

NIM: 231232028

Nama: Falmesino Abdul Hamid

URL Source Code (GitHub): <https://github.com/falmesino/sda-praktikum-6>

1 - Pengelolaan Data Besar Menggunakan Binary Search Tree

The screenshot shows a code editor with three tabs open, all containing the same file: `1-pengelolaan-data-besar-binary-search-tree.php`. The code implements a binary search tree (BST) for managing logs based on timestamp.

```
<?php
/*
 * ./1-pengelolaan-data-besar-binary-search-tree.php
 * Falmesino Abdul Hamid - 231232028
 */

// Kelas Node untuk merepresentasikan setiap node di BST
class LogNode {
    public $timestamp;
    public $message;
    public $left;
    public $right;

    public function __construct($timestamp, $message) {
        $this->timestamp = $timestamp;
        $this->message = $message;
        $this->left = null;
        $this->right = null;
    }
}

// Kelas untuk struktur Binary Search Tree
class LogTree {
    public $root;

    public function __construct() {
        $this->root = null;
    }

    // Fungsi untuk menambahkan data log ke dalam BST
    public function addLog($timestamp, $message) {
        $newNode = new LogNode($timestamp, $message);
        if ($this->root == null) {
            $this->root = $newNode;
        } else {
            $this->insertNode($this->root, $newNode);
        }
    }

    // Fungsi rekursif untuk menyiapkan node baru di tree
    private function insertNode($node, $newNode) {
        if ($newNode->timestamp < $node->timestamp) {
            if ($node->left === null) {
                $node->left = $newNode;
            } else {
                $this->insertNode($node->left, $newNode);
            }
        } else {
            if ($node->right === null) {
                $node->right = $newNode;
            } else {
                $this->insertNode($node->right, $newNode);
            }
        }
    }
}

// Fungsi untuk mencari log berdasarkan timestamp
public function searchLog($timestamp) {
    return $this->searchNode($this->root, $timestamp);
}

// Fungsi rekursif untuk mencari node berdasarkan timestamp
private function searchNode($node, $timestamp) {
    if ($node === null) {
        return null; // Data tidak ditemukan
    }

    if ($timestamp === $node->timestamp) {
        return $node; // Data ditemukan
    } elseif ($timestamp < $node->timestamp) {
        return $this->searchNode($node->left, $timestamp);
    } else {
        return $this->searchNode($node->right, $timestamp);
    }
}
```

The screenshot shows a terminal window with the following session:

```
sda-tugas-6 pwd
/Users/beitmacbook/Development/valet/sda-tugas-6
sda-tugas-6 ls
1-pengelolaan-data-besar-binary-search-tree.php 2-pengelolaan-data-terindeks.php
sda-tugas-6 valet php 1-pengelolaan-data-besar-binary-search-tree.php
Log found: Server 1 started at 1618300500.
sda-tugas-6
```

2 - Pengelolaan Data Terindeks

```
1 <?php
2
3 /**
4 * ./2-pengelolaan-data-terindeks.php
5 * Falenesino Abdul Hamid - 231232028
6 */
7
8 // Kelas Node untuk merepresentasikan setiap node di BST
9 class SensorNode {
10     public $sensorID;
11     public $temperature;
12     public $left;
13     public $right;
14
15     public function __construct($sensorID, $temperature) {
16         $this->sensorID = $sensorID;
17         $this->temperature = $temperature;
18         $this->left = null;
19         $this->right = null;
20     }
21 }
22
23 // Kelas untuk Binary Search Tree untuk data sensor
24 class SensorDataTree {
25     public $root;
26
27     public function __construct() {
28         $this->root = null;
29     }
30
31 // Fungsi untuk menambahkan data sensor ke dalam BST
32     public function addSensorData($sensorID, $temperature) {
33         $newNode = new SensorNode($sensorID, $temperature);
34
35         if ($this->root == null) {
36             $this->root = $newNode;
37         } else {
38             $this->insertNode($this->root, $newNode);
39         }
40
41 // Fungsi rekursif untuk menyediakan node baru di tree
42     private function insertNode($node, $newNode) {
43         if ($newNode->temperature < $node->temperature) {
44             if ($node->left === null) {
45                 $node->left = $newNode;
46             } else {
47                 $this->insertNode($node->left, $newNode);
48             }
49         } else {
50             if ($node->right === null) {
51                 $node->right = $newNode;
52             } else {
53                 $this->insertNode($node->right, $newNode);
54             }
55         }
56     }
57
58 // Fungsi untuk mencari data sensor berdasarkan
59 // temperatur
60     public function searchByTemperature($temperature) {
61         return $this->searchNode($this->root, $temperature);
62     }
63
64 // Fungsi rekursif untuk mencari node berdasarkan
65 // temperatur
66     private function searchNode($node, $temperature) {
67         if ($node === null) {
68             return null; // Data tidak ditemukan
69         }
70         if ($temperature === $node->temperature) {
71             return $node; // Data ditemukan
72         } elseif ($temperature < $node->temperature) {
73             return $this->searchNode($node->left, $temperature);
74         } else {
75             return $this->searchNode($node->right, $temperature);
76         }
77     }
78
79 // Contoh penggunaan
80 $sensorTree = new SensorDataTree();
81
82 // Menambahkan beberapa data sensor
83 $sensorTree->addSensorData(101, 22.5);
84 $sensorTree->addSensorData(102, 19.8);
85 $sensorTree->addSensorData(103, 25.1);
86 $sensorTree->addSensorData(104, 23.4);
87
88 // Mencari data sensor berdasarkan temperatur
89 $searchTemp = 23.4;
90 $foundNode = $sensorTree->searchByTemperature($searchTemp);
91 if ($foundNode != null) {
92     echo "Sensor with ID " . $foundNode->sensorID . " has a
93     temperature of " . $foundNode->temperature . "C.\n";
94 } else {
95     echo "No sensor data found for temperature " .
96     $searchTemp . "C.\n";
97 }
```

```
[+ sda-tugas-6 pwd
/Users/beitmackbook/Development/valet/sda-tugas-6
[+ sda-tugas-6 ls
1-pengelolaan-data-besar-binary-search-tree.php 2-pengelolaan-data-terindeks.php
[+ sda-tugas-6 valet php 2-pengelolaan-data-terindeks.php
Sensor with ID 104 has a temperature of 23.4C.
[+ sda-tugas-6 ]
```

3 - Penyimpanan Hasil Prediksi Model

```
<?php
/*
 * ./3-penyimpanan-hasil-prediksi-model.php
 * Fatenisa Abdul Hamid - 231232028
 */
// Kelas Node untuk setiap elemen dalam BST
class DataNode {
    public $id;
    public $left;
    public $right;
}

public function __construct($id) {
    $this->id = $id;
    $this->left = null;
    $this->right = null;
}

// Kelas untuk mengimplementasikan struktur Binary Search Tree
class DataIndexTree {
    public $root;
    public function __construct(){
        $this->root = null;
    }

    // Fungsi untuk menambahkan ID data ke dalam tree
    public function addData($id) {
        $newNode = new DataNode($id);
        if ($this->root === null) {
            $this->root = $newNode;
        } else {
            $this->insertNode($this->root, $newNode);
        }
    }

    // Fungsi rekursif untuk menyiapkan node baru di tree
    private function insertNode($node, $newNode) {
        if ($newNode->id < $node->id) {
            if ($node->left === null) {
                $node->left = $newNode;
            } else {
                $this->insertNode($node->left, $newNode);
            }
        } else {
            if ($node->right === null) {
                $node->right = $newNode;
            } else {
                $this->insertNode($node->right, $newNode);
            }
        }
    }
}

// Fungsi untuk mencari ID data dalam tree
public function searchData($id) {
    return $this->searchNode($this->root, $id);
}

// Fungsi rekursif untuk mencari node tertentu di dalam tree
private function searchNode($node, $id) {
    if ($node === null) {
        return false; // Data tidak ditemukan
    }
    if ($id === $node->id) {
        return true; // Data ditemukan
    } elseif ($id < $node->id) {
        return $this->searchNode($node->left, $id);
    } else {
        return $this->searchNode($node->right, $id);
    }
}

// Contoh penggunaan
$searchID = 150;
if ($dataTree->searchData($searchID)) {
    echo "Data with ID $searchID found in the index.\n";
} else {
    echo "Data with ID $searchID not found.\n";
}

```

```
[+ sda-tugas-6 pwd
/Users/beitmacbook/Development/valent/sda-tugas-6
[+ sda-tugas-6 ls
1-pengelolaan-data-besar-binary-search-tree.php 3-penyimpanan-hasil-prediksi-model.php
2-pengelolaan-data-terindeks.php
screenshots
[+ sda-tugas-6 valent php 3-penyimpanan-hasil-prediksi-model.php
Data with ID 150 found in the index.
sda-tugas-6 ]]
```

1-pengelolaan-data-besar-binary-search-tree.php

```

1 <?php
2
3 /**
4 * ./1-pengelolaan-data-besar-binary-search-tree.php.php
5 * Falmesino Abdul Hamid - 231232028
6 */
7
8 // Kelas Node untuk merepresentasikan setiap node di BST
9 class LogNode {
10     public $timestamp;
11     public $message;
12     public $left;
13     public $right;
14
15     public function __construct($timestamp, $message) {
16         $this->timestamp = $timestamp;
17         $this->message = $message;
18         $this->left = null;
19         $this->right = null;
20     }
21 }
22
23 // Kelas untuk struktur Binary Search Tree
24 class LogTree {
25     public $root;
26
27     public function __construct() {
28         $this->root = null;
29     }
30
31     // Fungsi untuk menambahkan data log ke dalam BST
32     public function addLog($timestamp, $message) {
33         $newNode = new LogNode($timestamp, $message);
34         if ($this->root === null) {
35             $this->root = $newNode;
36         } else {
37             $this->insertNode($this->root, $newNode);
38         }
39     }
40
41     // Fungsi rekursif untuk menyisipkan node baru di tree
42     private function insertNode($node, $newNode) {
43         if ($newNode->timestamp < $node->timestamp) {
44             if ($node->left === null) {
45                 $node->left = $newNode;
46             } else {
47                 $this->insertNode($node->left, $newNode);
48             }
49         } else {
50             if ($node->right === null) {
51                 $node->right = $newNode;
52             } else {
53                 $this->insertNode($node->right, $newNode);
54             }
55         }
56     }
57
58     // Fungsi untuk mencari log berdasarkan timestamp
59     public function searchLog($timestamp) {
60         return $this->searchNode($this->root, $timestamp);
61     }
62
63     // Fungsi rekursif untuk mencari node berdasarkan timestamp
64     private function searchNode($node, $timestamp) {
65         if ($node === null) {
66             return null; // Data tidak ditemukan
67         }
68         if ($timestamp === $node->timestamp) {
69             return $node; // Data ditemukan
70         } elseif ($timestamp < $node->timestamp) {
71             return $this->searchNode($node->left, $timestamp);
72         } else {
73             return $this->searchNode($node->right, $timestamp);
74         }
75     }
76
77     // Contoh Penggunaan
78     $logTree = new LogTree();
79
80     // Menambahkan beberapa data log
81     $logTree->addLog(161830000, "Server 1 started");
82     $logTree->addLog(1618300500, "Server 1 started");
83     $logTree->addLog(1618301000, "Database connection established");
84
85     // Mencari log berdasarkan timestamp
86     $searchTime = 1618300500; // 161300500
87     $foundLog = $logTree->searchLog($searchTime);
88
89     if ($foundLog !== null) {
90         echo "Log found: " . $foundLog->message . " at " .
91             $foundLog->timestamp . ".\n";
92     } else {
93         echo "No log found for timestamp " . $searchTime . ".\n";
94     }
95
96
97 ?>
```

```
● ○ ● sda-tugas-6 — beitmacbook@BeITs-MacBook-Air — ..t/sda-tugas-6 — -zsh — 124x37
[→ sda-tugas-6 pwd
/Users/beitmacbook/Development/valet/sda-tugas-6
[→ sda-tugas-6 ls
1-pengelolaan-data-besar-binary-search-tree.php 2-pengelolaan-data-terindeks.php
[→ sda-tugas-6 valet php 1-pengelolaan-data-besar-binary-search-tree.php
Log found: Server 1 started at 1618300500.
→ sda-tugas-6
```

2-pengelolaan-data-terindeks.php

```

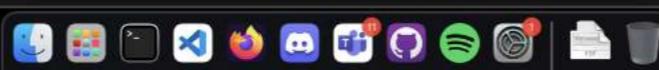
1 <?php
2
3 /**
4 * ./2-pengelolaan-data-terindeks.php
5 * Falmesino Abdul Hamid - 231232028
6 */
7
8 // Kelas Node untuk merepresentasikan setiap node di BST
9 class SensorNode {
10     public $sensorID;
11     public $temperature;
12     public $left;
13     public $right;
14
15     public function __construct($sensorID, $temperature) {
16         $this->sensorID = $sensorID;
17         $this->temperature = $temperature;
18         $this->left = null;
19         $this->right = null;
20     }
21 }
22
23 // Kelas untuk Binary Search Tree untuk data sensor
24 class SensorDataTree {
25     public $root;
26
27     public function __construct() {
28         $this->root = null;
29     }
30
31     // Fungsi untuk menambahkan data sensor ke dalam BST
32     public function addSensorData($sensorID, $temperature) {
33         $newNode = new SensorNode($sensorID, $temperature);
34         if ($this->root === null) {
35             $this->root = $newNode;
36         } else {
37             $this->insertNode($this->root, $newNode);
38         }
39     }
40
41     // Fungsi rekursif untuk menyisipkan node baru di tree
42     private function insertNode($node, $newNode) {
43         if ($newNode->temperature < $node->temperature) {
44             if ($node->left === null) {
45                 $node->left = $newNode;
46             } else {
47                 $this->insertNode($node->left, $newNode);
48             }
49         } else {
50             if ($node->right === null) {
51                 $node->right = $newNode;
52             } else {
53                 $this->insertNode($node->right, $newNode);
54             }
55         }
56     }
57
58     // Fungsi untuk mencari data sensor berdasarkan
59     // temperaature
60     public function searchByTemperature($temperature) {
61         return $this->searchNode($this->root, $temperature);
62     }
63
64     // Fungsi rekursif untuk mencari node berdasarkan
65     // temperature
66     private function searchNode($node, $temperature) {
67         if ($node === null) {
68             return null; // Data tidak ditemukan
69         }
70         if ($temperature === $node->temperature) {
71             return $node; // Data ditemukan
72         } elseif ($temperature < $node->temperature) {
73             return $this->searchNode($node->left, $temperature);
74         } else {
75             return $this->searchNode($node->right, $temperature);
76         }
77     }
78
79     // Contoh penggunaan
80     $sensorTree = new SensorDataTree();
81
82     // Menambahkan beberapa data sensor
83     $sensorTree->addSensorData(101, 22.5);
84     $sensorTree->addSensorData(102, 19.8);
85     $sensorTree->addSensorData(103, 25.1);
86     $sensorTree->addSensorData(104, 23.4);
87
88     // Mencari data sensor berdasarkan temperature
89     $searchTemp = 23.4;
90     $foundNode = $sensorTree->searchByTemperature($searchTemp);
91     if ($foundNode !== null) {
92         echo "Sensor with ID " . $foundNode->sensorID . " has a
93         temperature of " . $foundNode->temperature . "C.\n";
94     } else {
95         echo "No sensor data found for temperature " .
96         $searchTemp . "C.\n";
97     }
98
99 ?>
```

```
[→ sda-tugas-6 pwd  
/Users/beitmacbook/Development/valet/sda-tugas-6  
[→ sda-tugas-6 ls  
1-pengelolaan-data-besar-binary-search-tree.php 2-pengelolaan-data-terindeks.php  
[→ sda-tugas-6 valet php 2-pengelolaan-data-terindeks.php  
Sensor with ID 104 has a temperature of 23.4C.  
→ sda-tugas-6 ]
```

3-penyimpanan-hasil-prediksi-model.php × ... 3-penyimpanan-hasil-prediksi-model.php × ... 3-penyimpanan-hasil-prediksi-model.php × ...

```

1 <?php
2
3 /**
4 * ./3-penyimpanan-hasil-prediksi-model.php
5 * Falmesino Abdul Hamid - 231232028
6 */
7
8 // Kelas Node untuk setiap element dalam BST
9 class DataNode {
10     public $id;
11     public $left;
12     public $right;
13
14     public function __construct($id) {
15         $this->id = $id;
16         $this->left = null;
17         $this->right = null;
18     }
19 }
20
21 // Kelas untuk mengimplementasikan struktur Binary Search Tree
22 class DataIndexTree {
23     public $root;
24
25     public function __construct(){
26         $this->root = null;
27     }
28
29     // Fungsi untuk menambahkan ID data ke dalam tree
30     public function addData($id) {
31         $newNode = new DataNode($id);
32         if ($this->root === null) {
33             $this->root = $newNode;
34         } else {
35             $this->insertNode($this->root, $newNode);
36         }
37     }
38
39     // Fungsi rekursif untuk menyisipkan node baru di tree
40     private function insertNode($node, $newNode) {
41         if ($newNode->id < $node->id) {
42             if ($node->left === null) {
43                 $node->left = $newNode;
44             } else {
45                 $this->insertNode($node->left, $newNode);
46             }
47         } else {
48             if ($node->right === null) {
49                 $node->right = $newNode;
50             } else {
51                 $this->insertNode($node->right, $newNode);
52             }
53         }
54     }
55
56     // Fungsi untuk mencari ID data dalam tree
57     public function searchData($id) {
58         return $this->searchNode($this->root, $id);
59     }
60
61     // Fungsi rekursif untuk mencari node tertentu di dalam tree
62     private function searchNode($node, $id) {
63         if ($node === null) {
64             return false; // Data tidak ditemukan
65         }
66         if ($id === $node->id) {
67             return true; // Data ditemukan
68         } elseif ($id < $node->id) {
69             return $this->searchNode($node->left, $id);
70         } else {
71             return $this->searchNode($node->right, $id);
72         }
73     }
74
75     // Contoh penggunaan
76     $dataTree = new DataIndexTree();
77
78     // Menambahkan beberapa ID data
79     $dataTree->addData(101);
80     $dataTree->addData(205);
81     $dataTree->addData(150);
82     $dataTree->addData(89);
83
84     // Mencari ID data
85     $searchID = 150;
86     if ($dataTree->searchData($searchID)) {
87         echo "Data with ID $searchID found in the index.\n";
88     } else {
89         echo "Data with ID $searchID not found.\n";
90     }
91
92 ?>
```



```
● ○ ● sda-tugas-6 — beitmacbook@BeITs-MacBook-Air — ..t/sda-tugas-6 — -zsh — 124x37
[→ sda-tugas-6 pwd
/Users/beitmacbook/Development/valet/sda-tugas-6
[→ sda-tugas-6 ls
1-pengelolaan-data-besar-binary-search-tree.php 3-penyimpanan-hasil-prediksi-model.php
2-pengelolaan-data-terindeks.php           screenshots
[→ sda-tugas-6 valet php 3-penyimpanan-hasil-prediksi-model.php
Data with ID 150 found in the index.
→ sda-tugas-6
```