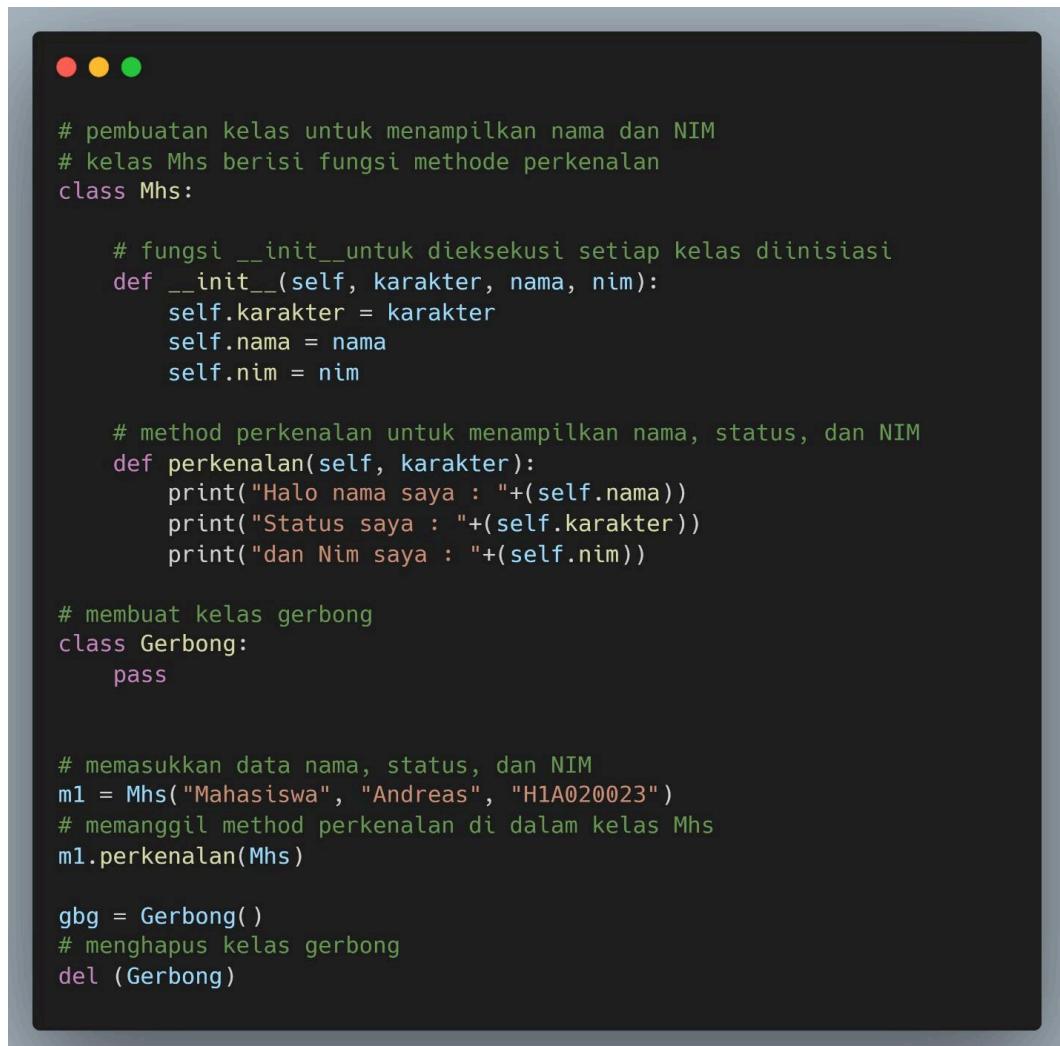
2. faktorial_recursive.py

```
● ● ●
1 import math
2
3 def hitung_faktorial(n):
4     if n == 0:
5         return 1
6     else:
7         return n * hitung_faktorial(n - 1)
8
9 def main():
10    try:
11        nilai = int(input("Masukkan nilai yang ingin dihitung faktorialnya: "))
12        if nilai < 0:
13            print("Masukkan nilai non-negatif.")
14        else:
15            hasil = hitung_faktorial(nilai)
16            print("Faktorial dari", nilai, "adalah", hasil)
17    except ValueError:
18        print("Masukkan nilai bilangan bulat.")
19
20 if __name__ == "__main__":
21    main()
```

6.4.5 Object Oriented Programming dengan Python

Masukkan Source Code ini pada Python, berikan comment, dan analisa outputnya! [Jika terdapat error, maka jelaskan mengapa terjadi error tersebut dan perbaikilah source code-nya!]

1. Class_Object.py



```
# pembuatan kelas untuk menampilkan nama dan NIM
# kelas Mhs berisi fungsi methode perkenalan
class Mhs:

    # fungsi __init__untuk dieksekusi setiap kelas diinisiasi
    def __init__(self, karakter, nama, nim):
        self.karakter = karakter
        self.nama = nama
        self.nim = nim

    # method perkenalan untuk menampilkan nama, status, dan NIM
    def perkenalan(self, karakter):
        print("Halo nama saya : "+(self.nama))
        print("Status saya : "+(self.karakter))
        print("dan Nim saya : "+(self.nim))

# membuat kelas gerbong
class Gerbong:
    pass

# memasukkan data nama, status, dan NIM
m1 = Mhs("Mahasiswa", "Andreas", "H1A020023")
# memanggil method perkenalan di dalam kelas Mhs
m1.perkenalan(Mhs)

gbg = Gerbong()
# menghapus kelas gerbong
del (Gerbong)
```

2. Class_Inheritance.py

```
# program class inheritance untuk menurunkan properti dan methode ke
# kelas lain
# kelas Mhs, berisi methode identitas
class Mhs:

    # fungsi __init__ untuk dieksekusi setiap kelas diinisiasi
    # properti ini akan dijadikan induk untuk kelas lain
    def __init__(self, info, nama, nim, univ, jurusan, angkatan):
        self.info = info
        self.nama = nama
        self.nim = nim
        self.univ = univ
        self.jurusan = jurusan
        self.angkatan = angkatan

    # methode identitas untuk menampilkan identitas
    def identitas(self, info):
        print("Menampilkan Bagian Identitas")
        print("Nama :" + self.nama)
        print("NIM : " + self.nim)
        print("\n")

# kelas Status, merupakan anak dari kelas Mhs,
# menurunkan properti dari kelas Mhs
class Status(Mhs):

    # methode detail untuk menampilkan informasi lanjutan
    def detail(self, info):
        print("Menampilkan Bagian Detail")
        print("Universitas:", self.univ)
        print("Jurusan:", self.jurusan)
        print("Angkatan:", self.angkatan)

    # memasukkan data untuk kelas Mhs
p1 = Mhs('Mahasiswa', 'Andreas', 'H1A020023',
         "Universitas Jenderal Soedirman", "Teknik Elektro", "2020")
# memanggil methode identitas pada kelas Mhs
p1.identitas('Mahasiswa')
# memasukkan data untuk kelas Status, properti kelas Status sama karena
# properti diturunkan dari kelas Mhs
p1 = Status('Mahasiswa', 'Andreas', 'H1A020023',
            "Universitas Jenderal Soedirman", "Teknik Elektro", "2020")
# memanggil methode detail pada kelas Status
p1.detail('Kuliah')
```

6.4.5 Double Linked List & Object Relations

Masukkan Source Code ini pada Python, berikan comment, dan tuliskan penjelasan yang lengkap tentang apa maksud dari kedua program di bawah ini!

1. Double_Linked_List.py



```
#melakukan simulasi double linked list pada python
import pickle

class DoublyNode(object):
    def __init__(self, val):
        self.val = val
        self.next = None
        self.prev = None

    def traversal(self):
        node = self
        while node != None:
            print(node.val)
            node = node.next

    def traverseprev(self):
        node = self

    def __del__(self):
        print(self.val, "gerbong meledak")

with open('company_datadn.pk1', 'wb') as output:
    company1 = DoublyNode(12)
    pickle.dump(company1, output, pickle.HIGHEST_PROTOCOL)
    company2 = DoublyNode(99)
    pickle.dump(company2, output, pickle.HIGHEST_PROTOCOL)
    company3 = DoublyNode(37)
    pickle.dump(company3, output, pickle.HIGHEST_PROTOCOL)

    company1.next = company2
    company2.next = company3
    company2.prev = company1
    company3.prev = company2

    lprev = company3
    lprev.traverseprev()
```

```
del company1
del company2
del company3

with open('company_datadn.pk1', 'rb') as input:
    company1 = pickle.load(input)
    print(company1.val)

    company2 = pickle.load(input)
    print(company2.val)

    company3 = pickle.load(input)
    print(company3.val)
```

Modifikasikanlah program ini agar dapat menampilkan info course yang telah dimasukkan datanya dengan menggunakan method pada Class!

2. Object_Relations.py

```
● ● ●

class Student:
    def __init__(self, name, student_number):
        self.name = name
        self.student_number = student_number
        self.classes = []

    def enrol(self, course_running):
        self.classes.append(course_running)
        course_running.add_student(self)

class department:
    def __init__(self, name, department_code):
        self.name = name
        self.department_code = department_code
        self.course = {}

    def add_course(self, description, course_code, credits):
        self.course[course_code] = Course(
            description, course_code, credits, self.name)
        return self.course[course_code]
```

```

class Course:
    def __init__(self, description, course_code, credits, department):
        self.description = description
        self.course_code = course_code
        self.credits = credits
        self.department = department
        self.running = []

    def add_running(self, year):
        self.running.append(CourseRunning(self, year))
        return self.running[-1]

class CourseRunning:
    def __init__(self, course, year):
        self.course = course
        self.year = year
        self.student = []

    def add_student(self, student):
        self.student.append(student)

electrical_engineering = department("Electrical Engineering", "TKE")
print(electrical_engineering.course)

TKE191226 = electrical_engineering.add_course(
    "Algorithm and Data Structures", "TKE191226", 1)
TKE191226_2020 = TKE191226.add_running(2020)

```

6.5 Pertanyaan

1. Buatlah program list sederhana dengan ketentuan sebagai berikut :
 - a. list berisikan kombinasi dari tipe data integer, string dan float. Isi list sebanyak 7 karakter
 - b. perintah 1 : menampilkan keseluruhan list
 - c. perintah 2 : menampilkan isi list indeks ke 5
 - d. perintah 3 : menampilkan panjang list
- ```

PS C:\Users\Bahari> & C:/Users/Bahari/AppData/Local/Programs/Python/Python310
/python.exe "c:/Users/Bahari/Downloads/soal_list.py"
list saya adalah : [12, 'ayam', 56.4, 'sapi', 456, 2, 7.5]
isi list indeks ke 5 adalah : 2
panjang list : 7
PS C:\Users\Bahari> []

```
2. Jelaskan perbedaan list dan tuples!
  3. Jelaskan konsep OOP pada python!
  4. Jelaskan pemahaman kalian tentang linked list!
  5. Apa itu function pada Python dan mengapa kita memerlukan function?

## **UNIT VII: CLASS GERBONG**

### **7.1 Tujuan Praktikum**

1. Mengenal Himpuan Perjalanan
2. Mengkonstruksi Himpunan Perjalanan
3. Merumuskan Algoritma
4. Menyusun Program
5. Mahasiswa mampu mendefinisikan sebuah Class.
6. Mahasiswa mampu menciptakan objek atau *instance* dari sebuah Class.
7. Mahasiswa mampu memahami penciptaan (*Create*) dan pemusnahan (*Destroy*) Objek.
8. Mahasiswa mampu memahami Constructor dan Destructor dari sebuah Class.

### **7.2 Dasar Teori**

#### **7.2.1 Pengenalan Konsep Class**

```
driverkereta.cpp gerbong.h gerbong.cpp
1 #ifndef GERBONG_H
2 #define GERBONG_H
3 #include <iostream>
4 using namespace std;
5 class Gerbong
6 {
7 public :
8 typedef struct {
9 int nokursi;
10 int statuskursi; //0 = kursi kosong; 1=kursi terisi orang/barang
11 } InfoKursi;
12 InfoKursi Kursi[4][10];
13 int NoGerbong;
14 public :
15 static int jumlahobject;
16 static int HitungObjek(void)
17 { return jumlahobject; }
18
19 Gerbong(void) {
20 // Constructor Class Gerbong, melakukan inisialisasi no gerbong dengan -
21 // dan menomori kursi setiap gerbong sejumlah 10 kursi dari no 1 sd 10
22 IsiGerbong(int _nogb, InfoKursi* _kursi);
23 // Kondisi Awal (IS) : Gerbong telah terdefinisi
24 // Kondisi Akhir (FS) : Gerbong terisi InfoKursi;
25 void CetakGerbong(void);
26 // Metod untuk melihat nomor gerbong dan statusnya seluruhnya
27 // gerbong dalam rangkaian Kereta
28 void PesanMakan(int nomer_kursi, string pesan);
29 ~Gerbong();
30 };
31 #endif
32
```

## 7.2.2 Penciptaan dan Pemusnahan Objek Tanpa Menggunakan Pointer

// Deklarasi Objek

Gerbong grb; // grb adalah objek atau instance

Object yang diciptakan tidak perlu dihapus dari driver, melainkan akan terhapus secara otomatis. Dan hal ini merupakan salah satu keuntungan menggunakan C++, karena pemrogram sudah tidak terlalu dirumitkan dengan penggunaan pointer (\*), walaupun programmer harus tetap memahaminya.

## 7.2.3 Penciptaan Objek Dengan Menggunakan Pointer

// Deklarasi Objek

Gerbong \*grb=new Gerbong();

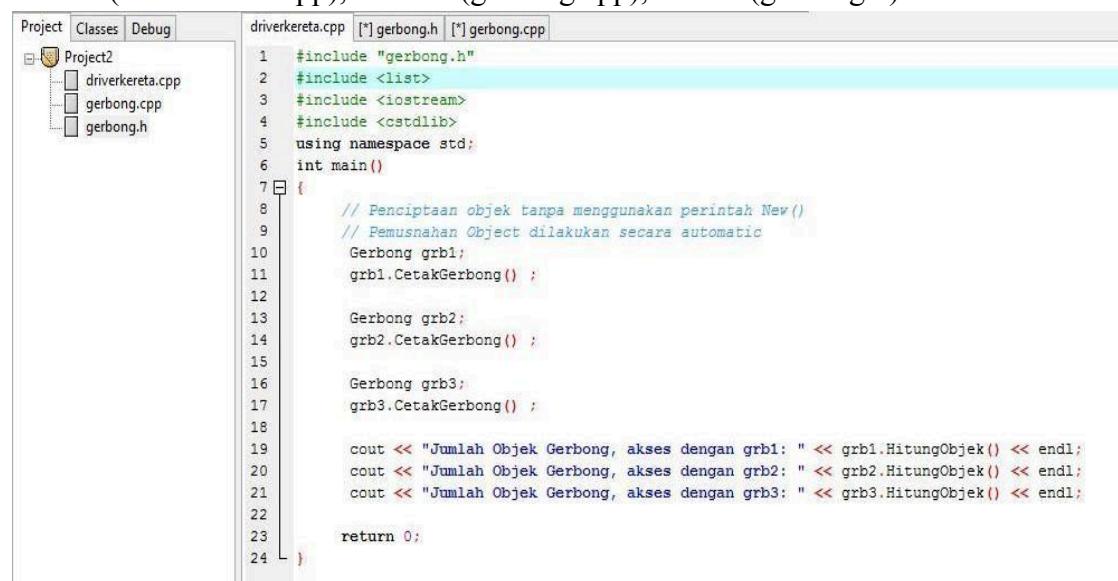
Objek ini harus dihilangkan dengan statement **delete grb**; yang diletakkan dibawah statement penciptaan objek secara simetris.

## 7.3 Alat dan Bahan

1. Komputer
2. IDE Dev C++ 5.4.2 lengkap dengan Library STL C++

## 7.4 Percobaan dan Pengamatan

1. Buatlah project31 seperti dibawah ini berisi file yang terkelompokan ke dalam driver  
(driverkereta.cpp), realisasi(gerbong.cpp), header (gerbong.h)



The screenshot shows the Dev C++ IDE interface. On the left is the Project Explorer with a single project named "Project2" containing three files: "driverkereta.cpp", "gerbong.cpp", and "gerbong.h". The main window displays the source code for "driverkereta.cpp". The code includes #include directives for "gerbong.h", <list>, <iostream>, and <cstdlib>. It uses the std namespace and defines a main() function. Inside main(), it creates three objects of type Gerbong: grb1, grb2, and grb3. Each object's CetakGerbong() method is called. Finally, it outputs the count of objects using cout. The code is color-coded with syntax highlighting.

```
Project Classes Debug
Project2
 - driverkereta.cpp
 - gerbong.cpp
 - gerbong.h

driverkereta.cpp [+] gerbong.h [+] gerbong.cpp
1 #include "gerbong.h"
2 #include <list>
3 #include <iostream>
4 #include <cstdlib>
5 using namespace std;
6 int main()
7 {
8 // Penciptaan objek tanpa menggunakan perintah New()
9 // Pemusnahan Object dilakukan secara automatic
10 Gerbong grb1;
11 grb1.CetakGerbong();
12
13 Gerbong grb2;
14 grb2.CetakGerbong();
15
16 Gerbong grb3;
17 grb3.CetakGerbong();
18
19 cout << "Jumlah Objek Gerbong, akses dengan grb1: " << grb1.HitungObjek() << endl;
20 cout << "Jumlah Objek Gerbong, akses dengan grb2: " << grb2.HitungObjek() << endl;
21 cout << "Jumlah Objek Gerbong, akses dengan grb3: " << grb3.HitungObjek() << endl;
22
23 return 0;
24 }
```

```
driverkereta.cpp [*]gerbong.h [*] gerbong.cpp
1 #ifndef GERBONG_H
2 #define GERBONG_H
3 #include <iostream>
4 using namespace std;
5 class Gerbong
6 {
7 public :
8 typedef struct {
9 int nokursi;
10 int statuskursi; //0 = kursi kosong; 1=kursi terisi orang/barang
11 } InfoKursi;
12 InfoKursi Kursi[4][10];
13 int NoGerbong;
14 public :
15 static int jumlahobject;
16 static int HitungObjek(void)
17 { return jumlahobject; }
18
19 Gerbong(void);
20 // Contructor Class Gerbong, melakukan inisialisasi no gerbong dengan -99
21 // dan menomori kursi setiap gerbong sejumlah 10 kursi dari no 1 sd 10
22 Gerbong IsiGerbong(int _nogb, InfoKursi* _kursi);
23 // Kondisi Aval (IS) : Gerbong telah terdefinisi
24 // Kondisi Akhir (FS) : Gerbong terisi InfoKursi;
25 void CetakGerbong(void);
26 // Metod untuk melihat nomor gerbong, no kursi beserta statusnya seluruh
27 // gerbong dalam rangkaian Kereta Api
28 void PesanMakan(int nomenu, int nokursi, int nogerbong);
29 ~Gerbong();
30 };
31 #endif
```

```
driverkereta.cpp [*]gerbong.h [*] gerbong.cpp
1 #include "gerbong.h"
2 int Gerbong::jumlahobject = 0;
3 Gerbong::Gerbong(void)
4 // Contructor Class Gerbong, melakukan inisialisasi no gerbong dengan -99
5 // dan menomori kursi setiap gerbong sejumlah 40 kursi dari no 1 sd 40, status kursi diset 0
6 {
7 int gb, kolom, baris;
8 jumlahobject++;
9 NoGerbong = -99;
10 kolom = 0;
11 while (kolom < 5) {
12 baris = 0;
13 while (baris < 2) {
14 Kursi[baris][kolom].nokursi = (2 * (kolom)) + (baris+1);
15 Kursi[baris][kolom].statuskursi = 0;
16 baris++;
17 }
18 kolom++;
19 }
20}
```

```
21
22 Gerbong::~Gerbong()
23 {
24 cout << "Hapus Object" << endl;
25 }
26
27 void Gerbong::CetakGerbong(void)
28 // Metod untuk menuliskan(ke layar)nomor gerbong, no kursi beserta statusnya
29 // seluruh gerbong dalam rangkaian Kereta Api yang sudah didefinisikan
30 {
31 int baris = 0;
32 cout << "NoGerbong : " << NoGerbong << endl;
33 while (baris < 2) {
34 int kolom = 0;
35 while (kolom < 5) {
36 cout << Kursi[baris][kolom].nokursi << "S" << Kursi[baris][kolom].statuskursi << "-";
37 kolom++;
38 }
39 cout << endl;
40 baris++;
41 }
42 }
43
```

: 0 Lines: 43 Length: 1074 Insert Done parsing



2. Kompilasilah project31, kemudian lakukan pengamatan debugging dan tracing dengan menggunakan tombol fungsi F7.

```
(globals)
Project Classes Debug
driverkereta.cpp gerbong.h gerbong.cpp
1 #include "gerbong.h"
2 #include <list>
3 #include <iostream>
4 #include <cstdlib>
5 using namespace std;
6 int main()
7 {
8 // Penciptaan objek tanpa menggunakan perintah New()
9 // Pemusnahan Object dilakukan secara automatic
10 Gerbong grb1;
11 grb1.CetakGerbong();
12
13 Gerbong grb2;
14 grb2.CetakGerbong();
15
16 Gerbong grb3;
17 grb3.CetakGerbong();
18
19 cout << "Jumlah Objek Gerbong, akses dengan grb1: " << grb1.MitingObjek();
20 cout << "Jumlah Objek Gerbong, akses dengan grb2: " << grb2.MitingObjek();
21 cout << "Jumlah Objek Gerbong, akses dengan grb3: " << grb3.MitingObjek();
22
23
24
Compiler Resources Compile Log Debug Find Results Close
Debug Add watch Next line Continue Next instruction
Stop Execution View CPU window Into function Skip function Into instruction
Evaluate:
Send command to GDB: break
<incomplete type>
->->value-history-end
->->pre-prompt
(gdb)
->->prompt
Line: 10 Col: 1 Sel: 0 Lines: 24 Length: 688 Insert Done parsing

```

3. Amatilah perubahan pengalamanan grb1, grb2, grb3, jumlahobject dengan menggunakan fasilitas **AddWatch**.

```
Project2 - [Project31.dev] - [Debugging] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tool CVS Window Help
File Project Class Debug
driverkereta.cpp gerbong.h gerbong.cpp
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int main()
5 {
6 Gerbong grb1;
7 grb1.CetakGerbong();
8
9 Gerbong grb2;
10 grb2.CetakGerbong();
11
12 Gerbong grb3;
13 grb3.CetakGerbong();
14
15 cout << "Jumlah Objek Gerbong, akses dengan grb1: " << grb1.MitingObjek() << endl;
16 cout << "Jumlah Objek Gerbong, akses dengan grb2: " << grb2.MitingObjek() << endl;
17 cout << "Jumlah Objek Gerbong, akses dengan grb3: " << grb3.MitingObjek() << endl;
18
19
20
21
22
23
24
return 0;

```

- Pada saat Tracing baris ke 13, double click object grb1 pada bagian kiri sehingga seperti gambar di bawah ini. Perhatikan berapakah nilai static jumlahobject ! Lihat pula jumlahobject untuk grb2, dan grb3. Berapakah nilainya ? samakah dengan grb1 ?

The screenshot shows a debugger window with the title "Project2 - [Project31.dev] - [Debugging]". The menu bar includes File, Edit, Search, View, Project, Execute, Tools, CVS, Window, and Help. The toolbar has various icons for file operations. The left pane displays a memory dump under the heading "(globals)" with numerous entries for 'statuskursi' and 'nokursi' variables. The right pane shows the source code of "driverkereta.cpp" with the following content:

```

1 #include "gerbong.h"
2 #include <list>
3 #include <iostream>
4 #include <cstdlib>
5 using namespace std;
6
7 int main()
8 {
9 // Penciptaan objek tanpa menggunakan perintah
10 // Pemusnahan Object dilakukan secara automati
11 Gerbong grb1;
12 grb1.CetakGerbong();
13
14 Gerbong grb2;
15 grb2.CetakGerbong();
16
17 Gerbong grb3;
18 grb3.CetakGerbong();
19
20 cout << "Jumlah Objek Gerbong, akses dengan g" << endl;
21 cout << "Jumlah Objek Gerbong, akses dengan g" << endl;
22 cout << "Jumlah Objek Gerbong, akses dengan g" << endl;
23
24 return 0;
}

```

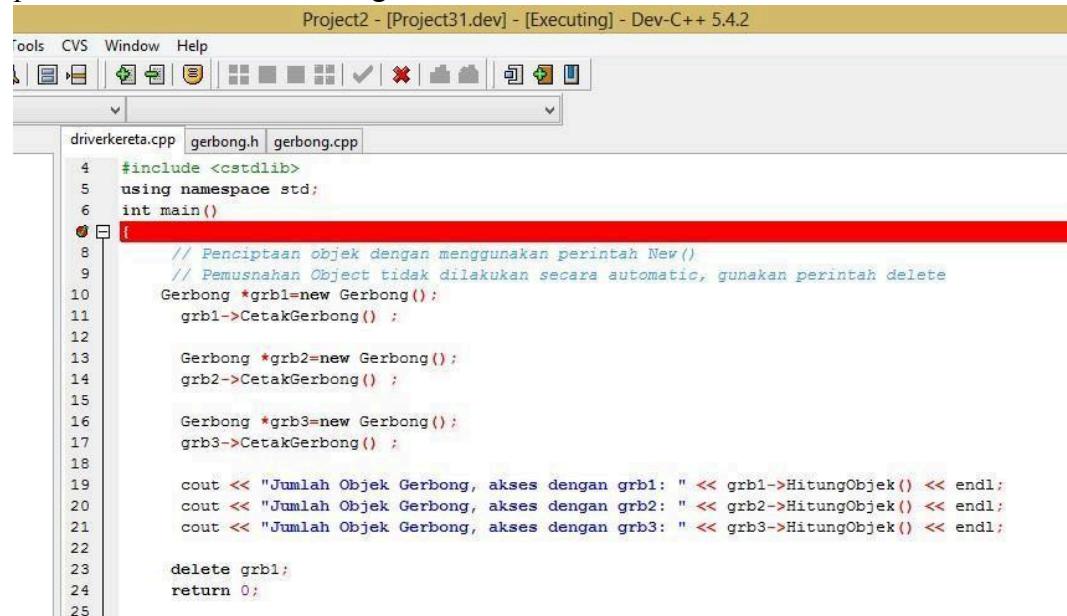
The cursor is positioned at line 13, where the variable 'grb1' is being traced. The status bar at the bottom shows Line: 13, Col: 1, Sel: 0, Lines: 24, Length: 688, Insert, Done parsing.

- Ulangi kembali langkah 2,3,4. Lakukan **tracing** menggunakan tombol fungsi F8.
- Pada saat kursor Tracing sedang mencapai baris ke 23, eksekusilah baris tersebut dengan tombol tracing F8. Kursor bar akan pindah ke baris 24 pada halaman file gerbong.cpp seperti di bawah ini.

```
driverkereta.cpp gerbong.h gerbong.cpp
1 #include "gerbong.h"
2 #include <list>
3 #include <iostream>
4 #include <cstdlib>
5 using namespace std;
6 int main()
7 {
8 // Penciptaan objek tanpa menggunakan perintah New
9 // Pemusnahan Object dilakukan secara automatic
10 Gerbong grb1;
11 grb1.CetakGerbong() ;
12
13 Gerbong grb2;
14 grb2.CetakGerbong() ;
15
16 Gerbong grb3;
17 grb3.CetakGerbong() ;
18
19 cout << "Jumlah Objek Gerbong, akses dengan grb1: " << grb1.HitungObjek() << endl;
20 cout << "Jumlah Objek Gerbong, akses dengan grb2: " << grb2.HitungObjek() << endl;
21 cout << "Jumlah Objek Gerbong, akses dengan grb3: " << grb3.HitungObjek() << endl;
22
23 return 0;
24 }
```

```
Project2 - [Project31.dev] - [Debugging] - Dev-C++ 5.4.2
File CVS Window Help
driverkereta.cpp gerbong.h gerbong.cpp
9 NoGerbong = -99;
10 kolom = 0;
11 while (kolom < 5) {
12 baris = 0;
13 while (baris < 2) {
14 Kursi[baris][kolom].nokursi = (2 * (kolom)) + (baris+1);
15 Kursi[baris][kolom].statuskursi = 0;
16 baris++;
17 }
18 kolom++;
19 }
20
21 Gerbong::~Gerbong ()
22 {
23 cout << "Hapus Object" << endl;
24 }
25
26
27 void Gerbong::CetakGerbong(void)
28 // Metod untuk menuliskan(ke layar)nomor gerbong, no kursi beserta stat
29 // seluruh gerbong dalam rangkaian Kereta Api yang sudah didefinisikan
30 {
31 int baris = 0;
32 cout << "NoGerbong : " << NoGerbong << endl;
33 while (baris < 2) {
34 cout << "Kursi : " << Kursi[baris][kolom].nokursi << endl;
35 cout << "Status Kursi : " << Kursi[baris][kolom].statuskursi << endl;
36 baris++;
37 }
38 }
```

7. Amatilah berapa kali method Gerbong::~Gerbong() dieksekusi !
8. Ulangi kembali langkah 1 sampai dengan langkah 7 dengan perubahan source code sebagai berikut.

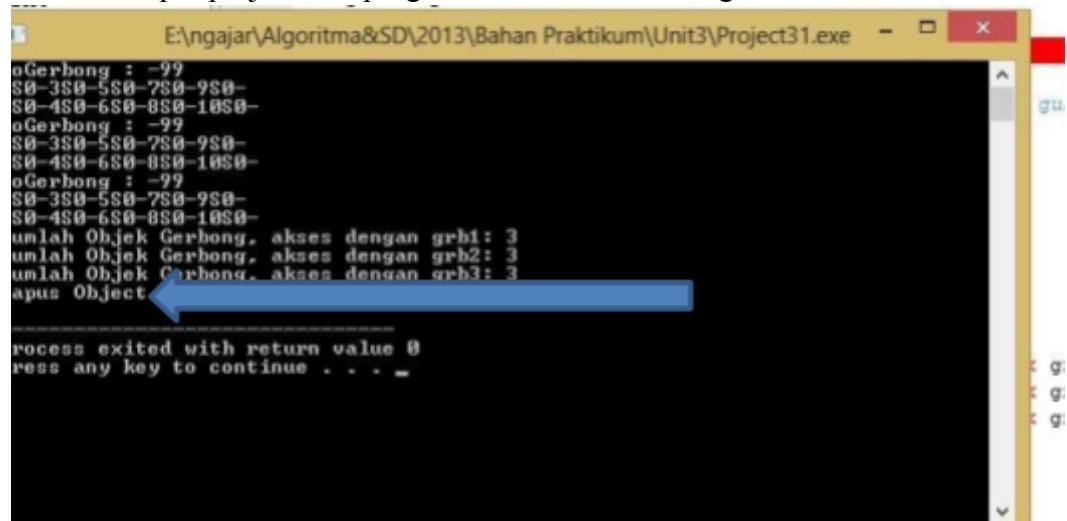


```

Project2 - [Project31.dev] - [Executing] - Dev-C++ 5.4.2
Tools CVS Window Help
File Project Run View Insert Editor Tools Options
driverkereta.cpp gerbong.h gerbong.cpp
4 #include <cstdlib>
5 using namespace std;
6 int main()
7 {
8 // Penciptaan objek dengan menggunakan perintah New()
9 // Pemusnahan Object tidak dilakukan secara automatic, gunakan perintah delete
10 Gerbong *grb1=new Gerbong();
11 grb1->CetakGerbong();
12
13 Gerbong *grb2=new Gerbong();
14 grb2->CetakGerbong();
15
16 Gerbong *grb3=new Gerbong();
17 grb3->CetakGerbong();
18
19 cout << "Jumlah Objek Gerbong, akses dengan grb1: " << grb1->HitungObjek() << endl;
20 cout << "Jumlah Objek Gerbong, akses dengan grb2: " << grb2->HitungObjek() << endl;
21 cout << "Jumlah Objek Gerbong, akses dengan grb3: " << grb3->HitungObjek() << endl;
22
23 delete grb1;
24 return 0;
25

```

9. Output project dari program di atas adalah sebagai berikut.



```

E:\ngajar\Algoritma&SD\2013\Bahan Praktikum\Unit3\Project31.exe
oGerbong : -99
$0-3S0-5S0-7S0-9S0-
$0-4S0-6S0-8S0-10S0-
oGerbong : -99
$0-3S0-5S0-7S0-9S0-
$0-4S0-6S0-8S0-10S0-
oGerbong : -99
$0-3S0-5S0-7S0-9S0-
$0-4S0-6S0-8S0-10S0-
jumlah Objek Gerbong, akses dengan grb1: 3
jumlah Objek Gerbong, akses dengan grb2: 3
jumlah Objek Gerbong, akses dengan grb3: 3
hapus Object
process exited with return value 0
press any key to continue . . .

```

10. Perhatikan Tulisan “Hapus Object” pada tampilan diatas, ada berapa baris ? Jika ingin menampilkan tulisan “Hapus Object” sejumlah object yang diciptakan, apa yang harus anda tambahkan pada program pada langkah 8 di atas.

## 7.5 Pertanyaan

1. Apa yang dilakukan oleh constructor *Gerbong::Gerbong(void)* pada program driverkereta.cpp?
2. Apa fungsi dari variable jumlahobject yang dideklarasikan sebagai static int pada class Gerbong?
3. Jika anda menciptakan objek A, B, C, D, E (tanpa pointer) bertipe Class Gerbong pada percobaan unit 3, maka berapakah pemanggilan *destructor* ?.
4. Apa perbedaan fungsi tombol F7 dan F8 pada aktifitas *Debugging* ?
5. Mengapa nilai **jumlahobject** diseluruh object grb1, grb2, grb3 selalu sama jika telah diciptakan minimal 1 objek ?

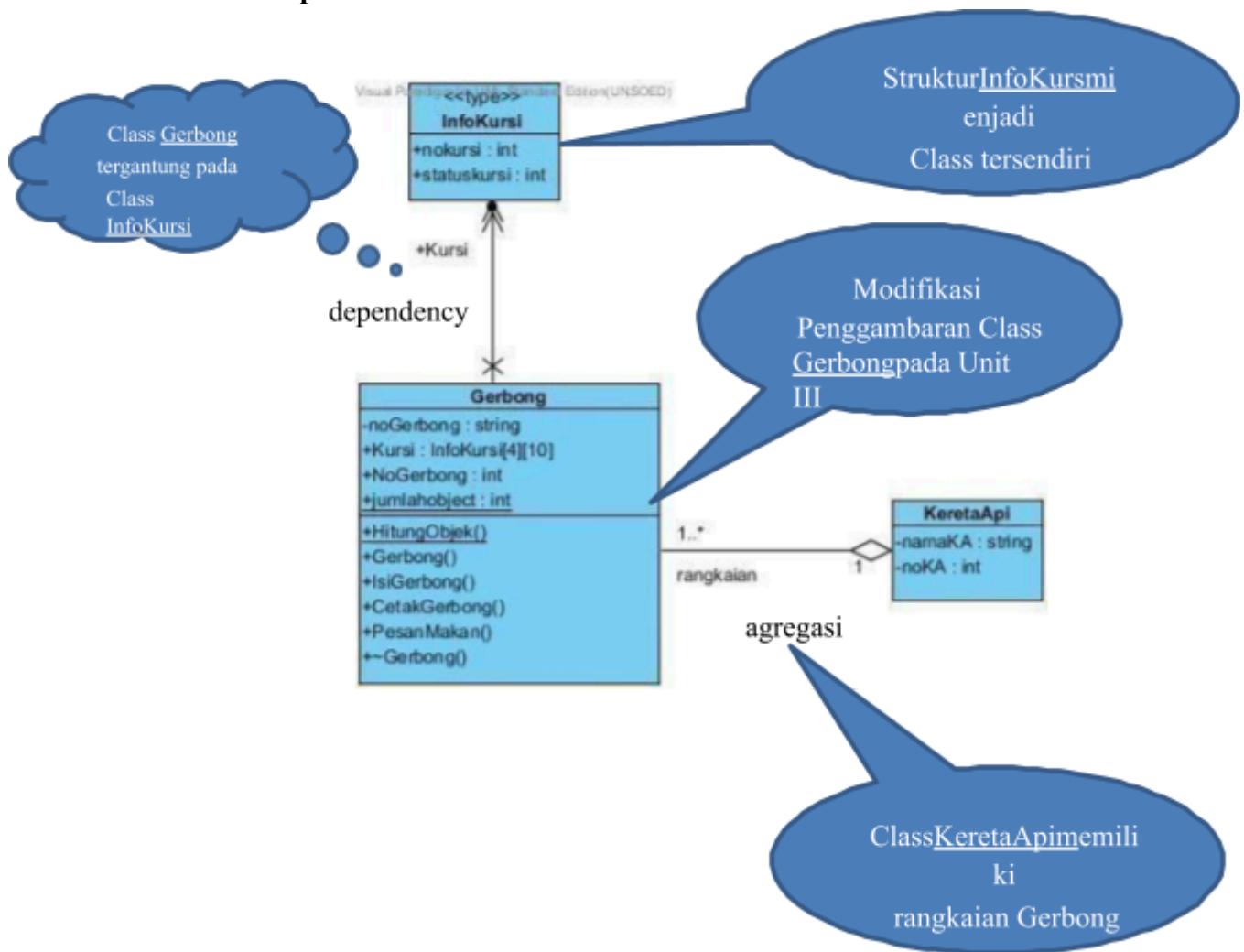
## **UNIT VIII: MODIFIKASI DAN VISUALISASI CLASS GERBONG**

### **8.1 Tujuan Praktikum**

1. Mahasiswa mampu mendefinisikan strukturisasi Class sebagai model untuk pemecahan masalah.
2. Mahasiswa mampu menggunakan Class dan memodifikasi Class dalam berbagai kebutuhan.
3. Mahasiswa mampu memahami hubungan antar Class (Asosiasi, Agregasi, Dependency)
4. Mahasiswa mampu membuat kode sumber untuk mengimplementasikan hubungan Asosiasi, Agregasi, Dependency.
5. Mahasiswa mampu menyusun program yang memuat berbagai hubungan antar Class.

### **8.2 Dasar Teori**

#### **8.2.1 Deskripsi Class Secara Visual**



Gambar 1. Struktur Class Gerbong

### 8.2.1.1 Hubungan Antar Class (Dependency)

Class Gerbong Dependency terhadap Class InfoKursi. Penulisan Programnya sebagai berikut :

*InfoKursi Kursi[4][10]; // Bagian dari atribut di Class Gerbong*

Dependency direpresentasikan dengan garis panah terputus-putus :



### 8.2.1.2 Hubungan Antar Class (Agregasi)

Class KeretaApi agregasi terhadap rangkaian Class Gerbong. Penulisan Programnya sebagai berikut :

```
List<Gerbong> rangkaian; //atau
Collection<Gerbong> rangkaian; // atau
Vector<Gerbong> rangkaian
```

Agregasi direpresentasikan dengan garis yg diujungnya ada bentuk diamond terbuka:

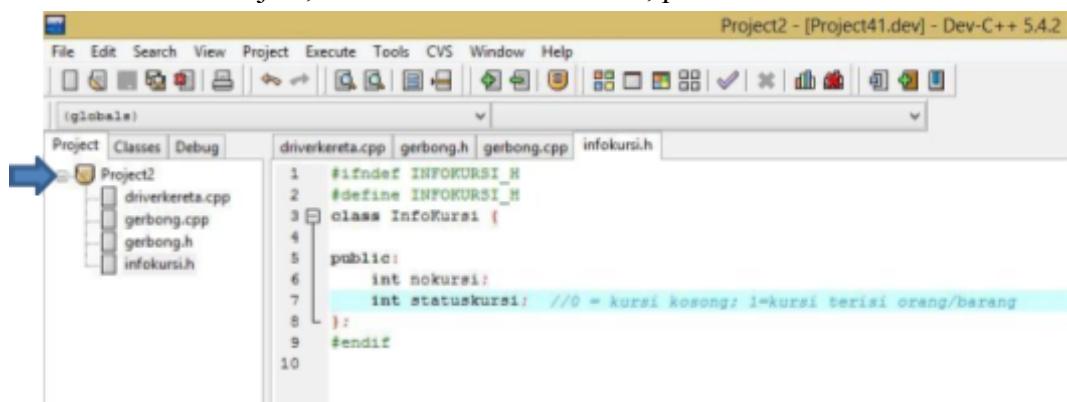


## 8.3 Alat dan Bahan

1. Komputer
2. IDE Dev C++ 5.4.2 lengkap dengan Library STIL C++

## 8.4 Percobaan dan Pengamatan

1. Buatlah project41, salin seluruh file (\*.h) dan (\*.cpp) dari project31.
2. Buatlah Class Baru bernama InfoKursi sebagai Class yang merepresentasikan informasi Kursi seperti di bawah ini. Cara : Arahkan mouse ke nama Project, klik tombol mouse kanan, pilih *New File*.



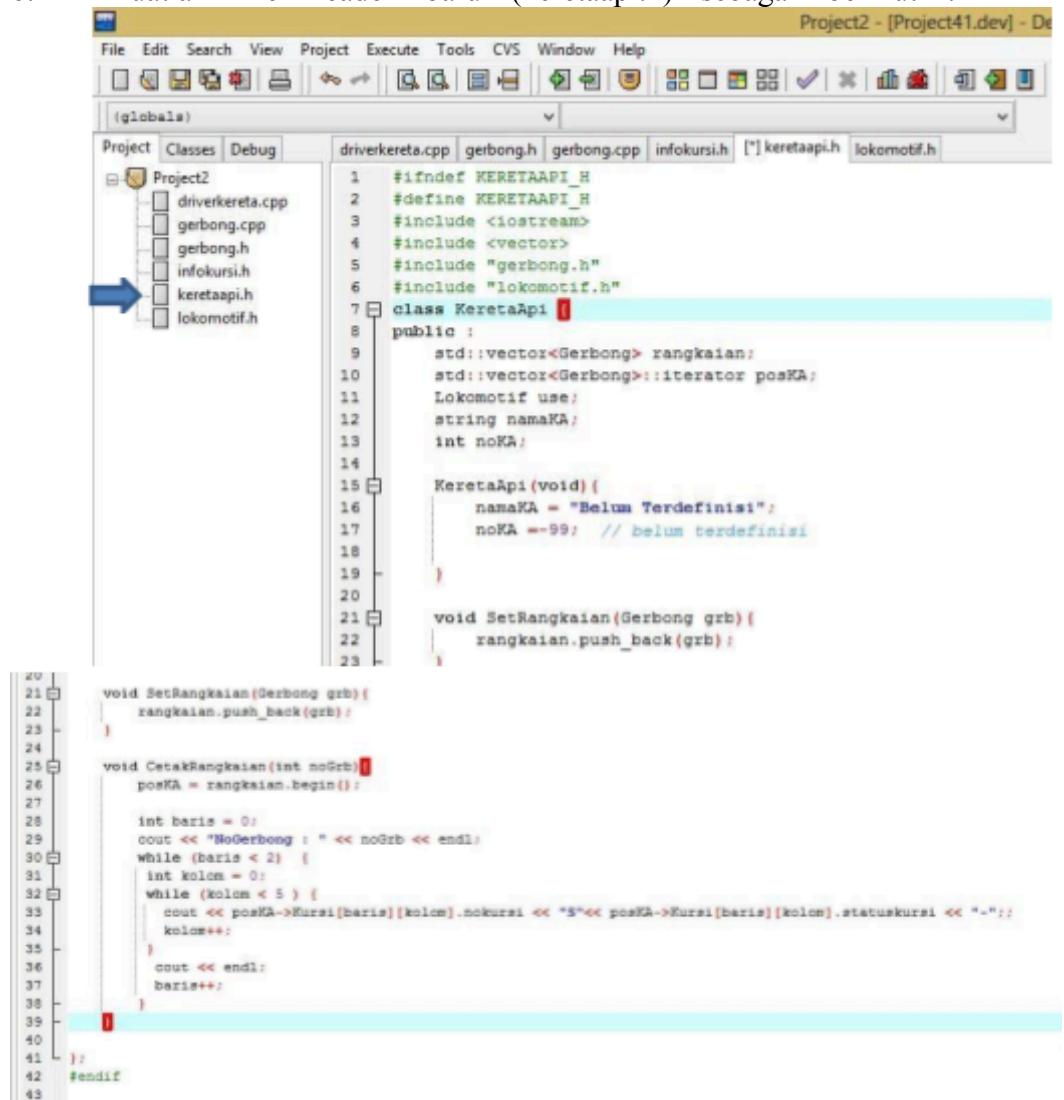
3. Modifikasilah Class Gerbong berikut ini.

The screenshot shows the Dev-C++ IDE interface with the title bar "Project2 - [Project41.dev] - Dev-C++ 5.4.2". The menu bar includes File, Edit, Search, View, Project, Execute, Tools, CVS, Window, and Help. Below the menu is a toolbar with various icons. The left pane shows a project tree for "Project2" containing files: driverkereta.cpp, gerbong.h, gerbong.cpp, and infokursi.h. The right pane displays the source code for "gerbong.cpp". The code defines a class Gerbong with methods like constructor, IsiGerbong, CetakGerbong, PesanMakan, and destructor. A blue speech bubble on the right side of the screen contains the text "Bandingkan dengan Project31".

```
1 #ifndef GERBONG_H
2 #define GERBONG_H
3 #include <iostream>
4 #include "infokursi.h"
5 using namespace std;
6 class Gerbong
7 {
8 public :
9 InfoKursi Kursi[4][10];
10 int NoGerbong;
11 public :
12 static int jumlahobject;
13 static int HitungObjek(void)
14 { return jumlahobject; }
15
16 Gerbong(void);
17 // Constructor Class Gerbong, melakukan inisialisasi no gerbong dengan -99
18 // dan menomori kursi setiap gerbong sejumlah 10 kursi dari no 1 sd 10
19 Gerbong IsiGerbong(int _nogb, InfoKursi* _kursi);
20 // Kondisi Awal (IS) : Gerbong telah terdefinisi
21 // Kondisi Akhir (FS) : Gerbong terisi InfoKursi;
22 void CetakGerbong(void);
23 // Metod untuk melihat nomor gerbong, no kursi beserta statusnya seluruh
24 // gerbong dalam rangkaian Kereta Api
25 void PesanMakan(int nomenu, int nokursi, int nogerbong);
26 ~Gerbong();
27 };
28 #endif
29
```

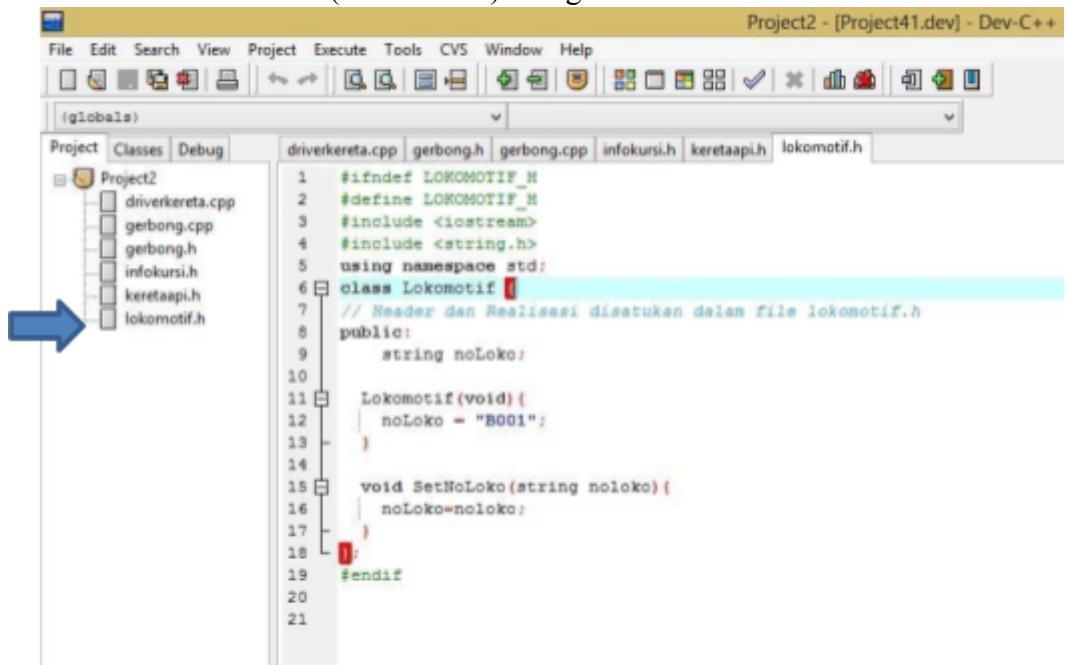
4. Simpanlah seluruh perubahan kode sumber, lakukan kompilasi dan eksekusi. Amati hasil eksekusi program tersebut, apakah ada perbedaan output dengan project31 ?
5. Lakukan Debugging seperti pada Project31. Jangan lupa untuk mengamati beberapa objek yang terbentuk melalui Add Watch. Apakah ada perbedaan dengan project31?

6. Buatlah file header baru (keretaapi.h) sebagai berikut :



```
Project2 - [Project41.dev] - De
File Edit Search View Project Execute Tools CVS Window Help
(globals)
Project Classes Debug
Project2
 driverkereta.cpp gerbong.h gerbong.cpp infokursi.h [*] keretaapi.h lokomotif.h
1 #ifndef KERETAAPI_H
2 #define KERETAAPI_H
3 #include <iostream>
4 #include <vector>
5 #include "gerbong.h"
6 #include "lokomotif.h"
7 class KeretaApi {
8 public :
9 std::vector<Gerbong> rangkaian;
10 std::vector<Gerbong>::iterator posKA;
11 Lokomotif use;
12 string namaKA;
13 int noKA;
14
15 KeretaApi(void) {
16 namaKA = "Belum Terdefinisi";
17 noKA = -99; // belum terdefinisi
18 }
19
20
21 void SetRangkaian(Gerbong grb) {
22 rangkaian.push_back(grb);
23 }
24
25 void CetakRangkaian(int noGrb) {
26 posKA = rangkaian.begin();
27
28 int baris = 0;
29 cout << "NoGerbong : " << noGrb << endl;
30 while (baris < 2) {
31 int kolom = 0;
32 while (kolom < 5) {
33 cout << posKA->kursi[baris][kolom].nokursi << " " << posKA->kursi[baris][kolom].statuskursi << " ";
34 kolom++;
35 }
36 cout << endl;
37 baris++;
38 }
39 }
40
41 }
42 #endif
```

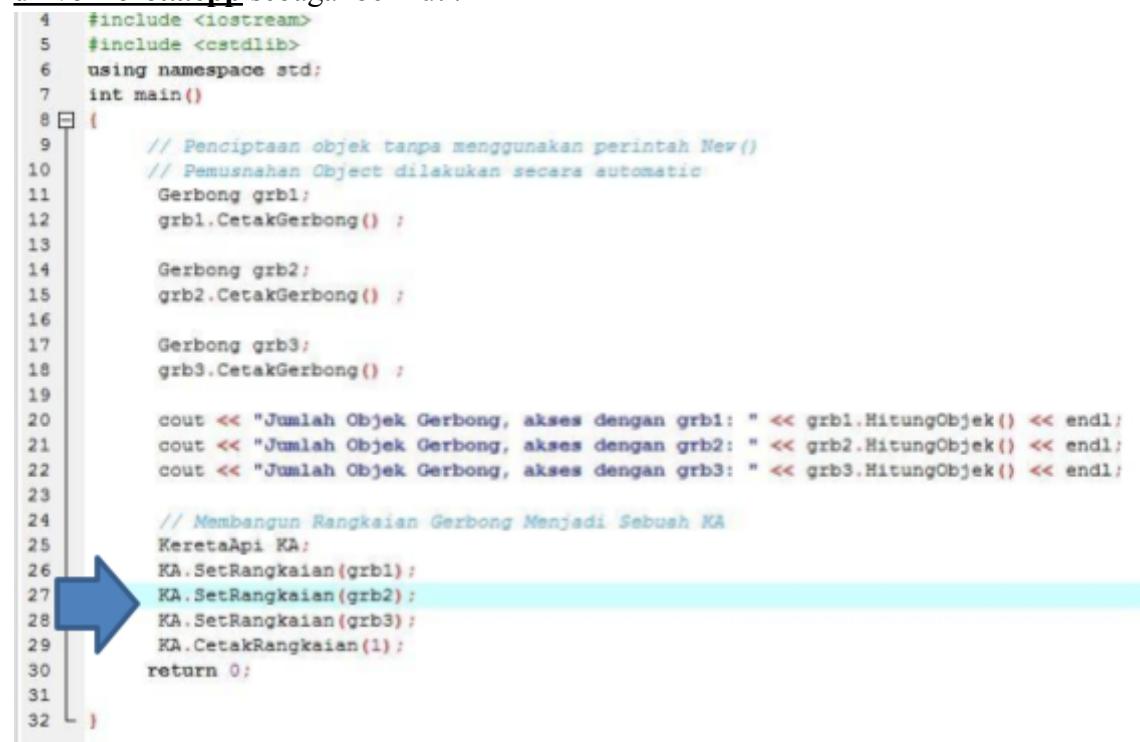
7. Buatlah file header baru (lokomotif.h) sebagai berikut:



The screenshot shows the Dev-C++ IDE interface with the title bar "Project2 - [Project41.dev] - Dev-C++". The menu bar includes File, Edit, Search, View, Project, Execute, Tools, CVS, Window, and Help. The toolbar has various icons for file operations. The global search bar contains "(globals)". The project navigation bar shows "Project", "Classes", and "Debug", with "Project" selected. The project tree under "Project2" lists files: driverkereta.cpp, gerbong.h, gerbong.cpp, infokursi.h, keretaapi.h, and lokomotif.h. A blue arrow points from the left towards the code editor. The code editor displays the following C++ code:

```
1 #ifndef LOKOMOTIF_H
2 #define LOKOMOTIF_H
3 #include <iostream>
4 #include <string.h>
5 using namespace std;
6 class Lokomotif {
7 // Header dan Realisasi disatukan dalam file lokomotif.h
8 public:
9 string noLoko;
10 Lokomotif(void) {
11 noLoko = "B001";
12 }
13 void SetNoLoko(string noloko) {
14 noLoko=noloko;
15 }
16};
17
18#endif
```

8. Tambahkan bagian program untuk menciptakan keretaapi pada driverkereta.cpp sebagai berikut :



The screenshot shows the continuation of the Dev-C++ IDE interface with the same title bar and menu. A blue arrow points from the left towards the code editor. The code editor displays the following C++ code:

```
4 #include <iostream>
5 #include <cstdlib>
6 using namespace std;
7 int main()
8 {
9 // Penciptaan objek tanpa menggunakan perintah New()
10 // Pemusnahan Object dilakukan secara automatic
11 Gerbong grb1;
12 grb1.CetakGerbong();
13
14 Gerbong grb2;
15 grb2.CetakGerbong();
16
17 Gerbong grb3;
18 grb3.CetakGerbong();
19
20 cout << "Jumlah Objek Gerbong, akses dengan grb1: " << grb1.HitungObjek() << endl;
21 cout << "Jumlah Objek Gerbong, akses dengan grb2: " << grb2.HitungObjek() << endl;
22 cout << "Jumlah Objek Gerbong, akses dengan grb3: " << grb3.HitungObjek() << endl;
23
24 // Membangun Rangkaian Gerbong Menjadi Sebuah KA
25 KeretaApi KA;
26 KA.SetRangkaian(grb1);
27 KA.SetRangkaian(grb2);
28 KA.SetRangkaian(grb3);
29 KA.CetakRangkaian(1);
30 return 0;
31
32 }
```

9. Lakukan Kompilasi, Eksekusi dan Pengamatan menggunakan Add Watch.
10. Amatilah penghapusan Objek

### **8.5 Pertanyaan**

1. Tunjukkanlah kode sumber yang bertugas merangkaikan grb1, grb2, grb3 menjadi rangkaian Kereta Api!.
2. Jelaskan perbedaan method CetakGerbong() pada gerbong.cpp dengan CetakRangkaian() di keretaapi.h!
3. Hitunglah ada berapa objek Gerbong yang terbentuk pada program di atas! Sebutkan nama-nama objeknya!
4. Hitunglah ada berapa objek Kereta Api, sebutkan namanya!

## UNIT IX : MANIPULASI STRUKTUR DATA KERETA API

### 9.1 Tujuan Praktikum

1. Mahasiswa mampu menambahkan method baru pada struktur Class Gerbong, Class KeretaApi, Class Lokomotif.
2. Mahasiswa mampu menggunakan method SearchingByBoolean dan SearchingByAddress pada struktur Class KeretaApi.

### 9.2 Dasar Teori

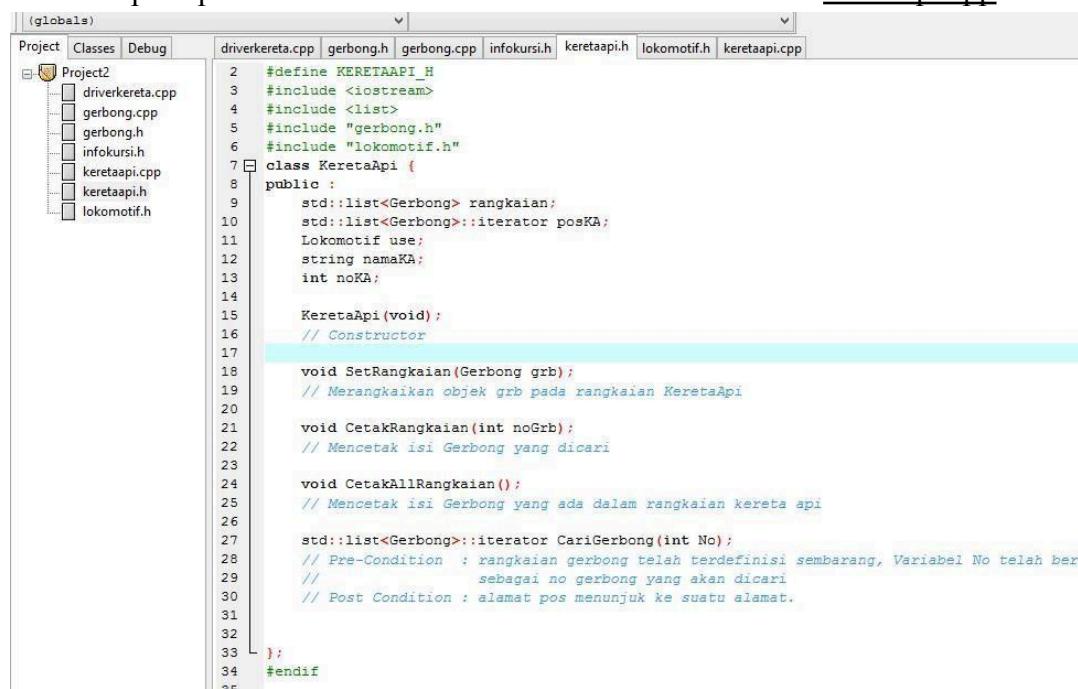
[Reference]

### 9.3 Alat dan Bahan

1. Komputer
2. IDE Dev C++ 5.4.2 lengkap dengan Library STL C++

### 9.4 Percobaan dan Pengamatan

1. Buatlah project51, salin seluruh file (\*.h) dan (\*.cpp) dari project41.
2. Pisahkanlah header dan realisasi dari Class KeretaApi yang terbagi 2 bagian yaitu keretaapi.h dan keretaapi.cpp. Sesuaikan isi file keretaapi.h seperti pada source code di bawah ini. Buatlah file baru keretaapi.cpp.



The screenshot shows the Dev C++ IDE interface. The top menu bar includes 'globals', 'Project', 'Classes', and 'Debug'. Below the menu is a toolbar with icons for file operations. The main area shows a project named 'Project2' with files: driverkereta.cpp, gerbong.h, gerbong.cpp, infokursi.h, keretaapi.h, lokomotif.h, and keretaapi.cpp. The 'keretaapi.cpp' file is open and displays the following C++ code:

```
#define KERETAAPI_H
#include <iostream>
#include <list>
#include "gerbong.h"
#include "lokomotif.h"

class KeretaApi {
public :
 std::list<Gerbong> rangkaian;
 std::list<Gerbong>::iterator posKA;
 Lokomotif use;
 string namaKA;
 int noKA;
}

KeretaApi(void);
// Constructor

void SetRangkaian(Gerbong grb);
// Merangkaikan objek grb pada rangkaian KeretaApi

void CetakRangkaian(int noGrb);
// Mencetak isi Gerbong yang dicari

void CetakAllRangkaian();
// Mencetak isi Gerbong yang ada dalam rangkaian kereta api

std::list<Gerbong>::iterator CariGerbong(int No);
// Pre-Condition : rangkaian gerbong telah terdefinisi sembarang, Variabel No telah ber
// sebagai no gerbong yang akan dicari
// Post Condition : alamat pos menunjuk ke suatu alamat.

};
```

```

1 #include "KeretaApi.h"
2
3 void KeretaApi::KeretaApi(void){
4 namaKA = "Belum Terdefinisi";
5 noKA = -99; // belum terdefinisi
6 }
7
8 void KeretaApi::SetRangkaian(Gerbong grb){
9 rangkaian.push_back(grb);
10}
11
12 void KeretaApi::CetakRangkaian(int noGrb){
13 posKA = CariGerbong(noGrb);
14 if (posKA == rangkaian.end()){
15 cout << "Gerbong No :" << noGrb << "Tidak Ditemukan" << endl;
16 }
17 else{
18 int baris = 0;
19 cout << "NoGerbong : " << noGrb << endl;
20 while (baris < 2) {
21 int kolom = 0;
22 while (kolom < 5) {
23 cout << posKA->Kursi[baris][kolom].nokursi << "S" << posKA->Kursi[baris][kolom].statuskursi << "-";
24 kolom++;
25 }
26 cout << endl;
27 baris++;
28 }
29 }
30 }
31

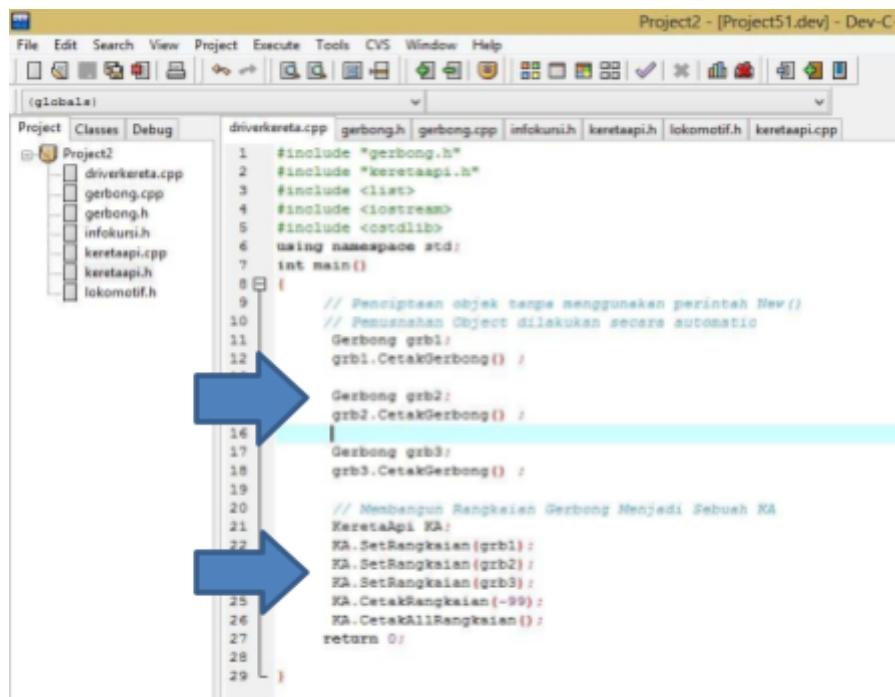
32 void KeretaApi::CetakAllRangkaian()
33 // Mencetak isi Gerbong yang ada dalam rangkaian kereta api
34 {
35 posKA = rangkaian.begin();
36 for (posKA; posKA != rangkaian.end(); posKA++)
37 {
38 int baris = 0;
39 cout << "NoGerbong : " << posKA->NoGerbong << endl;
40 while (baris < 2) {
41 int kolom = 0;
42 while (kolom < 5) {
43 cout << posKA->Kursi[baris][kolom].nokursi << "S" << posKA->Kursi[baris][kolom].statuskursi << "-";
44 kolom++;
45 }
46 cout << endl;
47 baris++;
48 }
49 }
50 }
51

52 std::list<Gerbong>::iterator KeretaApi::CariGerbong(int No)
53 {
54 int found = false;
55 posKA=rangkaian.begin();
56 while ((!found) && (posKA!=rangkaian.end())){
57
58 if (posKA->NoGerbong == No){
59 found = true;
60 return posKA; // Ketemu
61 }
62 else {
63 posKA++;
64 }
65 }
66 if (!found)
67 {
68 return posKA; // tidak ketemu
69 }
70 }
71

```

Sel: 0 Lines: 71 Length: 1929 Insert Done parsing

3. Uji Coba Class KeretaApi yang telah dimodifikasi. Buka file driverkereta.cpp ! Tambahkan statement berikut di bawah ini. Perhatikan, ada penambahan method baru yaitu CetakAllRangkaian() dan CariGerbong().



```
Project2 - [Project51.dev] - Dev-C++
File Edit Search View Project Execute Tools CVS Window Help
(global.a)
Project Classes Debug
Project2
 driverkereta.cpp
 gerbong.cpp
 gerbong.h
 infokursi.h
 keretaapi.cpp
 keretaapi.h
 lokomotif.h
driverkereta.cpp gerbong.h gerbong.cpp infokursi.h keretaapi.h lokomotif.h keretaapi.cpp
1 #include "gerbong.h"
2 #include "Keretaapi.h"
3 #include <list>
4 #include <iostream>
5 #include <cstdlib>
6 using namespace std;
7 int main()
8 {
9 // Penciptaan objek tanpa menggunakan perintah New()
10 // Pemusnahan Object dilakukan secara automatic
11 Gerbong grb1;
12 grb1.CetakGerbong();
13
14 Gerbong grb2;
15 grb2.CetakGerbong();
16
17 Gerbong grb3;
18 grb3.CetakGerbong();
19
20 // Membangun Rangkaian Gerbong Menjadi Sebuah KA
21 KeretaApi KA;
22 KA.SetRangkaian(grb1);
23 KA.SetRangkaian(grb2);
24 KA.SetRangkaian(grb3);
25 KA.CetakRangkaian(-99);
26 KA.CetakAllRangkaian();
27
28
29 }
```

4. Lakukan kompilasi, eksekusi dan debugging. Output program di atas adalah sebagai berikut :

```
NoGerbong : -99
1S0-3S0-5S0-7S0-9S0-
2S0-4S0-6S0-8S0-10S0-
NoGerbong : -99
1S0-3S0-5S0-7S0-9S0-
2S0-4S0-6S0-8S0-10S0-
NoGerbong : -99
1S0-3S0-5S0-7S0-9S0-
2S0-4S0-6S0-8S0-10S0-
Hapus Object
Hapus Object
Hapus Object ←
NoGerbong : -99
1S0-3S0-5S0-7S0-9S0-
2S0-4S0-6S0-8S0-10S0-
NoGerbong : -99
1S0-3S0-5S0-7S0-9S0-
2S0-4S0-6S0-8S0-10S0-
NoGerbong : -99
1S0-3S0-5S0-7S0-9S0-
2S0-4S0-6S0-8S0-10S0-
Hapus Object
Hapus Object
Hapus Object ←
Hapus Object
Hapus Object
Hapus Object
Hapus Object
Hapus Object
```

Process exited with return value 0  
Press any key to continue . . .

## 9.5 Pertanyaan

1. Jelaskan tentang penghapusan Objek yang telah dibuat pada program di atas. Output program menunjukkan ada 9x penghapusan objek.
2. Jelaskan perbedaan method CetakRangkaian() dengan CetakALLRangkaian()
3. Termasuk pada kategori mana method CariGerbong() ? SearchingByBoolean atau SearchingByAddress ?