

Nama : Noor Rizki Ramadhan
NIM : 2243912
Kelas : TI/PA

Mata Kuliah : Pemrograman Web
Jurusan : Teknik Informatika
Tugas : PHP OOP TODO

- Entity

```
1  /* Ini adalah kode PHP yang mendefinisikan kelas bernama 'Todolist' yang di dalamnya terdapat namespace
2  'Entity'. Kelas memiliki sebuah Properti pribadi '$todo' dan tiga metode: '__construct()', 'getTodo()', dan 'setTodo()'. '__construct ()'
3  adalah sebuah konstruktor yang menginisialisasi properti '$todo' dengan opsional parameter string.
4  Metode 'getTodo ()' mengembalikan nilai properti '$todo', dan 'setTodo ()' menetapkan nilai properti '$todo' dengan parameter string. Kelas ini bisa
5  Digunakan untuk membuat objek yang mewakili item daftar yang harus dilakukan.*/
6
7  namespace Entity {
8
9      class Todolist
10     {
11
12         private string $todo;
13
14         public function __construct(string $todo = "")
15         {
16             $this->todo = $todo;
17         }
18
19         public function getTodo(): string
20         {
21             return $this->todo;
22         }
23
24         public function setTodo(string $todo): void
25         {
26             $this->todo = $todo;
27         }
28     }
29 }
30
```

- Helper

```
1  /* Kode ini mendefinisikan sebuah namespace php yang disebut "Helper" dan di dalamnya terdapat sebuah kelas yang disebut "InputHelper".
2  kelas tersebut memiliki metode statis yang disebut "input" yang mengambil argumen string dan meminta pengguna untuk memasukkan
3  sebuah string yang disediakan sebagai prompt. Metode kemudian mengembalikan input pengguna dengan apapun dan menghapus whitespace.*/
4
5  namespace Helper {
6
7      class InputHelper
8      {
9
10         static function input(string $info): string
11         {
12             echo "$info : ";
13             $result = fgets(STDIN);
14             return trim($result);
15         }
16     }
17 }
```

- Repository

```
1  /* Kode ini mendefinisikan namespace 'Repository' dan menyatakan sebuah interface 'TodolistRepository' dengan tiga
2  Metode 'save()', 'remove()', dan 'findAll()'. Itu juga mendefinisikan kelas 'todolistrepositoryimpl'
3  class tersebut mengimplementasikan antarmuka 'todolistrepository' dan menyediakan implementasi untuk tiga metode.
4  Kelas memiliki properti publik '$todolist' yang merupakan array dan digunakan untuk menyimpan daftar todo item.
5  Metode 'save()' menambahkan item baru ke daftar TODO, metode
6  'remove()' menghapus item dari daftar TODO berdasarkan indeksinya,
7  dan metode 'findAll()' mengembalikan semua item di todo
8  yang ada di daftar. Pernyataan 'use' mengimpor kelas
9  'Entity\Todolist' ke namespace.*/
10
11
12 namespace Repository {
13     use Entity\Todolist;
14
15     interface TodolistRepository
16     {
17
18         function save(Todolist $todolist): void;
19
20         function remove(int $number): bool;
21
22         function findAll(): array;
23     }
24
25     class TodolistRepositoryImpl implements TodolistRepository
26     {
27
28         public array $todolist = array();
29
30         function save(Todolist $todolist): void
31         {
32             $number = sizeof($this->todolist) + 1;
33             $this->todolist[$number] = $todolist;
34         }
35
36         function remove(int $number): bool
37         {
38             if ($number > sizeof($this->todolist)) {
39                 return false;
40             }
41
42             for ($i = $number; $i < sizeof($this->todolist); $i++) {
43                 $this->todolist[$i] = $this->todolist[$i + 1];
44             }
45
46             unset($this->todolist[sizeof($this->todolist)]);
47
48             return true;
49         }
50
51         function findAll(): array
52         {
53             return $this->todolist;
54         }
55     }
56 }
57
```

- Service

```
1  /* Kode ini mendefinisikan namespace 'Service' yang berisi interface
2  'TodolistService' dan kelas 'todolistServiceImpl'
3  yang mengimplementasikan sebuah interface. Interface mendefinisikan tiga metode: 'showTodolist()', 'addTodolist(string)', dan
4  'remove(int)'. Kelas mengimplementasikan sebuah interface. Interface mendefinisikan tiga metode: 'showTodolist()', 'addTodolist(string)', dan 'removeTodolist(int)'.
5  Kelas 'TodolistServiceImpl' memiliki konstruktor yang mengambil contoh 'todolistrepository'
6  sebagai sebuah Parameter dan mengimplementasikan metode yang didefinisikan dalam interface 'TodolistService'.
7  Itu 'showTodolist()' metode mengambil semua item dalam todolist dari repository dan menampilkan mereka di konsol. Metode
8  'addTodolist(string)' membuat sebuah objek 'todolist' baru dengan
9  Diberikan todo dan menyimpannya ke repository. Metode
10 'removetodolist (int)' menghapus sebuah item dengan nomor yang diberikan dari repository.*/
11
12 namespace Service {
13
14     use Entity\Todolist;
15     use Repository\TodolistRepository;
16
17     interface TodolistService
18     {
19
20         function showTodolist(): void;
21
22         function addTodolist(string $todo): void;
23
24         function removeTodolist(int $number): void;
25     }
26
27     class TodolistServiceImpl implements TodolistService
28     {
29
30         private TodolistRepository $todolistRepository;
31
32         public function __construct(TodolistRepository $todolistRepository)
33         {
34             $this->todolistRepository = $todolistRepository;
35         }
36
37         function showTodolist(): void
38         {
39             echo "TODO LIST" . PHP_EOL;
40             $todolist = $this->todolistRepository->findAll();
41             foreach ($todolist as $number => $value) {
42                 echo "$number. " . $value->getTodo() . PHP_EOL;
43             }
44         }
45
46         function addTodolist(string $todo): void
47         {
48             $todolist = new Todolist($todo);
49             $this->todolistRepository->save($todolist);
50             echo "SUKSES MENAMBAH TODO LIST" . PHP_EOL;
51         }
52
53         function removeTodolist(int $number): void
54         {
55             if ($this->todolistRepository->remove($number)) {
56                 echo "SUKSES MENGHAPUS TODO LIST" . PHP_EOL;
57             } else {
58                 echo "GAGAL MENGHAPUS TODO LIST" . PHP_EOL;
59             }
60         }
61     }
62 }
```

- View

```
1  /* Kode ini mendefinisikan sebuah namespace 'View' dan mendefinisikan sebuah kelas 'todolistview' di dalamnya. Kelas ini memiliki
2  Tiga metode: 'showtodolist()', 'addtodolist()', dan 'removetodolist()'. Konstruktor dari Kelas mengambil objek 'todolistService'
3  sebagai parameter dan menugaskannya ke properti pribadi '$todolistService'.
4  Kelas ini juga mengimpor dua kelas 'todolistService' dan 'inputhelper' menggunakan 'Use' Keyword.*/
5
6  namespace View {
7
8      use Service\TodolistService;
9      use Helper\InputHelper;
10
11      class TodolistView
12      {
13
14          private TodolistService $todolistService;
15
16          public function __construct(TodolistService $todolistService)
17          {
18              $this->todolistService = $todolistService;
19          }
20
21          function showTodolist(): void
22          {
23              while (true) {
24                  $this->todolistService->showTodolist();
25
26                  echo "MENU" . PHP_EOL;
27                  echo "1. Tambah Todo" . PHP_EOL;
28                  echo "2. Hapus Todo" . PHP_EOL;
29                  echo "x. Keluar" . PHP_EOL;
30
31                  $pilihan = InputHelper::input("Pilih");
32
33                  if ($pilihan == "1") {
34                      $this->addTodolist();
35                  } else if ($pilihan == "2") {
36                      $this->removeTodolist();
37                  } else if ($pilihan == "x") {
38                      break;
39                  } else {
40                      echo "Pilihan tidak dimengerti" . PHP_EOL;
41                  }
42              }
43
44              echo "Sampai Jumpa Lagi" . PHP_EOL;
45          }
46
47          function addTodolist(): void
48          {
49              echo "MENAMBAH TODO" . PHP_EOL;
50
51              $todo = InputHelper::input("Todo (x untuk batal)");
52
53              if ($todo == "x") {
54                  echo "Batal menambah todo" . PHP_EOL;
55              } else {
56                  $this->todolistService->addTodolist($todo);
57              }
58          }
59
60          function removeTodolist(): void
61          {
62              echo "MENGHAPUS TODO" . PHP_EOL;
63
64              $pilihan = InputHelper::input("Nomor (x untuk batalkan)");
65
66              if ($pilihan == "x") {
67                  echo "Batal menghapus todo" . PHP_EOL;
68              } else {
69                  $this->todolistService->removeTodolist($pilihan);
70              }
71          }
72      }
73 }
```

- App

```
1  /* Kode ini mengimpor dan membutuhkan file dan kelas yang diperlukan untuk dijalankan oleh aplikasi Todolist.
2  Kemudian menciptakan contoh TodolistRepositoryImpl, TodolistServiceImpl, dan TodolistView
3  kelas, dan akhirnya instance TodolistService ke TodolistView Constructor. Akhirnya, memanggil Metode showtodolist dari instance TodolistView untuk menampilkan aplikasi Todolist.*/
4  require_once __DIR__ . '/Entity/Todolist.php';
5  require_once __DIR__ . '/Helper/InputHelper.php';
6  require_once __DIR__ . '/Repository/TodolistRepository.php';
7  require_once __DIR__ . '/Service/TodolistService.php';
8  require_once __DIR__ . '/View/TodolistView.php';
9
10 use Repository\TodolistRepositoryImpl;
11 use Service\TodolistServiceImpl;
12 use View\TodolistView;
13
14 echo "Aplikasi Todolist" . PHP_EOL;
15
16 $todolistRepository = new TodolistRepositoryImpl();
17 $todolistService = new TodolistServiceImpl($todolistRepository);
18 $todolistView = new TodolistView($todolistService);
19
20 $todolistView->showTodolist();
```