

Mata Kuliah : Pemrograman Berorientasi Objek
Tanggal : 01 Oktober 2024
Semester : 3 (Gasal)
Topik : Studi Kasus Java OOP: Improvement Feature
Minggu/Pertemuan/Sesi : 06/02/01-02
Aktivitas : Praktikum
Durasi : 340 menit
Setoran : Dokumen Laporan PDF
Batas Penyerahan : -
Tempat Penyerahan : <https://ecourse.del.ac.id/>

Daftar Isi

A.	Tujuan & Penilaian	2
1.	Tujuan	2
2.	Penilaian	2
3.	Dokumen Laporan	2
B.	Aktivitas Praktikum	3
1.	Latihan Array List	3
a)	Method	3
b)	Searching	5
c)	Sorting	6
d)	Sorting Object	6
2.	Studi Kasus Java OOP: Improvment Feature	8
a)	Setup Proyek	8
b)	Membuat Entity	8
c)	Membuat Repository	9
d)	Membuat Service	10
e)	Membuat Util	11
f)	Membuat View	12
g)	Menjalankan Aplikasi	13
3.	Fitur Update	14
4.	Fitur Sorting	15
5.	Fitur Searching	18

A. Tujuan & Penilaian

1. Tujuan

- ✓ Mahasiswa mampu menggunakan Java OOP untuk menyelesaikan studi kasus tertentu.

2. Penilaian

No	Kriteria	Bobot Penilaian (%)
1	Latihan Array List	5
2	Studi Kasus Java OOP: Improvment Feature	5
3	Fitur Update	20
4	Fitur Sorting	40
5	Fitur Searching	30

3. Dokumen Laporan

Buat laporan praktikum menggunakan [[Template Laporan Praktikum](#)] [[Contoh Laporan Praktikum](#)]. Sebelum mengirimkan dokumen laporan pada E-Course silahkan ubah penamaan file dengan format:

{NIM}-pbo-laporan-praktikum-6.pdf

Sebagai contoh:

11S18005-pbo-laporan-praktikum-6.pdf

B. Aktivitas Praktikum

1. Latihan Array List

ArrayList adalah struktur data yang digunakan untuk menyimpan sekumpulan objek dengan tipe data yang sama, memungkinkan penambahan, penghapusan, dan akses cepat.

a) Method

Method di Java adalah blok kode yang dapat dipanggil untuk menjalankan tugas tertentu, mungkin dengan atau tanpa parameter, dan dapat mengembalikan nilai (tipe kembalian) atau tidak.

| Method di ArrayList

Method	Keterangan
add()	Menambahkan elemen ke dalam ArrayList.
get()	Mengambil elemen dari ArrayList berdasarkan indeks.
remove()	Menghapus elemen dari ArrayList berdasarkan indeks atau objek yang sesuai.
size()	Mengembalikan jumlah elemen dari ArrayList.
clear()	Menghapus semua elemen dari ArrayList.
isEmpty()	Memeriksa apakah ArrayList kosong atau tidak.
contains()	Memeriksa apakah ArrayList berisi element tertentu.
indexOf()	Mengembalikan indeks pertama dari elemen tertentu dalam ArrayList.
toArray()	Mengkonversi ArrayList menjadi array biasa.

Buat dan modifikasi isi file **"Ex1.java"**, seperti berikut:

Ex1.java	
1	import java.util.ArrayList;
2	
3	public class Ex1 {
4	
5	public static void main(String[] args) {
6	// Deklarasi dan inisialisasi
7	ArrayList<String> data = new ArrayList<String>();
8	
9	// Menambahkan elemen
10	data.add("Belajar PBO");
11	data.add("Belajar PAM");
12	
13	// Menambahkan elemen di index tertentu
14	data.add(0, "Belajar PABWE");
15	
16	ShowDataByIndex(data);
17	
18	// Menghapus elemen berdasarkan index
19	data.remove(0);
20	
21	ShowData(data);
22	

Ex1.java

```
23 // Menghapus elemen berdasarkan objek
24 String target = "Belajar PAM";
25 data.remove(target);
26 ShowData(data);
27
28 // Mengambil jumlah data
29 int size = data.size();
30 System.out.println("Jumlah Data : " + size);
31 System.out.println();
32
33 // Memeriksa apakah data kosong
34 boolean isEmpty = data.isEmpty();
35 System.out.println("Apakah data kosong? " + isEmpty);
36 System.out.println();
37
38 // Menghapus semua elemen
39 data.clear();
40
41 // Memeriksa apakah data kosong
42 isEmpty = data.isEmpty();
43 System.out.println("Apakah data kosong? " + isEmpty);
44 System.out.println();
45
46 // Menambahkan elemen
47 data.add("Belajar PBO");
48 data.add("Belajar PAM");
49 data.add("Belajar PABWE");
50
51 ShowData(data);
52
53 // Memeriksa apakah data tersedia
54 String str = "Belajar PBO";
55 boolean isExist = data.contains(str);
56 System.out.println("Apakah data tersedia? " + isExist);
57 System.out.println();
58
59 // Mengembalikan index pertama dari data
60 int index = data.indexOf(str);
61 System.out.println("Index data berada pada posisi : " + index);
62 System.out.println();
63
64 // Mengkonversi ke array biasa
65 String[] newData = new String[data.size()];
66 data.toArray(newData);
67 System.out.println("Hasil Konversi ke Array Biasa :");
68 for (int i = 0; i < newData.length; i++) {
69     int position = i + 1;
70     String string = newData[i];
71     System.out.println(position + ". " + string);
72 }
73 System.out.println();
74
75 }
76
77 public static void ShowDataByIndex(ArrayList<String> data) {
78     System.out.println("Data Tersedia :");
79     for (int i = 0; i < data.size(); i++) {
80         int position = i + 1;
81         String string = data.get(i);
82         System.out.println(position + ". " + string);
83     }
84     System.out.println();
85 }
86
87 public static void ShowData(ArrayList<String> data) {
88     System.out.println("Data Tersedia :");
```

Ex1.java

```
89     for (String string : data) {
90         int position = data.indexOf(string) + 1;
91         System.out.println(position + ". " + string);
92     }
93     System.out.println();
94 }
95
96 }
```

Analisis dan amati hasil kode program di atas pada terminal.

b) Searching

Buat dan modifikasi isi file **"Ex2.java"**, seperti berikut:

Ex2.Java

```
1  import java.util.ArrayList;
2
3  public class Ex2 {
4
5      public static void main(String[] args) {
6          ArrayList<String> data = new ArrayList<String>();
7
8          data.add("Belajar PAM");
9          data.add("Belajar PBO");
10         data.add("Belajar PABWE");
11         data.add("Belajar ALSTRU DAT");
12         data.add("Belajar AOK");
13         data.add("Belajar SISOP");
14         data.add("Belajar EHPT");
15         data.add("Belajar PABSER");
16         data.add("Belajar PAPK");
17         data.add("Istirahat Sejenak");
18
19         ShowData(data);
20
21         SearchData(data, "PA");
22
23         SearchData(data, "enak");
24     }
25
26     public static void ShowData(ArrayList<String> data) {
27         System.out.println("Data Tersedia :");
28         for (String string : data) {
29             int position = data.indexOf(string) + 1;
30             System.out.println(position + ". " + string);
31         }
32         System.out.println();
33     }
34
35     public static void SearchData(ArrayList<String> data, String
36 keywords) {
37         System.out.println("Data Tersedia untuk Hasil Pencarian \"" +
38 keywords + "\" :");
39         int position = 0;
40         for (String string : data) {
41             if (string.contains(keywords)) {
42                 position++;
43                 System.out.println(position + ". " + string);
44             }
45         }
46         System.out.println();
47     }
48 }
```

Analisis dan amati hasil kode program di atas pada terminal.

c) Sorting

Buat dan modifikasi isi file **"Ex3.java"**, seperti berikut:

Ex3.java

```
1 import java.util.ArrayList;
2 import java.util.Collections;
3
4 public class Ex3 {
5
6     public static void main(String[] args) {
7         ArrayList<String> data = new ArrayList<String>();
8
9         data.add("Belajar PAM");
10        data.add("Belajar PBO");
11        data.add("Belajar PABWE");
12        data.add("Belajar ALSTRUDAT");
13        data.add("Belajar AOK");
14        data.add("Belajar SISOP");
15        data.add("Belajar EHPT");
16        data.add("Belajar PABSER");
17        data.add("Belajar PAPK");
18        data.add("Istirahat Sejenak");
19
20        ShowData(data);
21
22        System.out.println("Mengurutkan data secara Ascending : ");
23        SortData(data, true);
24        ShowData(data);
25
26        System.out.println("Mengurutkan data secara Descending : ");
27        SortData(data, false);
28        ShowData(data);
29    }
30
31    public static void ShowData(ArrayList<String> data) {
32        System.out.println("Data Tersedia :");
33        for (String string : data) {
34            int position = data.indexOf(string) + 1;
35            System.out.println(position + ". " + string);
36        }
37        System.out.println();
38    }
39
40    public static void SortData(ArrayList<String> data, boolean
ascending) {
41        if (ascending) {
42            Collections.sort(data);
43        } else {
44            Collections.sort(data, Collections.reverseOrder());
45        }
46    }
47
48 }
```

Analisis dan amati hasil kode program di atas pada terminal.

d) Sorting Object

Buat dan modifikasi isi file **"Ex4.java"**, seperti berikut:

Ex4.java

```
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.Comparator;
```

Ex4.java

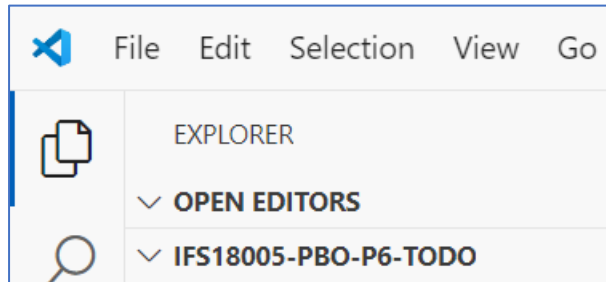
```
4
5 public class Ex4 {
6
7     static class Data {
8         private String name;
9
10        public Data(String name) {
11            this.name = name;
12        }
13
14        public String getName() {
15            return this.name;
16        }
17    }
18
19    public static void main(String[] args) {
20        ArrayList<Data> model = new ArrayList<Data>();
21
22        model.add(new Data("Belajar PAM"));
23        model.add(new Data("Belajar PBO"));
24        model.add(new Data("Belajar PABWE"));
25        model.add(new Data("Belajar ALSTRU DAT"));
26        model.add(new Data("Belajar AOK"));
27        model.add(new Data("Belajar SISOP"));
28        model.add(new Data("Belajar EHPT"));
29        model.add(new Data("Belajar PABSER"));
30        model.add(new Data("Belajar PAPK"));
31        model.add(new Data("Istirahat Sejenak"));
32
33        ShowData(model);
34
35        System.out.println("Mengurutkan data secara Ascending : ");
36        SortData(model, true);
37        ShowData(model);
38
39        System.out.println("Mengurutkan data secara Descending : ");
40        SortData(model, false);
41        ShowData(model);
42    }
43
44    public static void ShowData(ArrayList<Data> model) {
45        System.out.println("Data Tersedia :");
46        for (Data data : model) {
47            int position = model.indexOf(data) + 1;
48            System.out.println(position + ". " + data.name);
49        }
50        System.out.println();
51    }
52
53    public static void SortData(ArrayList<Data> model, boolean ascending)
54    {
55        Comparator<Data> comparator = Comparator.comparing(Data::getName);
56        if (!ascending) {
57            comparator = comparator.reversed();
58        }
59        Collections.sort(model, comparator);
60    }
61
62 }
```

Analisis dan amati hasil kode program di atas pada terminal.

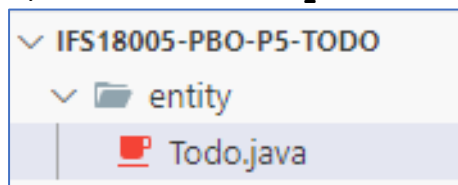
2. Studi Kasus Java OOP: Improvment Feature

a) Setup Proyek

Buat workspace baru dengan format penamaan "{username}-pbo-p6-todo".



b) Membuat Entity



Kode program "entity/TODO.java":

```

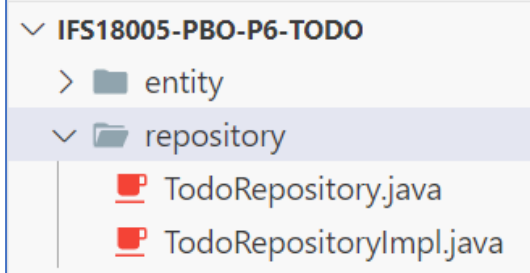
Todo.java
1  package entity;
2
3  public class Todo {
4      public static int total = 0;
5
6      private int id;
7      private String title;
8      private boolean finished;
9
10     private void setId() {
11         total++;
12         id = total;
13     }
14
15     public Todo() {
16         setId();
17     }
18
19     public Todo(String title) {
20         setId();
21         this.title = title;
22     }
23
24     public Todo(String title, boolean finished) {
25         setId();
26         this.title = title;
27         this.finished = finished;
28     }
29
30     public int getId() {
31         return id;
32     }
33
34     public String getTitle() {
35         return title;
36     }
37

```


Todo.java

```
38     public void setTitle(String title) {
39         this.title = title;
40     }
41
42     public boolean isFinished() {
43         return finished;
44     }
45
46     public void setFinished(boolean finished) {
47         this.finished = finished;
48     }
49
50     @Override
51     public String toString() {
52         String status = (finished) ? "Selesai" : "Belum Selesai";
53         return String.format("%d | %s | %s", id, title, status);
54     }
55
56 }
```

c) Membuat Repository



Kode program "repository/TodoRepository.java":

TodoRepository.java

```
1     package repository;
2
3     import java.util.ArrayList;
4     import entity.TODO;
5
6     public interface TodoRepository {
7         ArrayList<TODO> repoGetAllTodo();
8
9         void repoAddTodo(TODO newTodo);
10
11         boolean repoRemoveTodo(Integer idTodo);
12     }
```

Kode program "repository/TodoRepositoryImpl.java":

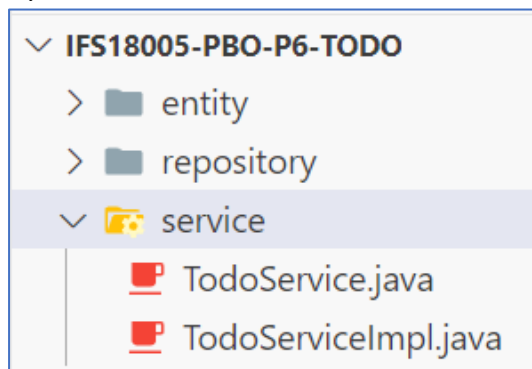
TodoRepositoryImpl.java

```
1     package repository;
2
3     import java.util.ArrayList;
4     import entity.TODO;
5
6     public class TodoRepositoryImpl implements TodoRepository {
7
8         public ArrayList<TODO> data = new ArrayList<TODO>();
9
10        @Override
11        public ArrayList<TODO> repoGetAllTodo() {
12            return data;
13        }
14    }
```

TodoRepositoryImpl.java

```
13     }
14
15     @Override
16     public void repoAddTodo(Todo newTodo) {
17         data.add(newTodo);
18     }
19
20     @Override
21     public boolean repoRemoveTodo(Integer idTodo) {
22         Todo targetTodo = data
23             .stream()
24             .filter(obj -> obj.getId() == idTodo)
25             .findFirst()
26             .orElse(null);
27
28         if (targetTodo == null) {
29             return false;
30         }
31
32         data.remove(targetTodo);
33         return true;
34     }
35 }
```

d) Membuat Service



Kode program "**service/ToDoService.java**":

ToDoService.java

```
1 package service;
2
3 public interface ToDoService {
4     void serviceShowTodo();
5
6     void serviceAddTodo(String title);
7
8     void serviceRemoveTodo(Integer id);
9 }
```

Kode program "**service/ToDoServiceImpl.java**":

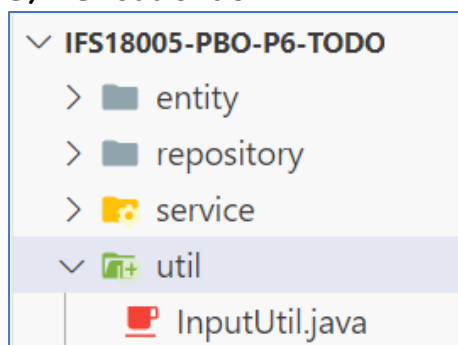
ToDoServiceImpl.java

```
1 package service;
2
3 import java.util.ArrayList;
4
5 import entity.Todo;
6 import repository.TodoRepository;
7
8 public class ToDoServiceImpl implements ToDoService {
```

TodoServiceImpl.java

```
9     private TodoRepository todoRepository;
10
11     public TodoServiceImpl(TodoRepository todoRepository) {
12         this.todoRepository = todoRepository;
13     }
14
15     @Override
16     public void serviceShowTodo() {
17         ArrayList<Todo> todos = todoRepository.repoGetAllTodo();
18         System.out.println("Daftar Todo:");
19
20         if (todos.size() <= 0) {
21             System.out.println("- Data todo belum tersedia!");
22             return;
23         }
24
25         for (Todo todo : todos) {
26             if (todo != null) {
27                 System.out.println(todo);
28             } else {
29                 break;
30             }
31         }
32     }
33
34     @Override
35     public void serviceAddTodo(String todo) {
36         Todo newTodo = new Todo(todo);
37         todoRepository.repoAddTodo(newTodo);
38     }
39
40     @Override
41     public void serviceRemoveTodo(Integer id) {
42         boolean success = todoRepository.repoRemoveTodo(id);
43         if (!success) {
44             System.out.printf("[!] Gagal menghapus todo dengan ID: %d.\n", id);
45             return;
46         }
47     }
48 }
```

e) Membuat Util



Kode program "util/inputUtil.java":

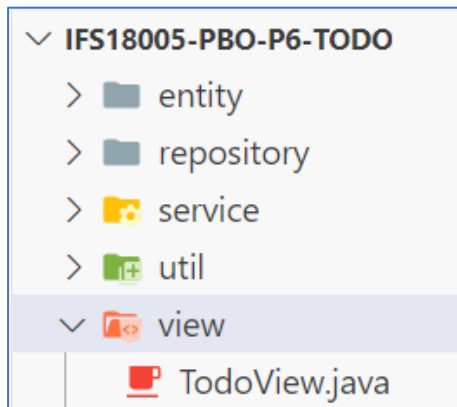
inputUtil.java

```
1     package util;
2
3     import java.util.Scanner;
4
5     public class InputUtil {
6         private static Scanner scanner = new Scanner(System.in);
```

inputUtil.java

```
7
8     public static String input(String info) {
9         System.out.print(info + " : ");
10        String data = scanner.nextLine();
11        return data;
12    }
13
14 }
```

f) Membuat View



Kode program "view/ToDoView.java":

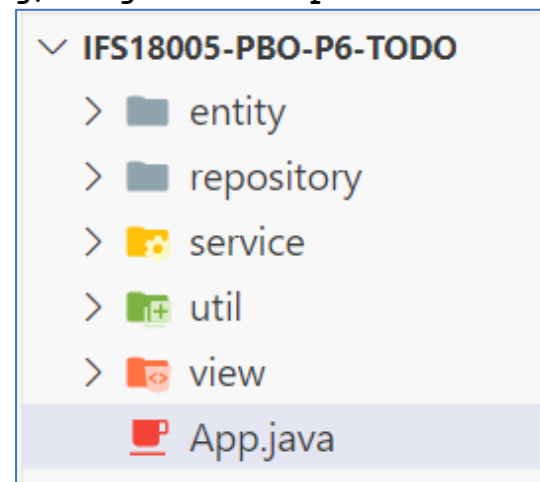
ToDoView.java

```
1     package view;
2
3     import service.TODOService;
4     import util.InputUtil;
5
6     public class ToDoView {
7
8         private TODOService todoService;
9
10        public ToDoView(TODOService todoService) {
11            this.todoService = todoService;
12        }
13
14        /**
15         * Menampilkan view todo
16         */
17        public void viewShowTodo() {
18            while (true) {
19                todoService.serviceShowTodo();
20
21                System.out.println("Menu:");
22                System.out.println("1. Tambah");
23                System.out.println("2. Ubah");
24                System.out.println("3. Hapus");
25                System.out.println("x. Keluar");
26
27                var input = InputUtil.input("Pilih");
28                var stop = false;
29                switch (input) {
30                    case "1" -> viewAddTodo();
31                    case "2" -> viewUpdateTodo();
32                    case "3" -> viewRemoveTodo();
33                    case "x" -> stop = true;
34                    default -> System.out.println("[!] Pilihan tidak dimengerti.");
35                }
36            }
37        }
38    }
```

TodoView.java

```
36
37     if (stop) {
38         break;
39     }
40
41     System.out.println();
42 }
43
44
45 /**
46  * Menampilkan view add todo
47  */
48 public void viewAddTodo() {
49     System.out.println("[Menambah Todo]");
50     var title = InputUtil.input("Judul (x Jika Batal)");
51
52     if (!title.equals("x")) {
53         todoService.serviceAddTodo(title);
54     }
55 }
56
57 /**
58  * Menampilkan view remove todo
59  */
60 public void viewRemoveTodo() {
61     System.out.println("[Menghapus Todo]");
62
63     var strIdTodo = InputUtil.input("[ID Todo] yang dihapus (x Jika
Batal)");
64
65     if (!strIdTodo.equals("x")) {
66         int idTodo = Integer.valueOf(strIdTodo);
67         todoService.serviceRemoveTodo(idTodo);
68     }
69 }
70
71 /**
72  * Menampilkan view update todo
73  */
74 public void viewUpdateTodo() {
75
76 }
77
78 }
```

g) Menjalankan Aplikasi



Kode program "App.java":

```
App.java
1  import repository.TODORepository;
2  import repository.TODORepositoryImpl;
3  import service.TODOService;
4  import service.TODOServiceImpl;
5  import view.TODOView;
6
7  public class App {
8      public static void main(String[] args) {
9          TODORepository todoRepository = new TODORepositoryImpl();
10         TODOService todoService = new TODOServiceImpl(todoRepository);
11         TODOView todoView = new TODOView(todoService);
12
13         todoView.viewShowTODO();
14     }
15 }
```

Jalankan aplikasi pada terminal dengan menuliskan perintah:

- Compile

```
javac -d output App.java
```

- Run

```
java -cp output App
```

[!] Ubah penamaan class App menjadi **TodoV3_{NIM}.java**

3. Fitur Update

Silahkan menambahkan fitur update pada aplikasi todo. Berikut merupakan contoh program saat dijalankan:

```
Daftar TODO:
- Data todo belum tersedia!
Menu:
1. Tambah
2. Ubah
3. Hapus
4. Urutan
5. Cari
x. Keluar
Pilih : 1
[Menambah TODO]
Judul (x Jika Batal) : Belajar OOP

Daftar TODO:
1 | Belajar OOP | Belum Selesai
Menu:
1. Tambah
2. Ubah
3. Hapus
4. Urutan
5. Cari
x. Keluar
Pilih : 2
[Mengubah TODO]
[ID TODO] yang diubah (x Jika Batal) : 1
```

```
[!] Tekan enter untuk menggunakan judul lama.  
Judul baru [Judul lama: Belajar OOP] : Belajar Java OOP  
Status [0 = Belum Selesai, 1 = Selesai] : 1  
  
Daftar Todo:  
1 | Belajar Java OOP | Selesai  
Menu:  
1. Tambah  
2. Ubah  
3. Hapus  
4. Urutan  
5. Cari  
x. Keluar  
Pilih : x
```

(!) Latar belakang oranye merupakan inputan.

4. Fitur Sorting

Silahkan menambahkan fitur sorting pada aplikasi todo. Berikut merupakan contoh program saat dijalankan:

```
Daftar Todo:  
- Data todo belum tersedia!  
Menu:  
1. Tambah  
2. Ubah  
3. Hapus  
4. Urutan  
5. Cari  
x. Keluar  
Pilih : 1  
[Menambah Todo]  
Judul (x Jika Batal) : Belajar Java OOP  
  
Daftar Todo:  
1 | Belajar Java OOP | Belum Selesai  
Menu:  
1. Tambah  
2. Ubah  
3. Hapus  
4. Urutan  
5. Cari  
x. Keluar  
Pilih : 1  
[Menambah Todo]  
Judul (x Jika Batal) : Belajar Dasar HTML  
  
Daftar Todo:  
1 | Belajar Java OOP | Belum Selesai  
2 | Belajar Dasar HTML | Belum Selesai  
Menu:  
1. Tambah  
2. Ubah  
3. Hapus
```

```
4. Urutan
5. Cari
x. Keluar
Pilih : 1
[Menambah Todo]
Judul (x Jika Batal) : Membaca kembali materi SISOP w4

Daftar Todo:
1 | Belajar Java OOP | Belum Selesai
2 | Belajar Dasar HTML | Belum Selesai
3 | Membaca kembali materi SISOP w4 | Belum Selesai
Menu:
1. Tambah
2. Ubah
3. Hapus
4. Urutan
5. Cari
x. Keluar
Pilih : 1
[Menambah Todo]
Judul (x Jika Batal) : Menyelesaikan tugas PROBSTAT

Daftar Todo:
1 | Belajar Java OOP | Belum Selesai
2 | Belajar Dasar HTML | Belum Selesai
3 | Membaca kembali materi SISOP w4 | Belum Selesai
4 | Menyelesaikan tugas PROBSTAT | Belum Selesai
Menu:
1. Tambah
2. Ubah
3. Hapus
4. Urutan
5. Cari
x. Keluar
Pilih : 4
[Mengurutkan Todo]
Menu:
1. Berdasarkan ID (Ascending)
2. Berdasarkan ID (Descending)
3. Berdasarkan Title (Ascending)
4. Berdasarkan Title (Descending)
x. Batal
Pilih : 4

Daftar Todo:
4 | Menyelesaikan tugas PROBSTAT | Belum Selesai
3 | Membaca kembali materi SISOP w4 | Belum Selesai
1 | Belajar Java OOP | Belum Selesai
2 | Belajar Dasar HTML | Belum Selesai
Menu:
1. Tambah
2. Ubah
3. Hapus
4. Urutan
5. Cari
```



```
x. Keluar
Pilih : 4
[Mengurutkan Todo]
Menu:
1. Berdasarkan ID (Ascending)
2. Berdasarkan ID (Descending)
3. Berdasarkan Title (Ascending)
4. Berdasarkan Title (Descending)
x. Batal
Pilih : 3

Daftar Todo:
2 | Belajar Dasar HTML | Belum Selesai
1 | Belajar Java OOP | Belum Selesai
3 | Membaca kembali materi SISOP w4 | Belum Selesai
4 | Menyelesaikan tugas PROBSTAT | Belum Selesai
Menu:
1. Tambah
2. Ubah
3. Hapus
4. Urutan
5. Cari
x. Keluar
Pilih : 4
[Mengurutkan Todo]
Menu:
1. Berdasarkan ID (Ascending)
2. Berdasarkan ID (Descending)
3. Berdasarkan Title (Ascending)
4. Berdasarkan Title (Descending)
x. Batal
Pilih : 2

Daftar Todo:
4 | Menyelesaikan tugas PROBSTAT | Belum Selesai
3 | Membaca kembali materi SISOP w4 | Belum Selesai
2 | Belajar Dasar HTML | Belum Selesai
1 | Belajar Java OOP | Belum Selesai
Menu:
1. Tambah
2. Ubah
3. Hapus
4. Urutan
5. Cari
x. Keluar
Pilih : 4
[Mengurutkan Todo]
Menu:
1. Berdasarkan ID (Ascending)
2. Berdasarkan ID (Descending)
3. Berdasarkan Title (Ascending)
4. Berdasarkan Title (Descending)
x. Batal
Pilih : 1
```

```
Daftar Todo:
1 | Belajar Java OOP | Belum Selesai
2 | Belajar Dasar HTML | Belum Selesai
3 | Membaca kembali materi SISOP w4 | Belum Selesai
4 | Menyelesaikan tugas PROBSTAT | Belum Selesai
Menu:
1. Tambah
2. Ubah
3. Hapus
4. Urutan
5. Cari
x. Keluar
Pilih : x
```

(!) Latar belakang oranye merupakan inputan.

5. Fitur Searching

Silahkan menambahkan fitur searching pada aplikasi todo. Berikut merupakan contoh program saat dijalankan:

```
Daftar Todo:
- Data todo belum tersedia!
Menu:
1. Tambah
2. Ubah
3. Hapus
4. Urutan
5. Cari
x. Keluar
Pilih : 1
[Menambah Todo]
Judul (x Jika Batal) : Membaca buku
```

```
Daftar Todo:
1 | Membaca buku | Belum Selesai
Menu:
1. Tambah
2. Ubah
3. Hapus
4. Urutan
5. Cari
x. Keluar
Pilih : 1
[Menambah Todo]
Judul (x Jika Batal) : Bermain game
```

```
Daftar Todo:
1 | Membaca buku | Belum Selesai
2 | Bermain game | Belum Selesai
Menu:
1. Tambah
2. Ubah
3. Hapus
4. Urutan
```

```
5. Cari
x. Keluar
Pilih : 1
[Menambah Todo]
Judul (x Jika Batal) : Menyelesaikan tugas

Daftar Todo:
1 | Membaca buku | Belum Selesai
2 | Bermain game | Belum Selesai
3 | Menyelesaikan tugas | Belum Selesai
Menu:
1. Tambah
2. Ubah
3. Hapus
4. Urutan
5. Cari
x. Keluar
Pilih : 5
[Mencari Todo]
Cari (x Jika Batal) : ga
Hasil pencarian: ga
Ditemukan 2 Todo:
2 | Bermain game | Belum Selesai
3 | Menyelesaikan tugas | Belum Selesai

Daftar Todo:
1 | Membaca buku | Belum Selesai
2 | Bermain game | Belum Selesai
3 | Menyelesaikan tugas | Belum Selesai
Menu:
1. Tambah
2. Ubah
3. Hapus
4. Urutan
5. Cari
x. Keluar
Pilih : x
```

(!) Latar belakang oranye merupakan inputan.