PROJECT REPORT
Implementation of B-Tree in Java Programming
Romy Budiansyah
06.02.0082
2011

FACULTY OF COMPUTER SCIENCE
SOEGIJAPRANATA CATHOLIC UNIVERSITY
Jl. Pawiyatan Luhur IV/1, Bendan Duwur, SEMARANG 50234
Telp. 024-8441555 (hunting) Web: http://www.unika.ac.id
Email: ikom@unika.ac.id
PROJECT REPORT

Implementation of B-Tree in Java Programming

This project report already approved and ratified by Dean of Faculty Computer Science and Supervisor on

With the approval,

Examiner, Examiner,
Suyanto EA, Ir., M.Sc Rosita Herawati, ST, MIT

Supervisor, Dean of Faculty of Computer Science,

Suyanto EA, Ir., M.Sc Hironimus Marlon Leong, S.Kom, M.Kom
STATEMENT of ORIGINALITY

Hereby signed:

Name: Romy Budiansyah
ID: 06.02.0082

Here by certify that this project was made by my self and not copy or plagiarizes from other people, except that in writing expressed to the other article.
If it is proven that this project was plagiarizes or copy the other, I'm ready to accept a sanction.

Semarang,
Romy Budiansyah
06.02.0082
FOREWORD

Thanks a lot of God because it has been able to be completed my final project, with title : Implementation of B-Tree in Java Programming. And in this opportunity, I would like to thanks :

- My Lord, that give me power to finish this project.
- My Lovely sister, Uci Novirudiana for her support and pray.
- My parents and my big family for their support, love, and pray.
- Suyanto EA., Ir, M.Sc, as my supervisor for helping, guiding and giving me ideas and advice in finishing this project.
- All the lecturers of Faculty of Computer Science for teaching me and give me knowledge while I'm studied in Faculty of Computer Science.
- All of my love friends which help and support me to finish this project, and also for people who have helped me in prayers and support.

Finally, I would like to apologize if the project is still many shortcomings. I look forward to suggestions and criticism.

Semarang,

Romy Budiansyah
06.02.0082
ABSTRACT

B-Tree is a tree data structure of the most common used in databases and filesystems, that keeps data sorted and balanced. The idea of B-Tree is an internal node can have a number of child nodes within some pre-defined range. When data is inserted or removed from a node, its number of child nodes changes. In order to maintain the pre-defined range, the internal nodes may be joined or split. Because a range of child nodes is permitted, B-Tree does not need re-balancing as frequently as other self-balancing search tree. However, B-Tree may waste some space, since nodes are not entirely full. The lower and upper bounds on the number of child nodes are typically fixed for a particular implementation.

B-Tree kept balanced by requiring that all leaf nodes at the same height. The height will increase slowly as elements are added to the tree, but an increase in the overall height or depth is infrequent. Because B-Tree is designed to have branches in large numbers and contains a number of keys on each node so that relative tree height small. By maximizing the number of child nodes within each internal node, the height of tree decreases, balancing occurs less than often, and efficiency increases.

Keywords : B-Tree, data structure, node, self-balancing search tree
# Table of Content

Approval and Ratification Page .................................................. i  
Statement of Originality .......................................................... ii  
Foreword ....................................................................................... iii  
Abstract ....................................................................................... iv  
Table of Content ........................................................................... v  
Table of Figure ............................................................................... vii  
Table of Table ................................................................................ ix  

## Chapter I Introduction
1.1. Background ............................................................................. 1  
1.2. Scope ....................................................................................... 1  
1.3. Objectives ................................................................................ 1  

## Chapter II Literature Study
2.1. Data Structure ........................................................................... 2  
2.1.1. Definition of B-Tree ............................................................. 2  
2.1.2. Height of B-Tree ................................................................. 3  
2.1.3. B-Tree Steps Description .................................................. 5  

## Chapter III Planning
3.1. Research Methodologies ......................................................... 8  
3.2. Project Management ............................................................... 8  

## Chapter IV Analysis and Design
4.1. Analysis ................................................................................... 9  
4.1.1. Use Case Diagram .............................................................. 9  
4.2. Design ..................................................................................... 10  
4.2.1. Class Diagram .................................................................... 10  
4.2.2. Details of Each Class Diagram .......................................... 11
Chapter V Implementation and Testing

5.1. Implementation .......................................................... 13
5.2. Testing ................................................................. 21

Chapter VI Conclusion and Further Research

6.1. Conclusion .......................................................... 25
6.2. Further Research .................................................. 25

References ........................................................................ 26
# Table of Figure

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1</td>
<td>Height of B-Tree</td>
<td>4</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Full Node Root</td>
<td>6</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>Split the Full Node Root</td>
<td>6</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>Full Node Child</td>
<td>6</td>
</tr>
<tr>
<td>Figure 2.5</td>
<td>Split the Full Node Child</td>
<td>28</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Use Case Diagram</td>
<td>29</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Class Diagram</td>
<td>29</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>BTreeApplet Class</td>
<td>11</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>BTree Class</td>
<td>11</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Node Class</td>
<td>12</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>ReturnValue Class</td>
<td>12</td>
</tr>
<tr>
<td>Figure 5.1</td>
<td>User Input Window</td>
<td>13</td>
</tr>
<tr>
<td>Figure 5.2</td>
<td>Code to Create User Input Window</td>
<td>13</td>
</tr>
<tr>
<td>Figure 5.3</td>
<td>Method to Construct JCheckbox</td>
<td>14</td>
</tr>
<tr>
<td>Figure 5.4</td>
<td>Condition to Check the User Input</td>
<td>14</td>
</tr>
<tr>
<td>Figure 5.5</td>
<td>Catch the Error If the Condition True</td>
<td>15</td>
</tr>
<tr>
<td>Figure 5.6</td>
<td>Condition If the Button was Pressed</td>
<td>15</td>
</tr>
<tr>
<td>Figure 5.7</td>
<td>Checkbox Selected and Added the Random Data</td>
<td>16</td>
</tr>
<tr>
<td>Figure 5.8</td>
<td>Checkbox not Selected and Added Less Data</td>
<td>16</td>
</tr>
<tr>
<td>Figure 5.9</td>
<td>Checkbox not Selected and Added More Data</td>
<td>16</td>
</tr>
<tr>
<td>Figure 5.10</td>
<td>Class to Construct Node</td>
<td>17</td>
</tr>
<tr>
<td>Figure 5.11</td>
<td>Method for Inserting Data to the Node</td>
<td>17</td>
</tr>
<tr>
<td>Figure 5.12</td>
<td>Insert Data to the Node Leaf</td>
<td>18</td>
</tr>
<tr>
<td>Figure 5.13</td>
<td>Find Node Leaf and Insert the Data</td>
<td>18</td>
</tr>
<tr>
<td>Figure 5.14</td>
<td>Split Full Node, Promote Median to New Node</td>
<td>19</td>
</tr>
<tr>
<td>Figure 5.15</td>
<td>Passing the Data that Greater Than Median</td>
<td>20</td>
</tr>
</tbody>
</table>
Figure 5.16 : Set the New Node as Next Child ................. 20
Figure 5.17 : User Input, if Not A Number .................... 21
Figure 5.18 : User Input, if Number more than 2 Digits .......... 21
Figure 5.19 : User Input, True .................................. 21
Figure 5.20 : Fill Data in the Node Root ......................... 22
Figure 5.21 : Splitting the Full Node Success .................... 22
Figure 5.22 : Node Child Full ...................................... 22
Figure 5.23 : Splitting the Node Child Success .................... 23
Figure 5.24 : All Child Nodes Full ................................. 23
Figure 5.25 : Node Root Full, Program Done ...................... 24
Table of Table

Table 3.1 : Project Management ........................................... 8