

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
PERULANGAN
TUGAS ALPRO PEKAN 5

Disusun Oleh :

KINAYA NOVRYA MANDA

(2511531016)

Kelas B Informatika

Dosen Pengampu :

DR. WAHYUDI, S.T, M.T

Asisten Praktikum :

MUHAMMAD ZAKI AL HAFIZ



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis sampaikan kehadirat Allah SWT. Salawat dan salam disampaikan kepada Nabi Muhammad SAW. Karena thaufik dan hidayah-Nya, laporan praktikum Java ini dapat diselesaikan dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban dari kegiatan praktikum yang telah dilaksanakan, sekaligus sebagai sarana untuk memperdalam pemahaman mengenai konsep dasar serta penerapan bahasa pemrograman Java.

Melalui praktikum ini, penulis memperoleh pengalaman langsung dalam memahami struktur perulangan yang memungkinkan program membuat keputusan dan menjalankan blok kode yang berbeda berdasarkan kondisi tertentu menggunakan Java. Diharapkan laporan ini dapat memberikan gambaran yang jelas mengenai materi yang telah dipelajari serta hasil dari percobaan yang dilakukan selama praktikum berlangsung.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, baik dari segi isi maupun penyajiannya. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa yang akan datang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, asisten laboratorium, serta semua pihak yang telah memberikan bimbingan, arahan, dan dukungan sehingga laporan praktikum ini dapat tersusun. Semoga laporan ini dapat bermanfaat bagi pembaca, khususnya dalam memahami dasar-dasar pemrograman Java.

Padang, Oktober 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
DAFTAR LAMPIRAN	iii
BAB 1 PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Tujuan Praktikum.....	2
1.3 Manfaat Praktikum.....	2
BAB 2 PEMBAHASAN	
2.1 Kode Program.....	3
2.1.1 Uraian Kode Program	4
2.1.2 Langkah Kerja	5
2.1.3 Analisis Hasil.....	8
2.2 Flowchart.....	10
2.3 Pseudocode.....	11
BAB 3 KESIMPULAN	
3.1 Hasil Praktikum	14
3.2 Saran Pengembangan	15
DAFTAR KEPUSTAKAAN	16

DAFTAR LAMPIRAN

Kode Program 2.1 (Kode Program tugasNestedFor).....	3
Gambar 2.1 (Langkah 1)	5
Gambar 2.2 (Langkah 2)	6
Gambar 2.3 (Langkah 3)	6
Gambar 2.4 (Langkah 4)	6
Gambar 2.5 (Langkah 5)	6
Gambar 2.6 (Langkah 6)	7
Gambar 2.7 (Langkah 7)	7
Gambar 2.8 (Langkah 8)	7
Gambar 2.9 (Langkah 9)	8
Gambar 2.10 (Flowchart).....	10
Gambar 3.1 (Hasil Praktikum tugasNestedFor)	14
Tabel 2.1 (Pseudocode).....	11

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pemrograman merupakan salah satu dasar penting dalam dunia teknologi informasi, di mana setiap sistem komputer bergantung pada logika dan algoritma untuk menjalankan perintah. Salah satu konsep penting yang harus dipahami dalam pemrograman adalah struktur kendali perulangan (*looping*). Perulangan digunakan untuk mengeksekusi perintah yang sama secara berulang tanpa harus menulis kode secara manual berulang kali. Melalui pemahaman struktur ini, programmer dapat menulis kode yang lebih efisien, ringkas, dan mudah dipahami.

Dalam praktikum ini, mahasiswa mempelajari berbagai contoh penerapan perulangan menggunakan struktur *for* dan *nested loop* (perulangan bersarang). Melalui tujuh program yang berbeda, mahasiswa diajak untuk memahami bagaimana perulangan digunakan untuk menampilkan data berurutan, melakukan operasi aritmetika, serta mencetak pola seperti segitiga dan persegi dari simbol tertentu. Praktikum ini juga memperkenalkan penggunaan kelas *Scanner* untuk mengambil input dari pengguna, sehingga program menjadi lebih interaktif dan dinamis. Dengan praktikum ini, mahasiswa dapat menguasai logika dasar dalam membangun algoritma menggunakan perulangan.

Selain itu, pemahaman tentang perulangan juga menjadi pondasi penting dalam pengembangan program yang lebih kompleks di masa depan. Struktur perulangan tidak hanya digunakan untuk menampilkan data sederhana, tetapi juga berperan penting dalam pembuatan algoritma seperti pencarian dan pengurutan. Oleh karena itu, praktikum ini bertujuan untuk membangun kemampuan berpikir logis dan analitis mahasiswa dalam memahami cara kerja komputer menjalankan instruksi berulang secara efisien dan terstruktur.

1.2 Tujuan Praktikum

Tujuan dilakukannya praktikum ini adalah sebagai berikut :

1. Memahami konsep dasar perulangan *for* dalam bahasa pemrograman Java.
2. Mengimplementasikan perulangan untuk menampilkan data secara berurutan dan melakukan perhitungan otomatis.
3. Mempelajari cara kerja *nested loop* untuk membentuk pola atau tampilan tertentu di layar.
4. Mengembangkan kemampuan logika pemrograman melalui penyusunan algoritma berbasis perulangan.

1.3 Manfaat Praktikum

Manfaat dilakukannya praktikum ini adalah sebagai berikut :

1. Mahasiswa memahami fungsi dan cara kerja struktur perulangan dalam pemrograman Java.
2. Mahasiswa mampu menulis program yang lebih efisien dengan menggunakan konsep pengulangan.
3. Mahasiswa dapat membedakan penggunaan *System.out.print* dan *System.out.println* untuk mengatur tampilan output.
4. Mahasiswa memperoleh pengalaman menerapkan logika berpikir sistematis dalam menyelesaikan permasalahan berulang.

BAB II

PEMBAHASAN

2.1 Kode Program

```

1 package Pekan5;
2
3 public class tugasNestedFor {
4
5     public static void main(String[] args) {
6         int[] titik = {0, 4, 8, 12, 12, 8, 4, 0};
7         int lebarMaks = 12; // jumlah titik terbanyak di tengah
8         int totalLebarIsi = lebarMaks + 4; // lebar isi maksimum (<> di kiri & kanan)
9         int lebarFrame = totalLebarIsi + 8; // tambah spasi kiri-kanan agar rata tengah
10
11         System.out.print("#");
12         for (int i = 0; i < lebarFrame; i++) {
13             System.out.print("=");
14         }
15         System.out.println("#");
16
17         for (int i = 0; i < titik.length; i++) {
18             System.out.print("|");
19
20             int spasi = (lebarFrame - (titik[i] + 4)) / 2;
21             for (int s = 0; s < spasi; s++) {
22                 System.out.print(" ");
23             }
24
25             if (titik[i] == 0) {
26                 System.out.print("<>");
27             } else {
28                 System.out.print("<>");
29                 for (int j = 0; j < titik[i]; j++) {
30                     System.out.print(".");
31                 }
32
33                 System.out.print("<>");
34             }
35             for (int s = 0; s < spasi; s++) {
36                 System.out.print(" ");
37             }
38
39             System.out.println("|");
40         }
41
42         System.out.print("#");
43         for (int i = 0; i < lebarFrame; i++) {
44             System.out.print("=");
45         }
46         System.out.println("#");
47     }
48 }

```

Kode Program 2.1 (Kode Program tugasNestedFor)

2.1.1 Uraian Kode Program

1. *package Pekan5;* → Menunjukkan bahwa program ini berada dalam paket bernama *Pekan5*.
2. *public class tugasNestedFor {* → Mendefinisikan kelas utama dengan nama *tugasNestedFor*.
3. *public static void main(String[] args) {* → Merupakan metode utama yang akan dijalankan pertama kali ketika program dieksekusi.
4. *int[] titik = {0, 4, 8, 12, 12, 8, 4, 0};* → Mendeklarasikan array titik yang berisi jumlah titik pada setiap baris. Nilai-nilai ini mengatur pola bertingkat (naik lalu turun).
5. *int lebarMaks = 12;* → Menentukan jumlah titik terbanyak yang berada di bagian tengah pola.
6. *int totalLebarIsi = lebarMaks + 4;* → Menentukan total lebar isi pola, termasuk tambahan karakter di kiri dan kanan.
7. *int lebarFrame = totalLebarIsi + 8;* → Menambahkan ruang kosong di kiri dan kanan agar pola tampak rata tengah dalam bingkai.
8. *System.out.print("#"); ... System.out.println("#");* → Mencetak garis bingkai atas dengan simbol “#” di sisi kiri dan kanan, serta karakter “=” di tengahnya.
9. *for (int i = 0; i < titik.length; i++) {* → Melakukan perulangan sebanyak jumlah elemen dalam array titik untuk membentuk setiap baris pola.
10. *int spasi = (lebarFrame - (titik[i] + 4)) / 2;* → Menghitung jumlah spasi di kiri dan kanan agar pola tetap berada di tengah secara simetris.

11. `System.out.print("/"); ... System.out.println("/");` → Menampilkan batas kiri dan kanan pola dengan karakter “/”.
12. `if (titik[i] == 0)` → Jika jumlah titik adalah 0 (bagian paling atas atau bawah), maka hanya akan menampilkan “◊” tanpa titik di antaranya.
13. `else { System.out.print("<>"); ... System.out.print("."); ... System.out.print("<>"); }` → Jika jumlah titik lebih dari 0, maka program akan mencetak tanda “<”, diikuti titik sesuai jumlah yang ditentukan dalam array, dan diakhiri dengan tanda “>”.
14. `System.out.print(" ");` → Memberikan jarak (spasi) agar posisi pola tetap seimbang di kiri dan kanan.
15. `System.out.print("#"); ... System.out.println("#");` → Menampilkan garis bingkai bawah dengan pola yang sama seperti bagian atas.

2.1.2 Langkah Kerja

1. Membuat kelas baru bernama *tugasNestedFor* dalam *package Pekan5*.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass:

Interfaces:

Which method stubs would you like to create?

☒ public static void main(String[] args)

☐ Constructors from superclass

☐ Inherited abstract methods

Gambar 2.1 (Langkah 1)

2. Program dijalankan dan masuk ke *main*.

```
public static void main(String[] args) {
```

Gambar 2.2 (Langkah 2)

3. Deklarasikan variabel *titik*, *lebarMaks*, *totalLebarIsi*, dan *lebarFrame*

```
int[] titik = {0, 4, 8, 12, 12, 8, 4, 0};
int lebarMaks = 12; // jumlah titik terbanyak di tengah
int totalLebarIsi = lebarMaks + 4; // lebar isi maksimum (<> di kiri & kanan)
int lebarFrame = totalLebarIsi + 8; // tambah spasi kiri-kanan agar rata tengah
```

Gambar 2.3 (Langkah 3)

4. Cetak tanda pagar “#” di kiri dan kanan, serta simbol “=” di tengahnya untuk membuat garis bingkai atas.

```
System.out.print("#");
for (int i = 0; i < lebarFrame; i++) {
    System.out.print("=");
}
System.out.println("#");
```

Gambar 2.4 (Langkah 4)

5. Lakukan perulangan sebanyak 8 kali (sesuaikan dengan panjang *titik*) untuk mencetak setiap baris pola.

```
for (int i = 0; i < titik.length; i++) {
    System.out.print("|");
```

Gambar 2.5 (Langkah 5)

6. Hitung dan cetak jumlah spasi di kiri agar posisi pola tetap ditengah.

```
int spasi = (lebarFrame - (titik[i] + 4)) / 2;
for (int s = 0; s < spasi; s++) {
    System.out.print(" ");
}
```

Gambar 2.6 (Langkah 6)

7. Cetak isi pola: jika titik[I] == 0, maka hanya menampilkan simbol "<>" (baris tanpa titik), jika tidak, cetak tanda "<" diikuti titik sebanyak nilai lalu ditutup dengan tanpa ">".

```
if (titik[i] == 0) {
    System.out.print("<><>");
} else {
    System.out.print("<");
    for (int j = 0; j < titik[i]; j++) {
        System.out.print(".");
    }
    System.out.print(">");
}
```

Gambar 2.7 (Langkah 7)

8. Tambahkan spasi di kanan dan tutup baris dengan tanda vertikal "|".

```
for (int s = 0; s < spasi; s++) {
    System.out.print(" ");
}

System.out.println("|");
```

Gambar 2.8 (Langkah 8)

9. Cetak bagian bawah bingkai yang sama seperti bagian atas.

```
System.out.print("#");
for (int i = 0; i < lebarFrame; i++) {
    System.out.print("=");
}
System.out.println("#");
```

Gambar 2.9 (Langkah 9)

2.1.3 Analisis Hasil

Program *tugasNestedFor* merupakan contoh penerapan perulangan bersarang (*nested loop*) dalam bahasa Java yang berfungsi untuk menampilkan pola simetris di tengah layar dengan tambahan bingkai di sekelilingnya. Program ini memperlihatkan bagaimana kombinasi perulangan dan pengkondisian dapat menghasilkan tampilan visual berbasis teks yang terstruktur.

Di awal program, terdapat array titik berisi nilai {0, 4, 8, 12, 12, 8, 4, 0} yang menentukan jumlah titik di setiap baris, membentuk pola naik dan turun secara simetris. Variabel *lebarMaks*, *totalLebarIsi*, dan *lebarFrame* digunakan untuk menghitung posisi serta memastikan pola tetap berada di tengah. Bagian utama program menggunakan *loop for* untuk mencetak setiap baris, dengan perulangan di dalamnya untuk menampilkan spasi, titik, dan tanda "<>". Kondisi *if-else* membedakan baris yang memiliki titik dan baris kosong di bagian atas serta bawah.

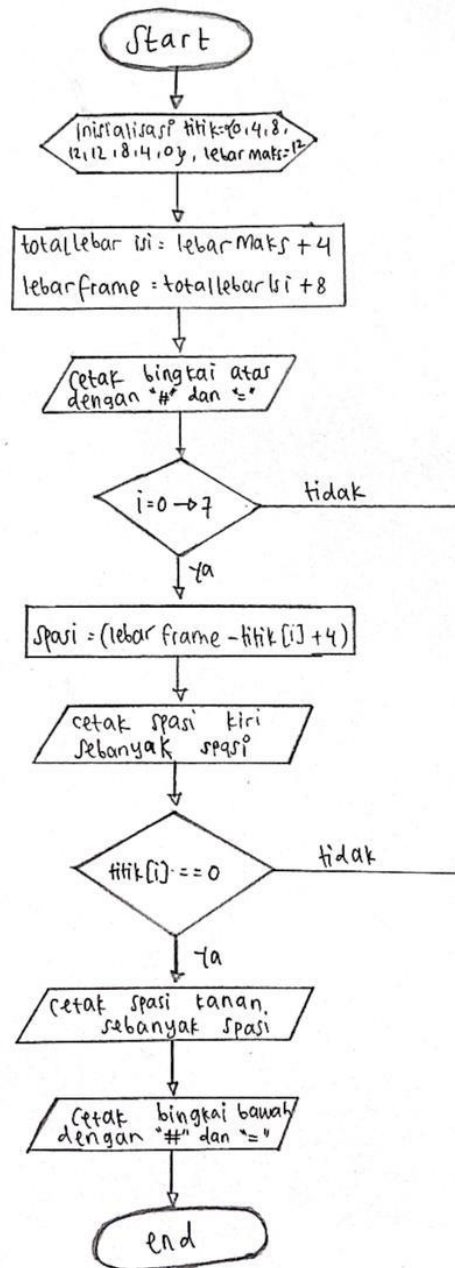
Hasil program menampilkan pola seperti panah horizontal atau bentuk wajik, dengan jumlah titik terbanyak di bagian tengah dan berkurang ke arah atas serta bawah. Pola tampak rata tengah karena adanya perhitungan spasi otomatis. Secara tampilan, hasilnya simetris, rapi, dan menunjukkan penerapan logika *looping* output yang baik.

Berdasarkan teori pemrograman, *nested loop* digunakan untuk membentuk pola dua dimensi dan mengulang proses di dalam perulangan lainnya. Sementara itu, struktur *if-else* membantu pengambilan keputusan dalam pengaturan bentuk pola. Program ini berhasil menerapkan konsep dasar kontrol alur (*control flow*) dan pemformatan teks dengan efektif, sehingga menjadi contoh yang baik dalam memahami pola logika dan tampilan output pada pemrograman terstruktur.

2.2 Flowchart

flowchart
tugas Nestedfor

Kinaya Novrya Manda
2511531016
B Informatika



Gambar 2.10 (Flowchart)

Flowchart dari program *tugasNestedFor* menggambarkan alur kerja logika program secara visual menggunakan simbol-simbol diagram alir. Proses dimulai dari simbol start, dilanjutkan dengan inisialisasi variabel dan perhitungan lebar bingkai. Setelah itu, flowchart menunjukkan proses pencetakan bingkai atas sebelum masuk ke bagian utama yang menggunakan perulangan (*loop*) untuk mencetak pola setiap baris.

Di dalam *loop* terdapat proses perhitungan spasi dan percabangan *if-else* yang menentukan apakah program mencetak simbol “<” atau pola dengan titik di tengah. Setelah setiap baris selesai, program kembali ke awal *loop* hingga semua elemen titik selesai diproses. Terakhir, flowchart menampilkan langkah pencetakan bingkai bawah dan diakhiri dengan simbol end. Diagram ini membantu memahami alur logika program secara cepat tanpa harus membaca kode baris per baris.

2.3 Pseudocode

<p>Judul:</p> <p>Program tugasNestedFor</p> <p>{Program untuk membuat pola simetris menggunakan <i>Nested Loop</i>}</p>
<p>Deklarasi:</p> <p>titik : integer; = {0, 4, 8, 12, 12, 8, 4, 0}</p> <p>lebarMaks : integer; = 12</p> <p>totalLebarIsi, lebarFrame, spasi: integer;</p> <p>i, j, s : integer;</p>
<p>Pseudocode:</p> <ol style="list-style-type: none"> 1. Start 2. Deklarasi int[] titik = {0, 4, 8, 12, 12, 8, 4, 0}

```

3. Set int lebarMaks = 12
4. Set int totalLebarIsi = lebarMaks + 4
5. Set int lebarFrame = totalLebarIsi + 8
6. print("#")
7. For i = 0 to lebarFrame - 1
8.     print("=")
9. End For
10. println("#")
11. For i = 0 to titik.length - 1
12.     print("|")
13.     int spasi = (lebarFrame - (titik[i] + 4)) / 2
14.     For s = 0 to spasi - 1
15.         print(" ")
16.     End For
17.     If titik[i] == 0 then
18.         print("<")
19.     Else
20.         print("<")
21.         For j = 0 to titik[i] - 1
22.             print(".")
23.         End For
24.         print(">")
25.     End If
26.     For s = 0 to spasi - 1
27.         print(" ")
28.     End For
29.     println("|")
30. End For
31. print("#")
32. For i = 0 to lebarFrame - 1
33.     print("=")

```


34.	End For
35.	println("#")
36.	End

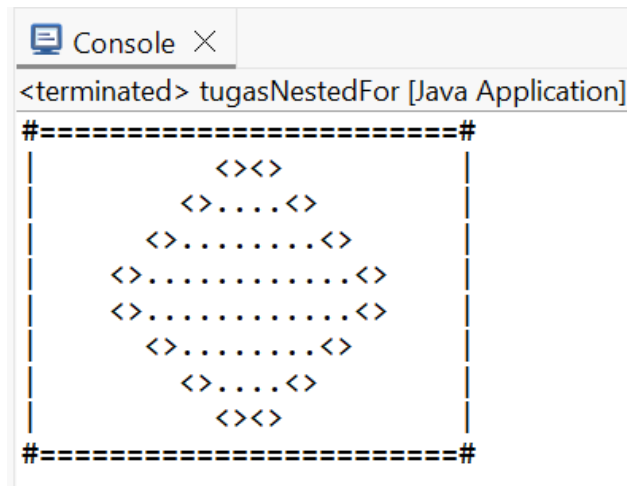
Tabel 2.1 (Pseudocode)

Pseudocode di atas menggambarkan langkah-langkah logika dari program *tugasNestedFor* menggunakan bahasa pemrograman Java. Prosesnya dimulai dengan mendeklarasikan jumlah titik pada setiap baris, serta beberapa variabel seperti *lebarMaks*, *totalLebarIsi*, dan *lebarFrame* yang berfungsi untuk menentukan ukuran dan posisi pola di tengah layar. Setelah itu, program mencetak bingkai bagian atas menggunakan simbol “#” dan “=”.

Kemudian dilakukan perulangan *for* untuk mencetak isi pola, di mana setiap baris dihitung jumlah spasinya agar posisi tetap rata tengah. Struktur *if-else* digunakan untuk menentukan apakah baris tersebut hanya mencetak “<” atau mencetak “<” diikuti titik sesuai nilai array *titik[i]* lalu diakhiri dengan “>”. Setelah semua baris selesai dicetak, program menutup tampilan dengan mencetak bingkai bawah yang sama seperti bagian atas. Pseudocode ini menampilkan logika utama program secara terstruktur dan mudah diterjemahkan menjadi kode Java yang lengkap.

BAB III KESIMPULAN

3.1 Hasil Praktikum



Gambar 3.1 (Hasil Praktikum tugasNestedFor)

Program ini menghasilkan visualisasi berbentuk pola simteris yang dikelilingi oleh bingkai. Bingkai tersebut dibuat menggunakan karakter ”#” di sudut-sudutnya dan karakter ”=” untuk sisi atas dan bawah. Sisi kiri dan kanan bingkai dibentuk oleh karakter ”|”.

Di dalam gambar, terdapat kombinasi karakter <>, <><>, dan ”.”. jumlah karakter ”.” pada setiap baris berbeda, membentuk pola berlian tersebut. Baris tengah memiliki jumlah karakter ”.” terbanyak, dan jumlahnya berkurang secara bertahap ke arah atas dan bawah.

Output ini menunjukkan bagaimana program Java tersebut menggunakan *looping* dan percabangan untuk mencetak karakter-karakter tertentu dalam pola yang terstruktur, sehingga menghasilkan bentuk pola yang simetris.

3.2 Saran Pengembangan

Program diatas sudah mampu menampilkan pola secara simetris dan rapi. Namun, masih terdapat beberapa hal yang dapat dikembangkan agar program menjadi lebih fleksibel dan interaktif. Pertama, program dapat ditambahkan fitur input dari pengguna, misalnya pengguna bisa menentukan sendiri jumlah baris atau nilai maksimum titik, sehingga pola yang dihasilkan bisa bervariasi sesuai kebutuhan. Kedua, program dapat dikembangkan agar pola dan simbol yang digunakan lebih beragam, misalnya dengan mengganti titik menjadi karakter lain seperti bintang (*) atau angka.

Selain itu, program juga bisa ditingkatkan dengan penggunaan metode terpisah (*modularisasi*) agar kode lebih terstruktur dan mudah dibaca. Penggunaan fungsi atau metode khusus untuk mencetak spasi, bingkai, dan isi pola akan membuat program lebih efisien dan mudah dikembangkan. Jika ingin lebih lanjut, hasil pola bisa juga disimpan ke dalam file teks agar pengguna dapat melihat hasilnya tanpa harus menjalankan ulang program.

Dengan beberapa pengembangan tersebut, program ini tidak hanya berguna untuk latihan konsep perulangan bersarang, tetapi juga dapat menjadi contoh penerapan logika tampilan teks yang interaktif dan dinamis dalam bahasa pemrograman Java.

DAFTAR PUSTAKA

- [1] R. A. Johnson, *Fundamentals of Computer Programming with Java*, New York: McGraw-Hill, 2020.
- [2] H. Schildt, *Java: The Complete Reference*, 12th Edition, New York: McGraw-Hill Education, 2021.
- [3] D. Liang, *Introduction to Java Programming and Data Structures*, 12th ed., Pearson Education, 2020.
- [4] G. Brookshear and D. Brylow, *Computer Science: An Overview*, 13th ed., Boston: Pearson, 2019.
- [5] R. Sebesta, *Concepts of Programming Languages*, 12th ed., Pearson, 2022.