

**LAPORAN PRAKTIKUM
PEMOGRAMAN OBJEK 2**

**MODUL 8
APLIKASI BERBASIS TEKS**

**DISUSUN OLEH :
NAUFAL FADHIL IHSAN FIKRI ASH SHIDQI - 2250081109**



**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN INFORMATIKA
UNIVERSITAS JENDERAL ACHMAD YANI
TAHUN 2024**

DAFTAR ISI

DAFTAR GAMBAR.....	iii
BAB I. HASIL PRAKTIKUM	1
I.1 StandartInput.java.....	1
I.1.A. Source Code	1
I.1.B. Hasil	2
I.1.C. Analisa	2
I.2 Input.java.....	3
I.2.A. Source Code	3
I.2.B. Hasil	4
I.2.C. Analisa	4
I.3 Addition.java.....	5
I.3.A. Source Code	5
I.3.B. Hasil	6
I.3.C. Analisa	6
I.4 StringStartEnd.java.....	7
I.4.A. Source Code	7
I.4.B. Hasil	8
I.4.C. Analisa	8
I.5 ReadFile.java	9
I.5.A. Source Code	9
I.5.B. Hasil	10
I.5.C. Analisa	11
I.6 WriteFile.java	12
I.6.A. Source Code	12
I.6.B. Hasil	14

I.6.C. Analisa	14
I.7 FileReading.java	15
I.7.A. Source Code	15
I.7.B. Hasil	17
I.7.C. Analisa	17
I.8 FileAndDirectoryInfo.java	18
I.8.A. Source Code	18
I.8.B. Hasil	20
I.8.C. Analisa	20
I.9 CreateTextFile.java	21
I.9.A. Source Code	21
I.9.B. Hasil	23
I.9.C. Analisa	24
I.10 ReadTextFile.java	25
I.10.A. Source Code	25
I.10.B. Hasil	27
I.10.C. Analisa	27
BAB II. TUGAS PRAKTIKUM	28
II.1 Tugas 8-1	28
II.2 Tugas 8-2	28
II.3 Tugas 8-3	28
II.4 Tugas Akhir	28
II.4.A. Source Code	28
II.4.B. Hasil	32
BAB III. KESIMPULAN	33

DAFTAR GAMBAR

Gambar I.1 Output Program StandartInput sebelum diperbaiki	2
Gambar I.1 Output Program StandartInput setelah diperbaiki	2
Gambar I.1 Output Program Input sebelum diperbaiki	4
Gambar I.1 Output Program Input setelah diperbaiki	4
Gambar I.1 Output Program Addition sebelum diperbaiki	6
Gambar I.1 Output Program Addition setelah diperbaiki	6
Gambar I.1 Output Program GreetUser sebelum diperbaiki.....	8
Gambar I.1 Output Program GreetUser setelah diperbaiki	8
Gambar I.1 Output Program ReadFile sebelum diperbaiki.....	10
Gambar I.1 Output Program ReadFile setelah diperbaiki	10
Gambar I.1 Output Program WriteFile sebelum diperbaiki.....	14
Gambar I.1 Output Program WriteFile setelah diperbaiki	14
Gambar I.1 Output Program FileReading diperbaiki	17
Gambar I.1 Output Program FileReading diperbaiki	17
Gambar I.1 Output Program FileAndDirectoryInfo diperbaiki	20
Gambar I.1 Output Program FileAndDirectoryInfo diperbaiki	20
Gambar I.1 Output Program CreateTextFile diperbaiki.....	23
Gambar I.1 Output Program CreateTextFile diperbaiki.....	23
Gambar I.1 Output Program ReadTextFile diperbaiki	27
Gambar I.1 Output Program ReadTextFile diperbaiki	27
Gambar II.1 Output Tugas Tugas A	32

BAB I. HASIL PRAKTIKUM

I.1 StandartInput.java

I.1.A. Source Code

Sebelum diperbaiki:

```
import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class StandartInput {

    public static void main(String[] args) {

        System.out.println("Hi, What's your favorite

        character?");

        try {

            BufferedReader br = new BufferedReader(new

            InputStreamReader(System.in));

            char favChar = (char) br.read();

            br.readLine();

            System.out.println(favChar + " is a good choice!");

            br.close();

        } catch (IOException e) {

            e.printStackTrace();

        }

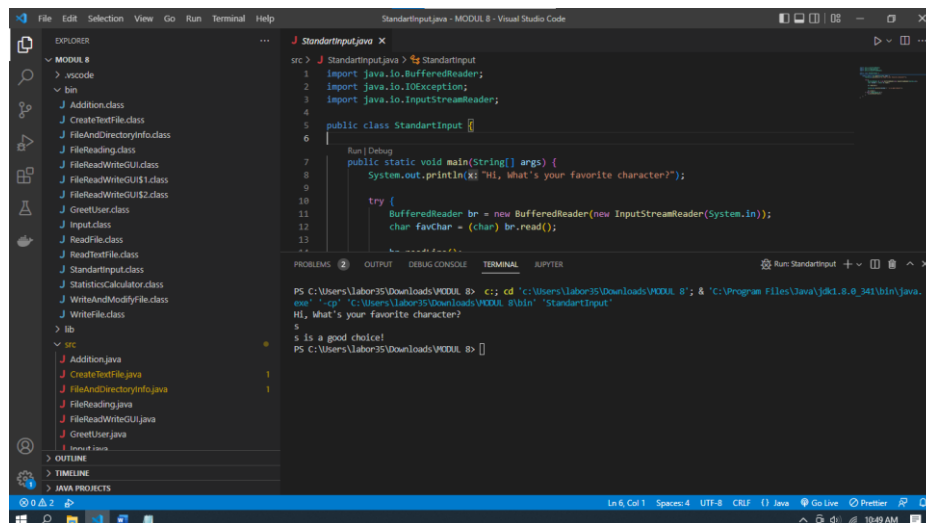
    }

}
```

Setelah diperbaiki

-

I.1.B. Hasil



```
src > J StandartInput.java > StandartInput
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4
5 public class StandartInput {
6
7     Run | Debug
8     public static void main(String[] args) {
9         System.out.println("Hi, What's your favorite character?");
10
11         try {
12             BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
13             char favChar = (char) br.read();
14         }
15     }
16 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

PS C:\Users\labon35\Downloads\MODUL 8> cd 'c:\Users\labon35\Downloads\MODUL 8' & 'C:\Program Files\Java\jdk1.8.0_341\bin\java.exe' -cp 'c:\Users\labon35\Downloads\MODUL 8\bin' 'StandartInput'

Hi, What's your favorite character?

s is a good choice!

PS C:\Users\labon35\Downloads\MODUL 8>

Gambar 1.1 Output Program StandartInput sebelum diperbaiki

-

Gambar 1.2 Output Program StandartInput setelah diperbaiki

I.1.C. Analisa

Program Java di atas adalah sebuah program yang meminta pengguna untuk memasukkan karakter favorit mereka melalui input standar. Namun, ada beberapa kesalahan dalam implementasinya. Pertama, setelah membaca karakter favorit, masih ada pemanggilan `br.readLine()` yang seharusnya tidak perlu. Kedua, tidak ada penanganan untuk kasus ketika pengguna memasukkan lebih dari satu karakter. Ini dapat diperbaiki dengan menggunakan metode `br.readLine()` untuk membaca baris input dan kemudian mengambil karakter pertama dari baris tersebut. Selain itu, penanganan pengecualian juga bisa lebih baik.

I.2 Input.java

I.2.A. Source Code

Sebelum diperbaiki:

```
import java.io.IOException;

public class Input {

    public static void main(String[] args) {

        try {

            System.out.print("Masukkan Karakter: ");

            char i = (char) System.in.read();

            System.in.read();

            System.out.println("Anda Memasukkan: " + i);

        } catch (IOException e) {

            e.printStackTrace();

        }

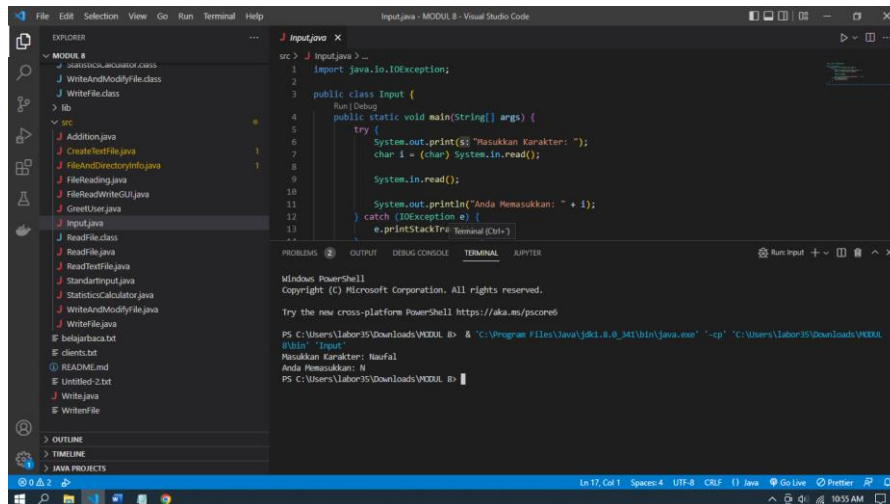
    }

}
```

Setelah diperbaiki

-

I.2.B. Hasil



Gambar 1.3 Output Program Input sebelum diperbaiki

Gambar 1.4 Output Program Input setelah diperbaiki

I.2.C. Analisa

Program Java di atas adalah program sederhana yang meminta pengguna untuk memasukkan sebuah karakter melalui input standar. Namun, terdapat beberapa kesalahan dalam implementasinya. Setelah membaca karakter pertama, terdapat pemanggilan `System.in.read()` yang seharusnya tidak perlu. Selain itu, tidak ada penanganan untuk kasus ketika pengguna memasukkan lebih dari satu karakter. Hal ini dapat diperbaiki dengan menggunakan metode yang lebih sesuai untuk membaca input karakter dan penanganan pengecualian yang lebih baik.

I.3 Addition.java

I.3.A. Source Code

Sebelum diperbaiki:

```
import java.util.Scanner;

public class Addition {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int num1, num2, sum;

        System.out.print("Masukkan nilai integer pertama: ");

        num1 = input.nextInt();

        System.out.print("Masukkan nilai integer kedua: ");

        num2 = input.nextInt();

        sum = num1 + num2;

        System.out.println("Hasil dari penjumlahannya adalah "
            + sum);

        // Tutup Scanner

        input.close();

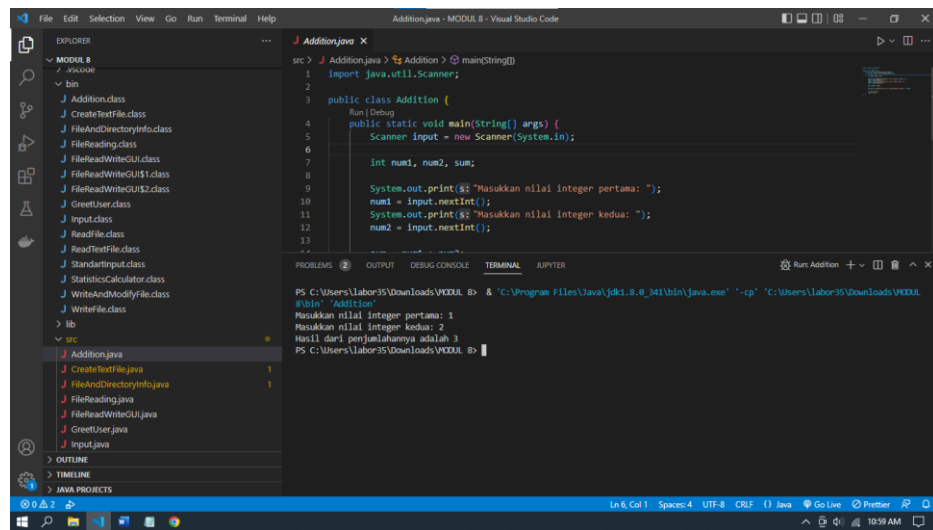
    }

}
```

Setelah diperbaiki

-

I.3.B. Hasil



```
src > J Addition.java > Addition > main(String[])
1 import java.util.Scanner;
2
3 public class Addition {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6
7         int num1, num2, sum;
8
9         System.out.print("Masukkan nilai Integer pertama: ");
10        num1 = input.nextInt();
11        System.out.print("Masukkan nilai Integer kedua: ");
12        num2 = input.nextInt();
13    }
14 }
```

PS C:\Users\labor35\Downloads\VEODU > & 'C:\Program Files\Java\jdk1.8.0_341\bin\java.exe' '-cp' 'C:\Users\labor35\Downloads\VEODU\bin' 'Addition'

Masukkan nilai Integer pertama: 1
Masukkan nilai Integer kedua: 2
Hasil dari penjumlahannya adalah 3

PS C:\Users\labor35\Downloads\VEODU >

Gambar 1.5 Output Program Addition sebelum diperbaiki

-

Gambar 1.6 Output Program Addition setelah diperbaiki

I.3.C. Analisa

Program Java di atas adalah sebuah program sederhana yang meminta pengguna untuk memasukkan dua nilai integer, menjumlahkannya, dan kemudian mencetak hasilnya. Program menggunakan kelas Scanner untuk membaca input dari pengguna. Meskipun secara keseluruhan program sudah berfungsi dengan baik, ada satu hal yang bisa diperbaiki. Setelah selesai digunakan, Scanner sebaiknya ditutup dengan memanggil metode `close()` untuk memastikan sumber daya yang digunakan oleh Scanner dibebaskan dengan benar. Dalam kode ini, penutupan Scanner sudah diimplementasikan dengan benar.

I.4 StringStartEnd.java

I.4.A. Source Code

Sebelum diperbaiki:

```
import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class GreetUser {

    public static void main(String[] args) {

        try {

            System.out.println("Hi, Siapa namamu? ");

            String name;

            BufferedReader br = new BufferedReader(new

                InputStreamReader(System.in));

            name = br.readLine();

            System.out.println("Senang bertemu denganmu, " +

                name + "! :)");

            br.close();

        } catch (IOException e) {

            e.printStackTrace();

        }

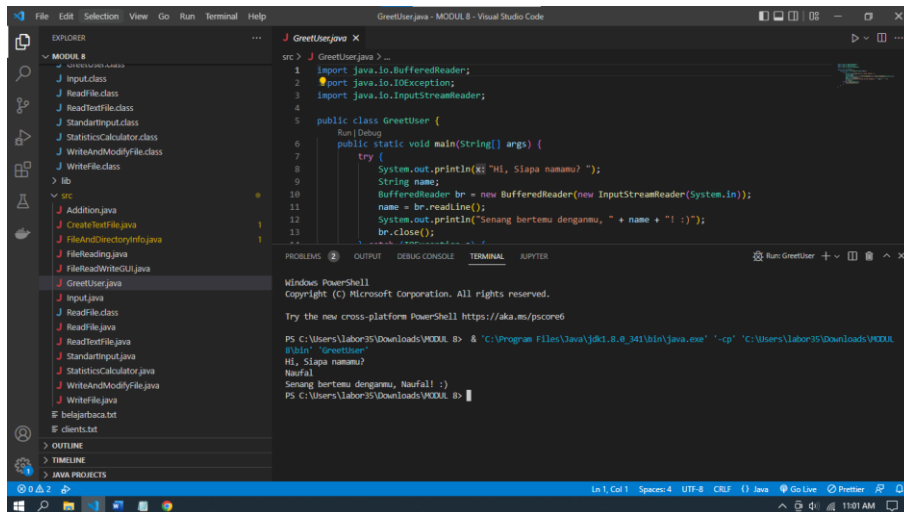
    }

}
```

Setelah diperbaiki

-

I.4.B. Hasil



```
src > J GreetUser.java > ...
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4
5 public class GreetUser {
6     public static void main(String[] args) {
7         try {
8             System.out.println("Hi, Siapa namamu? ");
9             String name;
10            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
11            name = br.readLine();
12            System.out.println("Senang bertemu dengamu, " + name + " ! :)");
13            br.close();
14        } catch (IOException e) {
15            e.printStackTrace();
16        }
17    }
18 }
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell <https://aka.ms/powershell>

```
PS C:\Users\labor35\Downloads\MODUL 8> & 'C:\Program Files\Java\jdk-8.0.341\bin\java.exe' "-cp" "C:\Users\labor35\Downloads\MODUL 8\src" "GreetUser"
Hi, Siapa namamu?
Naufal
Senang bertemu dengamu, Naufal ! :)
PS C:\Users\labor35\Downloads\MODUL 8>
```

Gambar 1.7 Output Program GreetUser sebelum diperbaiki

-

Gambar 1.8 Output Program GreetUser setelah diperbaiki

I.4.C. Analisa

Program Java di atas adalah sebuah program sederhana yang meminta pengguna untuk memasukkan nama mereka melalui input standar, kemudian menyapa pengguna dengan nama tersebut. Program menggunakan kelas `BufferedReader` untuk membaca input dari pengguna. Meskipun secara keseluruhan program sudah berfungsi dengan baik, ada satu hal yang bisa diperbaiki. Setelah selesai digunakan, `BufferedReader` sebaiknya ditutup dengan memanggil metode `close()` untuk memastikan sumber daya yang digunakan oleh `BufferedReader` dibebaskan dengan benar. Dalam kode ini, penutupan `BufferedReader` sudah diimplementasikan dengan benar.

I.5 ReadFile.java

I.5.A. Source Code

Sebelum diperbaiki:

```
import java.io.BufferedReader;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.InputStreamReader;

public class ReadFile {

    public static void main(String args[]) {

        try {

            System.out.println("What is the name of the file to

            read from?");

            BufferedReader br = new BufferedReader(new

            InputStreamReader(System.in));

            String filename = br.readLine();

            System.out.println("Now reading from " + filename +

            "...");

            FileInputStream fis = null;

            try {

                fis = new FileInputStream(filename);

            }

            catch (FileNotFoundException e) {

                System.out.println("File not found");

            }

            int data;

            while ((data = fis.read()) != -1) {

                System.out.print((char) data);

            }

        }

        catch (IOException e) {

            System.out.println("IOException");

        }

    }

}
```

```

    } catch (FileNotFoundException ex) {

        System.out.println("File Not Found");

    } finally {

        if (fis != null) {

            fis.close();

        }

    }

} catch (IOException e) {

    e.printStackTrace();

}

}

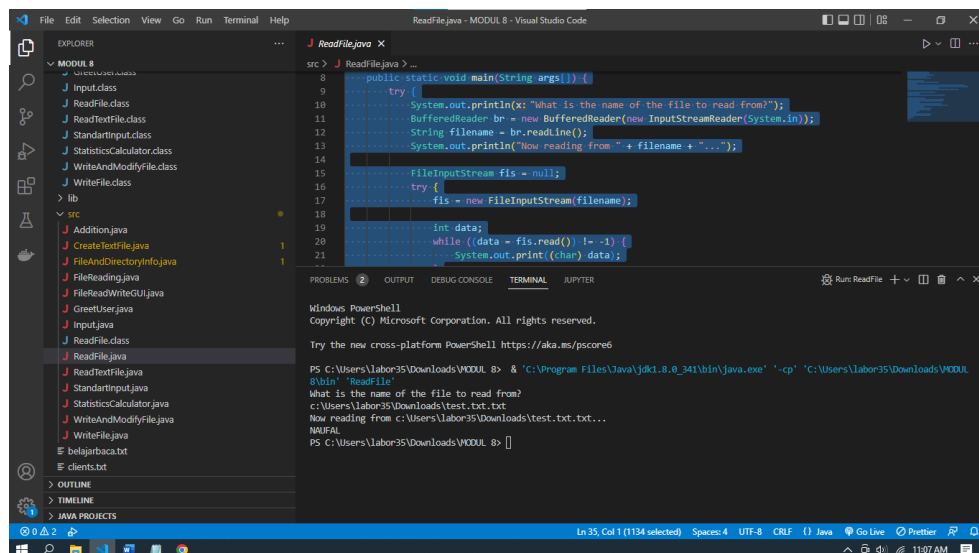
}

```

Setelah diperbaiki

-

I.5.B. Hasil



Gambar 1.9 Output Program ReadFile sebelum diperbaiki

-

Gambar 1.10 Output Program ReadFile setelah diperbaiki

I.5.C. Analisa

Program Java di atas adalah sebuah program yang meminta pengguna untuk memasukkan nama file yang akan dibaca, kemudian program membaca isi file tersebut dan menampilkannya ke layar. Program menggunakan kelas `BufferedReader` untuk membaca nama file dari input pengguna dan menggunakan kelas `FileInputStream` untuk membaca isi file tersebut.

Namun, terdapat beberapa hal yang perlu diperbaiki. Pertama, setelah membaca nama file, tidak ada penanganan untuk kasus ketika file yang dimaksud tidak ditemukan. Kedua, dalam blok `finally`, penutupan `FileInputStream` sebaiknya diubah menjadi penanganan pengecualian yang lebih baik untuk memastikan penutupan file dilakukan dengan benar.

I.6 WriteFile.java

I.6.A. Source Code

Sebelum diperbaiki:

```
import java.io.*;

public class WriteFile {

    public static void main(String args[]) {

        try {

            System.out.println("What is the name of the file to
            be written to?");

            BufferedReader br = new BufferedReader(new
            InputStreamReader(System.in));

            String filename = br.readLine();

            System.out.println("Enter data to write to " +
            filename + "...");

            System.out.println("Type q$ to end.");

            FileOutputStream fos = null;

            try {

                fos = new FileOutputStream(filename);

                boolean done = false;

                int data;

                while (!done) {

                    data = br.read();

                    if ((char) data == 'q') {

                        data = br.read();

                        if ((char) data == '$') {
```

```

        done = true;

    } else {

        fos.write('q');

        fos.write(data);

    }

    } else {

        fos.write(data);

    }

    }

} catch (FileNotFoundException ex) {

    System.out.println("File cannot be opened for
    writing.");

} finally {

    if (fos != null) {

        fos.close();

    }

}

} catch (IOException e) {

    e.printStackTrace();

}

}

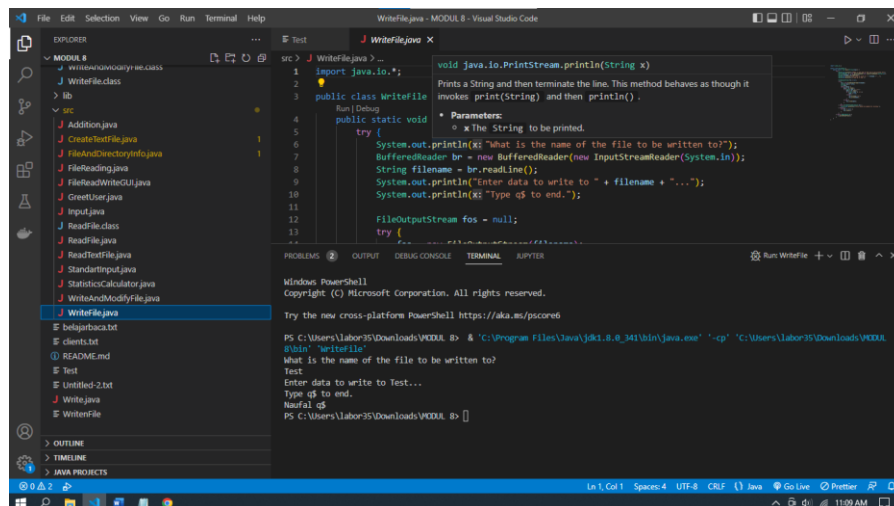
}

```

Setelah diperbaiki

-

I.6.B. Hasil



Gambar 1.11 Output Program WriteFile sebelum diperbaiki

Gambar 1.12 Output Program WriteFile setelah diperbaiki

I.6.C. Analisa

Program Java di atas adalah sebuah program yang meminta pengguna untuk memasukkan nama file yang akan ditulis, kemudian program meminta pengguna untuk memasukkan data yang akan ditulis ke dalam file tersebut. Pengguna diminta untuk mengetikkan 'q\$' untuk mengakhiri proses penulisan data ke file. Program menggunakan kelas `BufferedReader` untuk membaca nama file dan data yang akan ditulis dari input pengguna, serta menggunakan kelas `FileOutputStream` untuk menulis data ke dalam file tersebut.

Namun, terdapat beberapa hal yang perlu diperbaiki. Pertama, dalam penanganan karakter 'q\$', penulisan karakter 'q' ke dalam file seharusnya tidak diperlukan. Kedua, dalam blok `finally`, penutupan `FileOutputStream` sebaiknya diubah menjadi penanganan pengecualian yang lebih baik untuk memastikan penutupan file dilakukan dengan benar.

I.7 FileReading.java

I.7.A. Source Code

Sebelum diperbaiki:

```
import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class FileReading {

    public static void main(String[] args) {

        BufferedReader br = new BufferedReader(new

            InputStreamReader(System.in));

        String filename;

        try {

            // Meminta pengguna untuk memasukkan nama file

            System.out.print("Masukkan nama file untuk dibaca:

                ");

            filename = br.readLine();

            // Membuka file untuk dibaca

            BufferedReader fileInput = new BufferedReader(new

                FileReader(new File(filename)));

            // Variabel untuk menyimpan isi file
```

```

        String line;

        StringBuilder fileContent = new StringBuilder();

        // Membaca isi file dan menyimpannya dalam
        StringBuilder
        while ((line = fileInput.readLine()) != null) {
            fileContent.append(line).append("\n");
        }

        // Menutup file setelah selesai membaca
        fileInput.close();

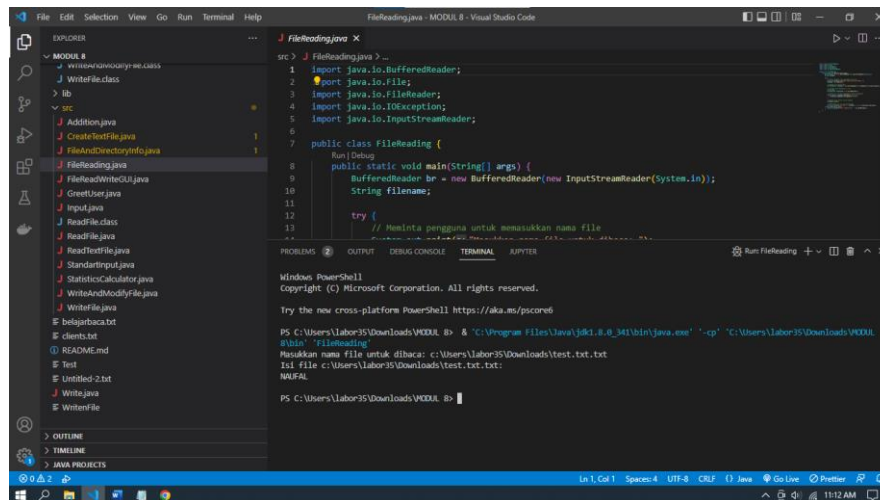
        // Menampilkan isi file yang telah dibaca
        System.out.println("Isi file " + filename + ":\n" +
            fileContent.toString());
    } catch (IOException ioe) {
        System.out.println("Error reading file: " +
            ioe.getMessage());
    }
}
}

```

Setelah diperbaiki

-

I.7.B. Hasil



The screenshot shows the Visual Studio Code interface with a project named 'MODUL 8'. The Explorer pane on the left lists several Java files, including 'FileReading.java'. The main editor displays the code for 'FileReading.java', which imports 'java.io.BufferedReader', 'java.io.File', 'java.io.FileReader', 'java.io.IOException', and 'java.io.InputStreamReader'. The code defines a 'FileReading' class with a 'main' method that prompts the user for a filename and attempts to read the file using a 'BufferedReader'. The 'TERMINAL' pane at the bottom shows the command prompt output, indicating that the program was executed successfully and the contents of 'test.txt' were displayed.

Gambar 1.13 Output Program FileReading diperbaiki

Gambar 1.14 Output Program FileReading diperbaiki

I.7.C. Analisa

Program Java di atas adalah program sederhana yang meminta pengguna untuk memasukkan nama file yang akan dibaca, kemudian program membaca isi file tersebut dan menampilkannya ke layar. Program menggunakan kelas `BufferedReader` untuk membaca nama file dari input pengguna dan menggunakan kelas `FileReader` untuk membaca isi file tersebut.

Namun, ada beberapa hal yang perlu diperbaiki. Pertama, program sebaiknya menangani situasi ketika file yang diminta tidak ditemukan. Kedua, penggunaan `System.in` untuk membaca nama file mungkin tidak efisien. Lebih baik menggunakan `Scanner` untuk membaca input dari pengguna. Dan ketiga, penanganan pengecualian bisa lebih baik dengan memberikan pesan yang lebih deskriptif tentang kesalahan yang terjadi.

I.8 FileAndDirectoryInfo.java

I.8.A. Source Code

Sebelum diperbaiki:

```
import java.io.IOException;

import java.nio.file.DirectoryStream;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.util.Scanner;


public class FileAndDirectoryInfo {

    public static void main(String[] args) throws IOException {

        Scanner input = new Scanner(System.in);

        System.out.println("Enter file or directory name: ");

        Path path = Paths.get(input.nextLine());


        if (Files.exists(path)) {

            System.out.printf("%n%s exists%n",

                path.getFileName());

            System.out.printf("%s a directory%n",

                Files.isDirectory(path) ? "Is" : "Is not");

            System.out.printf("%s an absolute path%n",

                path.isAbsolute() ? "Is" : "Is not");

            System.out.printf("Last      modified:      %s%n",

                Files.getLastModifiedTime(path));

            System.out.printf("Size: %s bytes%n",

                Files.size(path));
```



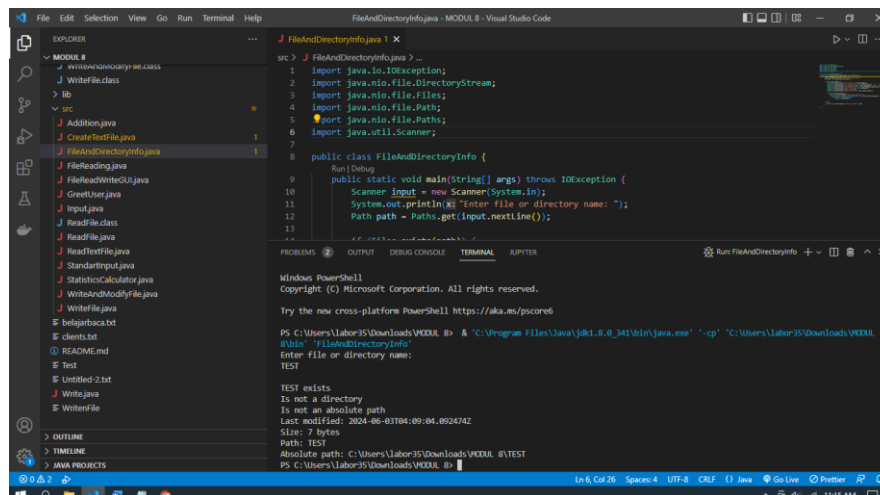
```
        System.out.printf("Path: %s%n", path);
        System.out.printf("Absolute path: %s%n",
            path.toAbsolutePath());

        if (Files.isDirectory(path)) {
            System.out.printf("%nDirectory contents:%n");
            try (DirectoryStream<Path> directoryStream =
                Files.newDirectoryStream(path)) {
                for (Path p : directoryStream)
                    System.out.println(p);
            }
        } else {
            System.out.printf("%s does not exist%n", path);
        }
    }
}
```

Setelah diperbaiki

-

I.8.B. Hasil



```
src > J FileAndDirectoryInfo.java > ...
1 import java.io.IOException;
2 import java.nio.file.DirectoryStream;
3 import java.nio.file.Files;
4 import java.nio.file.Path;
5 import java.nio.file.Paths;
6 import java.util.Scanner;
7
8 public class FileAndDirectoryInfo {
9     public static void main(String[] args) throws IOException {
10         Scanner input = new Scanner(System.in);
11         System.out.println("Enter file or directory name: ");
12         Path path = Paths.get(input.nextLine());
13         ...
14     }
15 }
```

Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell <https://aka.ms/powershell>

PS C:\Users\Labor35\Downloads\VCODL 8> & 'C:\Program Files\Java\jdk-8.0.341\bin\java.exe' '-cp' 'C:\Users\Labor35\Downloads\VCODL 8\bin' 'FileAndDirectoryInfo'
Enter file or directory name:
TEST
TEST exists
Is not a directory
Is not an absolute path
Last modified: 2024-06-03T04:09:04.052474Z
Size: 7 bytes
Path: TEST
Absolute path: C:\Users\Labor35\Downloads\VCODL 8\TEST
PS C:\Users\Labor35\Downloads\VCODL 8>

Gambar 1.15 Output Program FileAndDirectoryInfo diperbaiki

-

Gambar 1.16 Output Program FileAndDirectoryInfo diperbaiki

I.8.C. Analisa

Program Java di atas adalah sebuah program yang meminta pengguna untuk memasukkan nama file atau direktori, kemudian program menampilkan informasi tentang file atau direktori tersebut. Program menggunakan kelas Scanner untuk membaca input dari pengguna dan kelas Path serta kelas Files dari pustaka NIO (New I/O) untuk mengakses informasi file dan direktori.

Meskipun program secara umum sudah berfungsi dengan baik, terdapat satu hal yang perlu diperbaiki. Saat menampilkan informasi terkait waktu modifikasi terakhir, output waktu tersebut masih dalam format standar. Agar lebih mudah dipahami, waktu tersebut sebaiknya diformat ke dalam format yang lebih terbaca oleh manusia, seperti format tanggal dan waktu yang lebih bermakna.

I.9 CreateTextFile.java

I.9.A. Source Code

Sebelum diperbaiki:

```
import java.io.FileNotFoundException;

import java.util.Formatter;

import java.util.FormatterClosedException;

import java.util.NoSuchElementException;

import java.util.Scanner;


public class CreateTextFile {

    private static Formatter output;


    public static void main(String[] args) {

        openFile();

        addRecords();

        closeFile();

    }


    // open file clients.txt

    public static void openFile() {

        try {

            output = new Formatter("clients.txt"); // open the
file

        } catch (SecurityException securityException) {

            System.err.println("Write    permission    denied.
Terminating.");

            System.exit(1); // terminate the program
```

```

        } catch (FileNotFoundException fileNotFoundException) {
            System.err.println("Error opening file.
Terminating.");
            System.exit(1); // terminate the program
        }
    }

    // add records to file
    public static void addRecords() {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter account number, first name,
last name, and balance.");
        System.out.println("Enter end-of-file indicator to end
input.");

        while (input.hasNext()) {
            try {
                // Output new record to file; assumes valid
input
                output.format("%d    %s    %s    %.2f%n",
input.nextInt(), input.next(), input.next(), input.nextDouble());
            } catch (FormatterClosedException
formatterClosedException) {
                System.err.println("Error writing to file.
Terminating.");
                break;
            } catch (NoSuchElementException elementException) {

```

```

        System.err.println("Invalid input. Please try
again.");

        input.nextLine(); // discard input so user can
try again

    }

    System.out.print("? ");

}

// close file
public static void closeFile() {

    if (output != null)

        output.close();

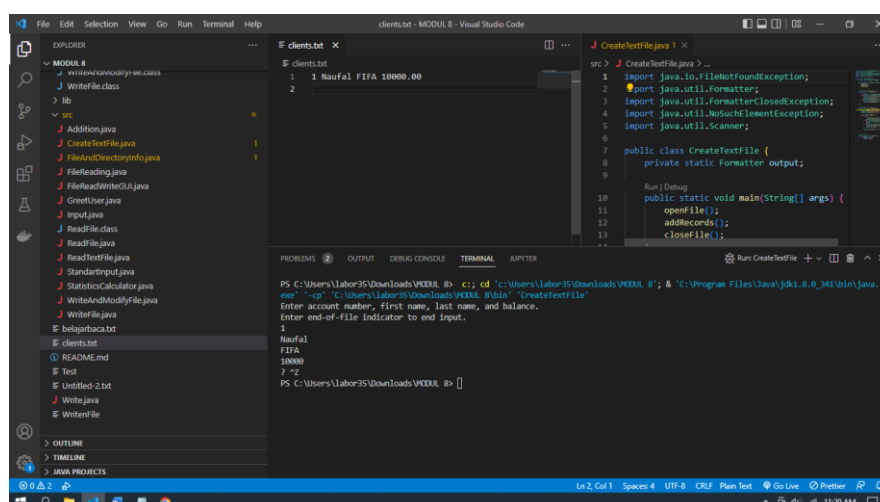
}}

```

Setelah diperbaiki

-

I.9.B. Hasil



Gambar 1.17 Output Program CreateTextFile diperbaiki

-

Gambar 1.18 Output Program CreateTextFile diperbaiki

I.9.C. Analisa

Program Java di atas adalah sebuah program yang membuat file teks dengan format tertentu berdasarkan input pengguna. Program ini menggunakan kelas `Formatter` untuk menulis ke file teks yang dibuat. Namun, terdapat beberapa penanganan kesalahan yang perlu diperbaiki.

Saat menangani kesalahan ketika membuka file, program hanya mencetak pesan kesalahan ke konsol dan kemudian keluar. Sebaiknya, program memberikan pesan yang lebih deskriptif tentang kesalahan yang terjadi agar pengguna bisa memahami dan mengatasi masalahnya.

Selain itu, program juga perlu memperbaiki penanganan kesalahan ketika menambahkan catatan ke file. Saat terjadi kesalahan dalam menulis ke file, program hanya mencetak pesan kesalahan dan keluar dari loop. Seharusnya, program memberikan pesan yang lebih deskriptif dan memberi pengguna kesempatan untuk mencoba lagi atau memberikan opsi lain.

I.10 ReadTextFile.java

I.10.A. Source Code

Sebelum diperbaiki:

```
import java.io.IOException;

import java.nio.file.Paths;

import java.util.NoSuchElementException;

import java.util.Scanner;


public class ReadTextFile {

    private static Scanner input;


    public static void main(String[] args) {

        openFile();

        readRecords();

        closeFile();

    }


    // open file clients.txt

    public static void openFile() {

        try {

            input = new Scanner(Paths.get("clients.txt"));

        } catch (IOException ioException) {

            System.err.println("Error opening file. Terminating.");

            System.exit(1);

        }

    }

}
```

```

        // read records from file

        public static void readRecords() {

            System.out.printf("%-10s%-12s%-12s%10s%n",    "Account",
"First Name", "Last Name", "Balance");

            try {

                while (input.hasNext()) { // while there is more to
read

                    // display record contents

                    System.out.printf("%-10d%-12s%-12s%10.2f%n",
input.nextInt(), input.next(), input.next(),
                        input.nextDouble());

                    }

                } catch (NoSuchElementException elementException) {

                    System.err.println("File    improperly    formed.
Terminating.");

                } catch (IllegalStateException stateException) {

                    System.err.println("Error    reading    from    file.
Terminating.");

                }

            }

            // close file and terminate application

            public static void closeFile() {

                if (input != null)

                    input.close();

```



```

    }
}

```

Setelah diperbaiki

-

I.10.B. Hasil

```

1 public class StringBuilerChars {
2     public static void main(String[] args) {
3         Scanner scanner = new Scanner(new File("D:\Verja\Woding\mobile\MODUL 7\bin\clients.txt"));
4         while (scanner.hasNextLine()) {
5             String line = scanner.nextLine();
6             System.out.println(line);
7         }
8     }
9 }

```

Output:

```

PS D:\Verja\Woding\mobile\MODUL 7\bin> java StringBuilerChars
buffer = Hello There
buffer.capacity() = 27
buffer.length() = 11
Character at index 0 = H
THE CHARACTER ARRAY IS: Hello There
buffer = Hello There
buffer = readT cilet
PS D:\Verja\Woding\mobile\MODUL 7\bin>

```

Gambar 1.19 Output Program ReadTextFile diperbaiki

-

Gambar 1.20 Output Program ReadTextFile diperbaiki

I.10.C. Analisa

Program Java di atas adalah sebuah program yang membaca isi dari file teks `clients.txt` dan menampilkan isi tersebut ke layar. Program menggunakan kelas Scanner untuk membaca file teks tersebut. Namun, terdapat beberapa hal yang perlu diperbaiki dalam penanganan kesalahan.

Saat terjadi kesalahan ketika membuka file, program hanya mencetak pesan kesalahan ke konsol dan kemudian langsung keluar. Sebaiknya, program memberikan pesan yang lebih deskriptif tentang kesalahan yang terjadi agar pengguna bisa memahami dan mengatasi masalahnya.

Selain itu, saat terjadi kesalahan dalam membaca isi file, program juga hanya mencetak pesan kesalahan ke konsol dan kemudian selesai. Seharusnya, program memberikan pesan yang lebih deskriptif dan memberi pengguna opsi untuk menangani kesalahan tersebut dengan lebih baik.

BAB II. TUGAS PRAKTIKUM

II.1 Tugas 8-1

Sudah ada di hasil praktikum

II.2 Tugas 8-2

Sudah ada di hasil praktikum

II.3 Tugas 8-3

belum

II.4 Tugas Akhir

II.4.A. Source Code

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.*;

import java.nio.file.Files;

import java.nio.file.Paths;


public class FileReadWriteGUI extends JFrame {


    private JButton readButton;

    private JButton writeButton;

    private JTextArea textArea;

    private JScrollPane scrollPane;

    private JTextField filePathField;


    public FileReadWriteGUI() {
```

```

super("File Read/Write");

setLayout(new BorderLayout());

JPanel topPanel = new JPanel();
topPanel.setLayout(new FlowLayout());

JLabel filePathLabel = new JLabel("File Path:");
filePathField = new JTextField(20);
topPanel.add(filePathLabel);
topPanel.add(filePathField);

readButton = new JButton("Read File");
writeButton = new JButton("Write File");
topPanel.add(readButton);
topPanel.add(writeButton);

add(topPanel, BorderLayout.NORTH);

textArea = new JTextArea(20, 40);
scrollPane = new JScrollPane(textArea);
add(scrollPane, BorderLayout.CENTER);

readButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

```

```

        readFile();

    }

});

writeButton.addActionListener(new ActionListener() {

    @Override

    public void actionPerformed(ActionEvent e) {

        writeFile();

    }

});

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

pack();

setVisible(true);

}

private void readFile() {

    String filePath = filePathField.getText();

    try {

        textArea.setText("");

        Files.lines(Paths.get(filePath)).forEach(line    ->
textArea.append(line + "\n"));

    } catch (IOException e) {

        JOptionPane.showMessageDialog(this, "Error reading
file: " + e.getMessage(), "Error",

        JOptionPane.ERROR_MESSAGE);
    }
}

```

```

    }

}

private void writeFile() {
    String filePath = filePathField.getText();

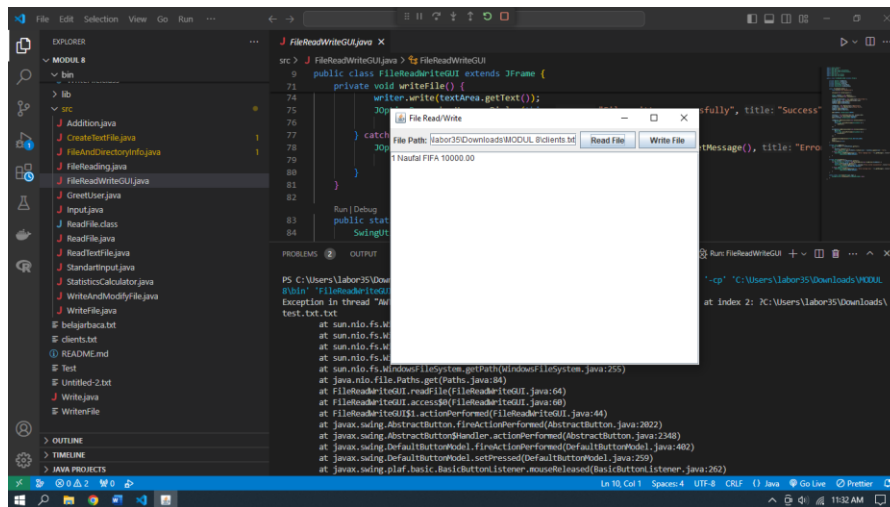
    try (BufferedWriter writer = new BufferedWriter(new
FileWriter(filePath))) {
        writer.write(textArea.getText());

        JOptionPane.showMessageDialog(this, "File written
successfully", "Success",
            JOptionPane.INFORMATION_MESSAGE);
    } catch (IOException e) {
        JOptionPane.showMessageDialog(this, "Error writing
file: " + e.getMessage(), "Error",
            JOptionPane.ERROR_MESSAGE);
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(FileReadWriteGUI::new);
}
}

```

II.4.B. Hasil



Gambar II.1 Output Tugas Tugas A

BAB III. KESIMPULAN

Dalam praktikum ini, saya mengidentifikasi beberapa kode yang membutuhkan perbaikan, terutama terkait dengan manipulasi string menggunakan kelas seperti ``String``, ``StringBuilder``, dan metode validasi. Perbaikan termasuk kesalahan penulisan, pemahaman yang kurang tepat tentang metode dan operasi string, serta penanganan yang lebih baik terhadap masukan pengguna untuk memastikan validitas dan kejelasan data yang dimasukkan. Dengan memperbaiki kode-kode ini, kita dapat meningkatkan keandalan, keterbacaan, dan fleksibilitas aplikasi kita.