Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2025/2026

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

NIM	**71251172**
Nama Lengkap	**Ang, Joshua Alexander Hartono**
Minggu ke / Materi	**02 / Variable, Expression dan Statements**

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2026

BAGIAN 1: MATERI MINGGU INI (40%)

Values dan type

Value adalah komponen dasar dalam program, contohnya angka dan huruf seperti 1, 2, ‘a’, ‘z’ atau “Hello World”. Setiap value punya tipe data yang berbeda. Misalnya 2 memiliki tipe integer, sedangkan “Hello World” memiliki tipe string. Untuk string dapat dikenali dengan penggunaan tanda petik. Di Python, perintah print() digunakan untuk menampilkan jenis value.

```
>>> print(4)
4
>>> print(10.876)
10.876
```

Gambar 2.1: Contoh penggunaan perintah print untuk menampilkan tipe value. *Sumber:* https://sl1nk.com/TbY8S

Selain string terdapat juga value seperti integer (bilangan bulat), float (bilangan pecahan), character (huruf), boolean (True/False). Setiap value pasti memiliki tipe data. Untuk mengetahui tipe data tersebut. Python menyediakan fungsi bawaan type(). Python juga mendukung beberapa tipe data angka seperti int untuk bilangan bulat, float untuk bilangan desimal, complex untuk bilangan kompleks. Satu hal yang perlu diingat, Python tidak menggunakan tanda koma sebagai pemisah ribuan.

```
>>> x=5
>>> print(x, "tipenya adalah ", type(x))
5 tipenya adalah <class int'>
```

Gambar 2.2: Contoh pengecekan tipe data menggunakan fungsi type(). *Sumber:* https://sl1nk.com/TbY8S

Jika menuliskan 1,000,000, Python akan menggangapnya sebagai beberapa parameter yang terpisah dalam fungsi ptint(), bukan sebagai satu bilangan besar.

```
>>> print(1,000,000)
1,0,0
```

Gambar 2.3: Contoh output perintah print dengan penggunaan tanda koma sebagai pemisah parameter. *Sumber:* https://sl1nk.com/TbY8S

Variabel

Variabel merupakan salah satu fitur penting dalam bahasa pemrograman Python. Variabel adalah lokasi di memori yang digunakan untuk menyimpan suatu nilai. Nilai yang disimpan dalam variabel dapat digunakan kembali dan dapat berubah selama program dijalankan. Pada saat sebuah variabel dibuat, Python akan menyediakan ruang di memori untuk menyimpan data.

```
>>> pesan = 'selamat pagi, mari belajar python'
>>> n = 17
>>> pi = 3.1415926535897931
```

Gambar 2.4: Contoh pendefinisian variabel dalam Python. *Sumber:* https://sl1nk.com/TbY8S

Berdasarkan gambar di atas, variabel pesan menyimpan data bertipe string, variabel n menyimpan bilangan bulat (integer), dan variabel pi menyimpan bilangan pecahan (float). Python bersifat dinamis, sehingga tipe data variabel tidak perlu ditentukan secara langsung.

```
>>> print(n)
17
>>> print(ipk)
3.29
```

Gambar 2.5: Menampilkan nilai variabel menggunakan print. *Sumber:* https://sl1nk.com/TbY8S

Nama Variabel dan Keyword

Pemberian nama variabel dalam Python harus mengikuti beberapa aturan. Nama variabel boleh diawali dengan huruf atau garis bawah (_), dan karakter selanjutnya dapat berupa huruf, angka, atau garis bawah. Contoh: _nama, n2, dan nilai1.Python bersifat case sensitive, sehingga huruf besar dan kecil pada nama variabel dianggap berbeda, misal variabel_Ku dan variabek_ku. Selain itu, nama variabel tidak boleh menggunakan kata kunci (keyword) yang sudah tersedian di

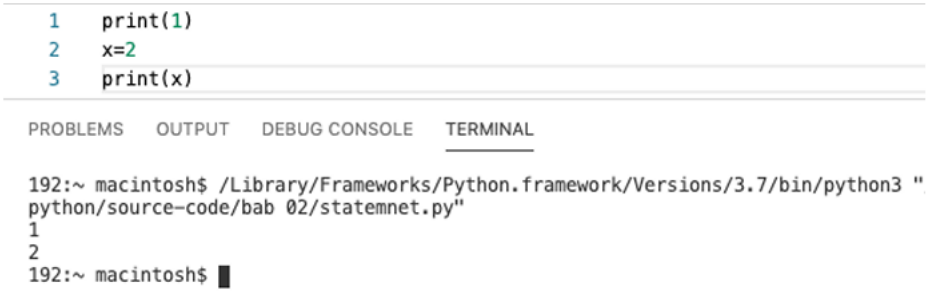
Python, seperti if, for, while, True, dan False karena kata-kata tersebut memiliki fungsi khusus di Python.

and	**del**	**from**	**None**	**True**
as	**elif**	**global**	**nonlocaly**	**try**
assert	**else**	**if**	**not**	**while**
break	**except**	**import**	**or**	**width**
class	**False**	**in**	**pass**	**yield**
continue	**finally**	**is**	**raise**	**async**
def	**for**	**lamda**	**return**	**wait**

Gambar 2.6: 35 keyword yang tidak boleh digunakan untuk memberi nama variabel. *Sumber:* https://sl1nk.com/TbY8S

Statements

Statements merupakan perintah dalam Python yang dapat dijalankan oleh interpreter. Contoh statement yang sering digunakan adalah print untuk menampilkan data dan assignment untuk memberikan nilai ke variabel. Saat Python digunakan dalam mode interaktif, setiap statement yang diketik akan langsung dieksekusi dan hasilnya langsung ditampilkan. Hal ini berbeda dengan script mode, di mana kumpulan statement di tulis dalam satu file dan dijalankan secara berurutan dari atas ke bawah.



Gambar 2.7: Contoh statement dan outputnya. *Sumber:* https://sl1nk.com/TbY8S

Operator dan Operand

Operator adalah simbol yang digunakan dalam Python untuk melakukan operasi aritmatika maupun logika. Nilai yang dikenai operasi disebut operand. Sebagai contoh pada operasi 2+3,

tanda + merupakan operator penjumlahan, sedangkan angka 2 dan 3 disebut operand. Python meyediakan operator aritmatika yang digunakan untuk melakukan operasi matematika seperti penjumlahan, pengurangan, perkalian, pembagian, dan sebagainya.

Operator	Nama dan Fungsi	Contoh
+	Penjumlahan, menjumlahkan 2 buah operand	x + y
–	Pengurangan, mengurangkan 2 buah operand	x - y
*	Perkalian, mengalikan 2 buah operand	x * y
/	Pembagian, membagi 2 buah operand	x / y
**	Pemangkatan, memangkatkan bilangan	x **y

Gambar 2.8: Operator aritmatika pada Python. *Sumber:* https://sl1nk.com/TbY8S

Expressions

Expressions merupakan gabungan nilai antara values. variabel, dan operator. Sebuah value tunggal dapat disebut sebagai expression. Ketika expression dijalankan pada mode interaktif, interpreter Python akan langsung melakukan evaluasi dan menampilkan hasilnya.

```
>>> 1 + 1
2
>>> 3 + 2
5
```

Gambar 2.9: Expression dalam mode interaktif. *Sumber:* https://sl1nk.com/TbY8S

Urutan Operasi

Urutan operasi digunakan Ketika sebuah expression memiliki lebih dari satu operator. Python menentukan hasil perhitungan berdasarkan prioritas operator, yang mengikuti aturan matematika umum yang dikenal dengan PEMDAS (Parentheses, Exponentiation, Multiplication and Division, Operator).

Parentheses atau tanda kurung menjadi prioritas tertinggi dan digunakan untuk menentukan urutan perhitungan yang diinginkan, misalnya 2 * (3-1) menghasilkan 4. Setelah tanda kurung ada Exponentiation atau pemangkatan, contohnya 2**1 +1 menghasilkan 3 karena operasi pemangkatan dilakukan terlebih dahulu. Selanjutnya ada pembagian dan perkalian yang bernilai

setara dan dikerjakan sebelum penjumlahan dan pengurangan, sehingga 2 * 3-1 akan menghasilkan 5. Terahkir ada operator, operator memiliki prioritas yang sama dibaca dari kiri ke kanan, contohnya 5 - 3 -1 hasilnya 1 bukan 3.

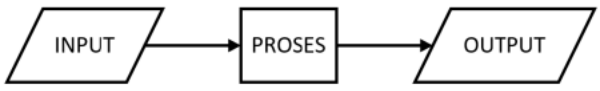
Operator Modulus dan String

Operator modulus digunakan untuk mendapatkan sisa hasil bagi dari pembagian dua bilangan dan hanya berlaku untuk tipe data integer. Dalam Python, operator modulus dilambangkan dengan tanda persen (%). Sebagai contoh pembagian 7 dengan 3 menggunakan operator pembagian bulat akan menghasilkan 2, sedangkan penggunaan operator modulus 7 % 3 menghasilkan sisa pembagian yaitu 1. Operator modulus sering digunakan untuk mengecek apakah suatu bilangan dapat dibagi oleh bilangan lain, misalnya jika hasil x % y adalah 0 maka x habis dibagi y. Selain itu, operator modulus juga dapat digunakan untuk mengambil digit terahkir dari suatu bilangan seperti x % 10 untuk digit paling kanan.

Pada tipe data string, operator + tidak dapat digunakan untuk penjumlahan, melainkan untuk menggabungkan (concatenation) dua string. Contohnya, penggabungan string ‘100’ dan ‘150’ akan mrnghasilkan ‘100150’. Selain itu, operator * juga dapat digunakan pada string untuk mengulang isi string sebanyak bilangan integer tertentu. Misalnya, string ‘Test’ yang dikalikan dengan angka 3 akan menghasilkan output ‘Test Test Test’.

Menangani Input dari Pengguna

Sebuah program pada umumnya memiliki alur kerja Input – Proses – Output.



Gambar 2.10: Input – Proses – Output. *Sumber:* https://sl1nk.com/TbY8S

Input merupakan data atau masukan yang dibutuhkan agar program dapat berjalan. Proses adalah langkah-langkah yang dilakukan oleh program untuk mengolah input dan menyelesaikan suatu masalah. Output adalah hasil yang diperoleh setelah proses berhasil dijalankan. Contoh penerapan konsep ini dapat dilihat pada pengambilan uang melalui mesin ATM. Kartu ATM, PIN,

serta nominal uang yang dimasukkan merupakan input, sedangkan prosesnya meliputi pengecekan saldo dan status kartu oleh sistem bank. Output dari proses tersebut adalah uang yang dikeluarkan oleh mesin ATM serta berkurangnya saldo rekening pengguna.

Python juga dapat menerima input dari pengguna melalui keyboard. Untuk mengambil input tersebur, Python menyediakan fungsi bawaan input(). Ketika fungsi ini dipanggil, program akan berhenti sementara dan menunggu pengguna memasukkan data, kemudian melanjutkan program setelah tombol Enter ditekan. Data yang dimasukkan oleh pengguna akan disimpan dalam bentuk string. Agar lebih jelas bagi pengguna, biasanya ditampilkan pesan atau prompt sebelum input dimasukkan. Prompt tersebut dapat disertakan langsung di dalam fungsi input(). Tanda \n digunakan untuk membuat baris baru sehingga input pengguna muncul di bawah teks prompt.

Jika input yang dimasukkan diharapkan berupa bilangan, maka data tersebut perlu dikonversi ke tipe data integer menggunakan fungsi int(). Namun, perlu diperhatikan bahwa konversi ini akan menyebabkan eror apabila memasukkan data selain angka.

Komentar

Komentar dalam Python digunakan untuk memberikan penjelasan kode program agar lebih mudah dipahami oleh programmer. Komentar ditandai dengan tanda pagar (#) dan tidak akan diproses oleh integer Python, sehingga tidak mempengaruhi hasil eksekusi program. Komentar dapat diletakkan pada baris tersendiri atau di ahkir baris setelah perintah program. Python tidak memiliki komentar khusus untuk multibaris, sehingga setiapa baris komentar harus diawali dengan tanda pagar secara terpisah.

```
# Komentar pertama
print("hai dunia!") # Komentar kedua
```

Gambar 2.11: Contoh penggunaan komentar. *Sumber:* https://sl1nk.com/TbY8S

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

Source code:

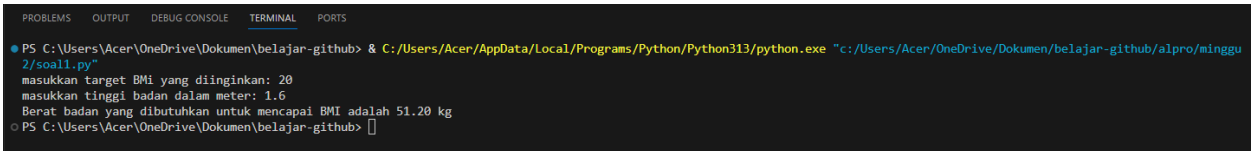
```
BMI = float(input("masukkan target BMi yang diinginkan: "))
tinggi = float(input("masukkan tinggi badan dalam meter: "))
berat = BMI * (tinggi**2)

print(f"Berat badan yang dibutuhkan untuk mencapai BMI adalah {berat:.2f} kg")
```

Penjelasan:

Pada baris pertama, program meminta pengguna memasukkan target dari BMI yang ingin dicapai. **input()** berfungsi untuk mengambil data dari pengguna lalu diproses dengan **float()** sehingga nilai yang dimasukkan dapat diproses menjadi angka decimal. Nilai tersebut kemudian akan disimpan ke variabel **BMI**. Pada baris kedua, program kembali meminta pengguna memasukkan data, yaitu tinggi badan dalam satuan meter. Sama seperti sebelumnya, **input()** digunakan untuk menerima data dari pengguna, lalu nilai tersebut diubah menjadi angka desimal **float()** agar bisa digunakan dalam perhitungan matematika. Setelah diubah, nilai tinggi kemudian disimpan ke variabel tinggi.

Pada baris ketiga, program melakukan proses perhitungan untuk menentukan berat badan yang dibutuhkan. Rumus yang digunakan adalah **berat = BMI * (tinggi**2)**. Tanda **2 berarti nilai tinggi dipangkatkan dua. Hasil dari perhitungan ini kemudian akan disimpan ke dalam variabel **berat**. Pada baris terahkir, program menampilkan hasil perhitungan ke layer menggunakan **print()**. Di dalamnya terdapat format **{berat:.2f}** yang berfungsi untuk menampilkan nilai berat denga dua angka dibelakang koma agar nilai menjadi lebih mudah dibaca.

Output:



SOAL 2

Source code:

```
print("f(x) = 2x^3 + 2x + 15/x")
nilai = int(input("masukkan nilai x: "))

hasil = 2 * nilai**3 + 2 * nilai + 15 / nilai

print(f"hasil dari f({nilai}) adalah {hasil:.2f}")
```

Penjelasan:

Pada baris pertama, program menampilkan tulisan **f(x) = 2x^3 + 2x + 15/x** menggunakan fungsi **print()**. Baris ini hanya bertujuan untuk memberi tahu pengguna bentuk fungsi yang akan dihitung nanti. Selanjutnya, program meminta pegguna memasukkan nilai untuk variabel **x**.fungsi **input()** digunakan untuk menerima data dari pengguna, lalu nilai tersebut diubah menjadi tipe data bilangan bulat menggunakan **int()**. Setelah itu, nilai yang sudah dikonversi disimpan ke dalam variabel **nilai**.

Pada baris keempat, program mulai melakukan perhitungan sesuai dengan fungsi yang sudah ditampilkan. Perhitungan dilakukan dengan rumus **f(x) = 2x^3 + 2x + 15/x**. bagian **nilai **3** beratti nilai dipangkat 3. Kemudian hasilnya dikalikan 2, ditambah 2 kali nilai, lalu ditambah 15 bagi nilai. Hasil ahkir dari seluruh operasi tersebut disimpanke dalam variabel **hasil**. Terahkir, program menampilkan hasil perhitungan ke layar menggunakan f-string. Bagian **{hasil:.2f}** bergfungsi untuk menampilkan nilai hasil dengan dua angka dibelakang koma.

Output:

```
PS C:\Users\Acer\OneDrive\Dokumen\belajar-github> & C:/Users/Acer/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Acer/OneDrive/Dokumen/belajar-github/alpro/minggu 2/soal2.py"
f(x) = 2x^3 + 2x + 15/x
masukkan nilai x: 3
hasil dari f(3) adalah 65.00
```

SOAL 3

Source code:

```
gaji = int(input("Masukkan gaji yang diinginkan: "))
jam_kerja = int(input("Masukkan jumlah jam kerja perminggu: "))

pendapatan1 = gaji * jam_kerja * 5 #pendapatan sebelum kena pajak
pendapatan2 = pendapatan1 - pendapatan1 * 0.14 #pendapatan setelah kena pajak

pengeluaran1 = pendapatan2 * 0.1 #pengeluaran untuk beli pakaian dan aksesoris
pengeluaran2 = pendapatan2 * 0.01 #pengeluaran untuk beli alat tulis

sisa_uang = pendapatan2 - pengeluaran1 - pengeluaran2

pengeluaran3 = sisa_uang * 0.25 #pengeluaran untuk sedekah
sedekah1 = pengeluaran3 * 0.30 #sedekah anak yatim
sedekah2 = pengeluaran3 - sedekah1 #sedekah kaum dhuafa

print(f"1. Pendapatan sebelum pajak: Rp.{pendapatan1:,}")
print(f"2. Pendapatan setelah pajak: Rp.{pendapatan2:,.0f}")
print(f"3. Pengeluaran untuk beli pakaian dan aksesoris: Rp.{pengeluaran1:,.0f}")
print(f"4. Pengeluaran untuk beli alat tulis: Rp.{pengeluaran2:,.0f}")
print(f"5. Pengeluaran untuk sedekah: Rp.{pengeluaran3:,.0f}")
print(f"6. Sedekah untuk anak yatim: Rp.{sedekah1:,.0f}")
print(f"7. Sedekah untuk kaum dhuafa: Rp.{sedekah2:,.0f}")
```

Penjelasan:

Program diawali dengan meminta pengguna untuk memasukkan nilai gaji yang diinginkan dan jumlah jam kerja per minggu. Data yang dimasukkan melalui **input()** kemudian diubah menjadi bilangan bulat menggunakan **int()** agar dapat dihitung. Nilai tersebut disimpan dalam variabel **gaji** dan **jam_kerja**. Setelah itu, program menghitung total pendapatan sebelum pajak disimpan dalam **pendapatan1**. Perhitungan dilakukan dengan mengalikan gaji per jam, jumlah jam perminggu, dan 5 minggu kerja. Kemudian dihitung pendapatan setelah pajak sebesar 14% (0.14) yang disimpan dalam **pendapatan2**. Pajak dihitung dari total pendapatan sebelum pajak lalu dikurangi **pendapatan1 * 0.14**.

Setelah pendapatan yang dikenakan pajak, program menghitung beberapa pengeluaran. Pengeluaran pertama disimpan dalam variabel **pengeluaran1**, yaitu sebesar 10% (0.10) dari pendapatan setelan pajak untuk membeli pakaian dan aksesoris. Selanjutnya, program menghitung **pengeluaran2**, yaitu 1% dari pendapatan setelah pajak untuk membeli alat tulis. Kedua pengeluaran tersebut kemudian dikurangkan dengan pendapatan setelah pajak dengan rumus **pendapatan2 - pengeluaran1 - pengeluaran2**. Nilai tersbut lalu disimpan ke variabel **sisa_uang**.

Dari sisa_uang, program melakukan perhitungan untuk pengeluaran sedekah. Perhitungan menggunakan rumus **pengeluaran3 = sisa_uang * 0.25** yang berarti 25% dari sisa_uang dialokasikakn untuk sedekah dan disimpan ke **pengeluaran3**. Kemudian jumlah sedekah tersebut dibagi menjadi 2 bagian menggunakan rumus **sedekah1 = pengeluaran3 * 0.30**, sehingga 30% diberikan kepada anak yatim dan disimpan dalam variabel sedekah1. Sisa dari total sedekah dihitung dengan rumus **sedekah2 = pengeluaran3 - sedekah1**, lalu diberikan kepada kaum dhuafa dan disimpan dalam variabel **sedekah2**.

Pada bagian ahkir, program menampilakan seluruh hasil perhitungan dengan menggunakan fungsi print(). Format angka menggunakan tanda koma sebagai pemisah ribuan agar lebih mudah dibaca.

Output: