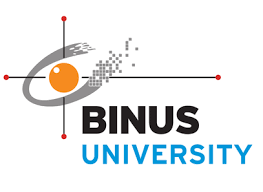
DATA INVESTOR

LAPORAN PROYEK AKHIR

MATA KULIAH COMP6362 – DATA STRUCTURES

KELAS BD20



Oleh:

2301938440 - ATHALLAH DADY BRAMASTA PUTRA

2301913690 - WILLIAM DEOVALDO

2301878170 - YOGA PRASETYA

Semester Genap (2019/2020)

MALANG

LEMBAR PERSETUJUAN PROYEK AKHIR

MATA KULIAH COMP6362 – DATA STRUCTURES

KELAS BD20

Semester Genap (2019/2020)

Laporan akhir proyek ini adalah benar karya kami :

ATHALLAH DADY BRAMASTA PUTRA WILLIAM DEOVALDO YOGA PRASETYA

2301938440 2301913690 2301878170

Malang, …… - ……………. ……..

**Hanugra Aulia Sidharta**

**D5850**

LEMBAR PENILAIAN

**PROGRAM DATA INVESTOR**

MATA KULIAH COMP6362 – DATA STRUCTURES

KELAS BD20

Semester Genap, 2019/2020

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DAFTAR MAHASISWA** | **NILAI** | | | | **BOBOT** | | | | **KREDIT** | | | | **TOTAL KREDIT** |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | 4 | 1 | 2 | 3 | 4 |
| 2301938440 - ATHALLAH DADY BRAMASTA PUTRA |  |  |  |  | 20% | 20% | 24% | 36% |  |  |  |  |  |
| 2301913690 - WILLIAM DEOVALDO |  |  |  |  | 20% | 20% | 24% | 36% |  |  |  |  |  |
| 2301878170 - YOGA PRASETYA |  |  |  |  | 20% | 20% | 24% | 36% |  |  |  |  |  |
| **TOTAL** | | | | | | | | | | | | |  |

**KETERANGAN :**

* **Skala Penilaian : 0 sd 100**
* **Komponen**

**Malang, …… - ……………. ……..**

**(Hanugra Aulia Sidharta)**

**< D5850>**

|  |  |  |
| --- | --- | --- |
| 1 | : | Laporan |
| 2 | : | Produk |
| 3 | : | Pengetahuan |
| 4 | : | Solusi |

**DAFTAR ISI**

HALAMAN SAMPUL 1

LEMBAR PERSETUJUAN 2

LEMBAR PENILAIAN 3

DAFTAR ISI 4

BAB 1 5

BAB 2 6

BAB 3 7

BAB 4 8

DAFTAR PUSTAKA 16

**BAB 1**

**LATAR BELAKANG**

Tim kami berencana untuk membuat aplikasi yang bisa memberikan informasi tentang keadaan saham suatu perusahaan dan data tentang para pemegang saham ,jumlah pemegang saham dan daftar para pemegang saham menggunakann Binary Search Tree.

Alasan memilih topik dan kecocokan antara kasus yang dipilih dan metode sturktur data. Karena simple dan mudah untuk dikerjarakan dan mudah untuk digunakan juga, kecocokan antara struktur data dan kasus yang dipilih juga berguna karena berhubungan satu sama lain.

Program yang dibuat memberikan pilihan untuk memasukkan data pemegang saham baru dan daftar serta jumlah investor yang ada di perusahaan yang keseluruhannya menampilkan data pemegang saham perusahaan, dengan memasukkan kelengapan data, seperti nama, dan jumlah yag diinvestasikan.

**BAB 2**

**Tinjauan Pustaka**

Data adalah salah satu hal penting yang tidak dapat ditinggalkan dalam pemakaian computer. Data dapat diperoleh dari berbagai sumber seperti pengukuran laboratorium, hasil survei, hasil angket dan lain sebagainya. Data sederhana dapat kita himpun ke dalam suatu struktur data yang memuat informasi tentang hubungan antara data yang terdapat di dalamnya.

Dalam istilah ilmu komputer, struktur data adalah cara penyimpanan, pengorganisasian, dan pengaturan data di dalam media penyimpanan komputer sehingga data tersebut dapat digunakan secara efisien. Dalam teknik pemrograman, struktur data berarti tata letak data yang berisi kolom-kolom data,baik itu kolom yang tampak oleh pengguna ataupun kolom yang hanya digunakan untuk keperluan pemrograman yang tidak tampak oleh pengguna.

Pada garis besarnya, data dapat dikategorikan menjadi beberapa jenis, yaitu :

* Tipe data sederhana tunggal (Integer, Real, Boolean, Karakter)
* Tipe data sederhana majemuk (String)
* Struktur data sederhana (Array, Record)
* Struktur data majemuk (Stack, Queue, Binary tree, Binary search tree, Graph)

Pemakaian struktur data yang tepat di dalam proses pemrograman akan menghasilkan algoritma yang lebih jelas dan tepat, sehingga menjadikan program secara keseluruhan lebih efisien dan sederhana.

Tree adalah salah satu bentuk struktur data yang menggambarkan hubungan hierarki antar elemen-elemennya. Sebuah node dalam tree biasanya bisa memiliki beberapa node lagi sebagai percabangan atas dirinya.

Binary tree adalah tree yang hanya dapat mempunyai maksimal 2 percabangan saja.

Binary Search Tree adalah sebuah konsep penyimpanan data, dimana data disimpan dalam bentuk tree yang setiap node dapat memiliki anak maksimal 2 node. Selain itu, terdapat juga aturan dimana anak kiri dari parent selalu memiliki nilai lebih kecil dari nilai parent dan anak kanan selalu memiliki nilai lebih besar dari parent.

Ada 3 jenis cara untuk melakukan penelusuran data (traversal) pada BST :

* PreOrder : Print data, telusur ke kiri, telusur ke kanan
* InOrder : Telusur ke kiri, print data, telusur ke kanan
* Post Order : Telusur ke kiri, telusur ke kanan, print data

**BAB 3**

**GAMBARAN UMUM PROGRAM**

Pada kesempatan ini kami mengambil masalah mengenai program penyimpanan data investor menggunakan bahasa pemrograman C. Program ini memudahkan admin untuk mengelola data para investor.

Aplikasi ini, memiliki isi dimana kita dapat memilih untuk menambahkan data investor, mencari data investor, melihat data investor, dan mengecek jumlah investor.

1. Buat data investor baru

Pada menu ini, petugas dapat menambahkan data investor baru yaitu, nama investor, nama perusahaan, dan jumlah investasinya.

2. Cari data

Pada menu ini, petugas dapat memastikan bahwa data investor yang telah dimasukkan sudah terdata.

3. List

Pada menu ini, petugas dapat melihat semua data investasi yang telah diinput berdasarkan nilai investasi paling tinggi atau paling rendah.

4. Cek jumlah investor

Pada menu ini, petugas dapat mengecek berapa banyak orang yang telah berinvestasi secara kkeseluruhan.

Berikut contoh data investor yang telah dimasukan:

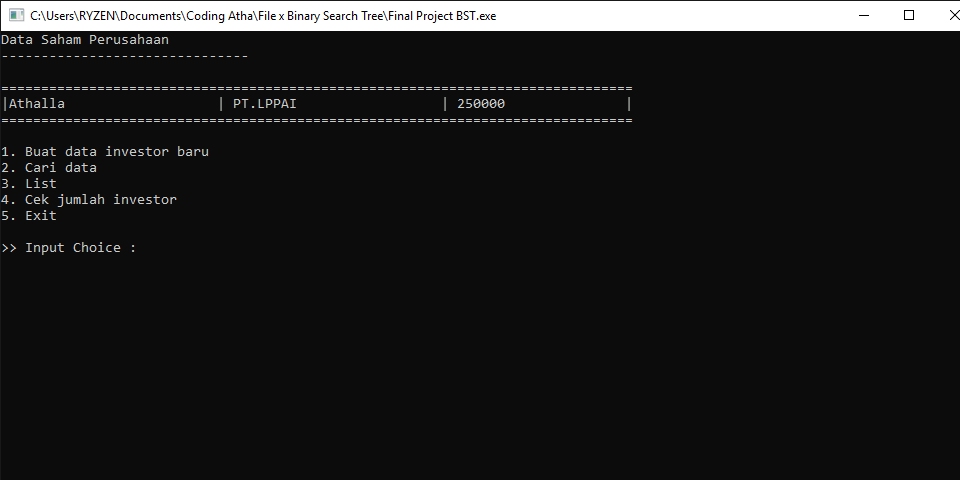
|  |  |  |
| --- | --- | --- |
| Nama Investor | Nama Perusahaan | Jumlah Investasi |
| Athalla | PT.LPPAI | 250000 |

**BAB 4**

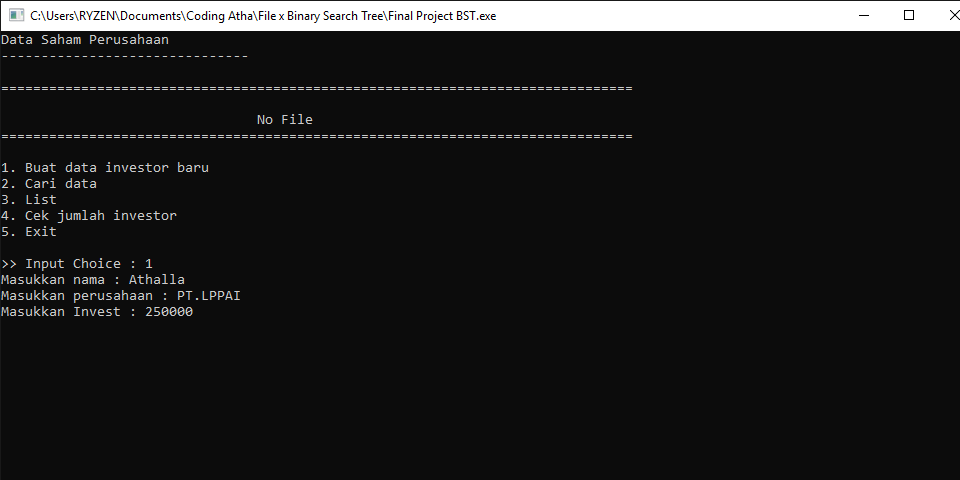
**HASIL**

**Screenshot program**

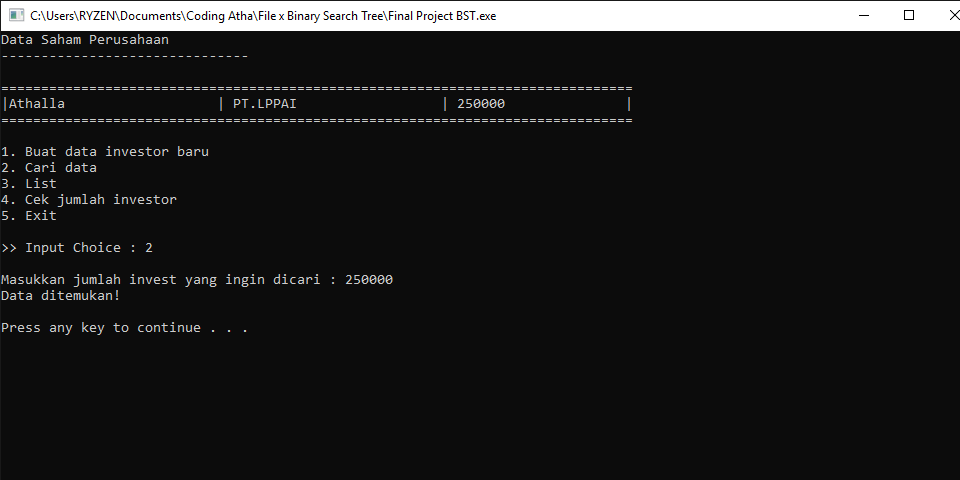
* Halaman menu utama

****

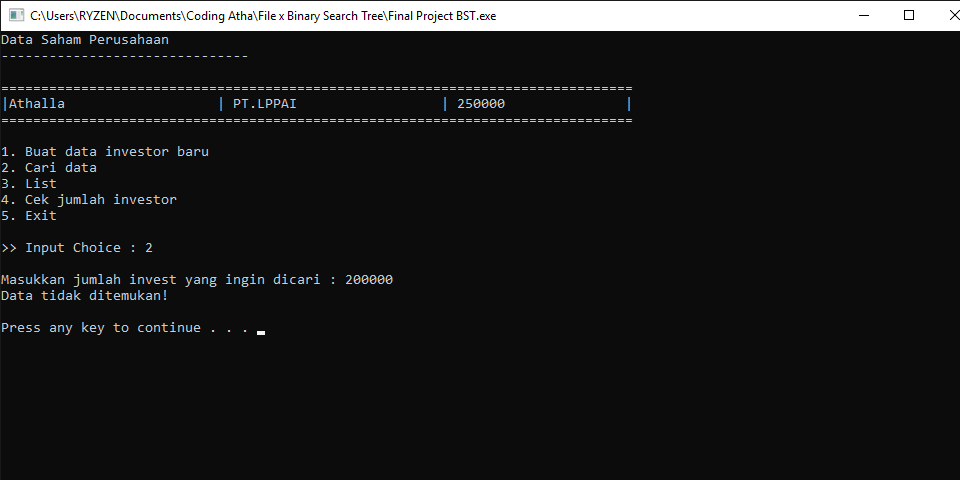
* Menu buat data investor baru

****

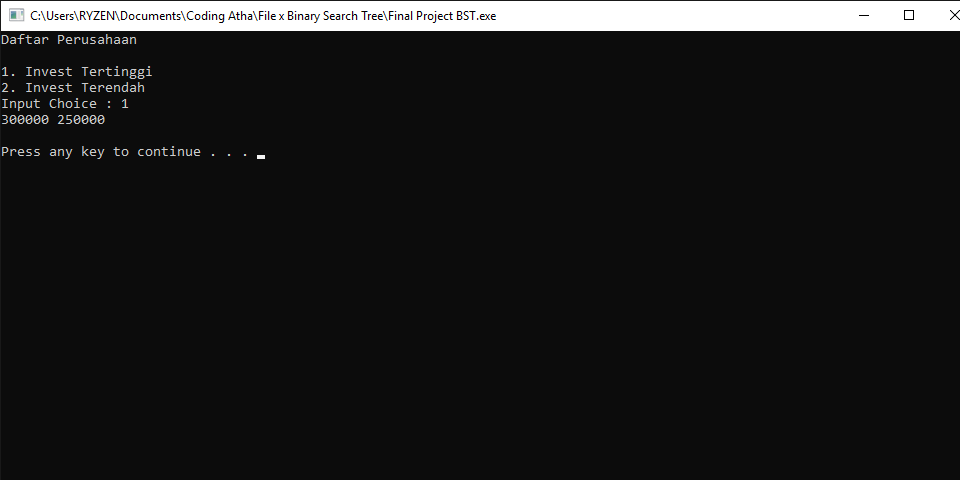
* Menu cari data (ditemukan)

****

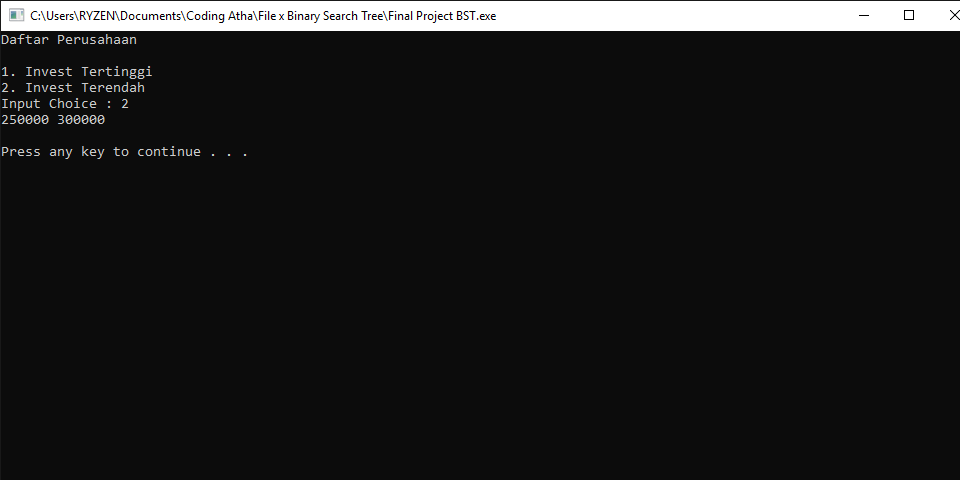
* Menu cari data (tidak ditemukan)



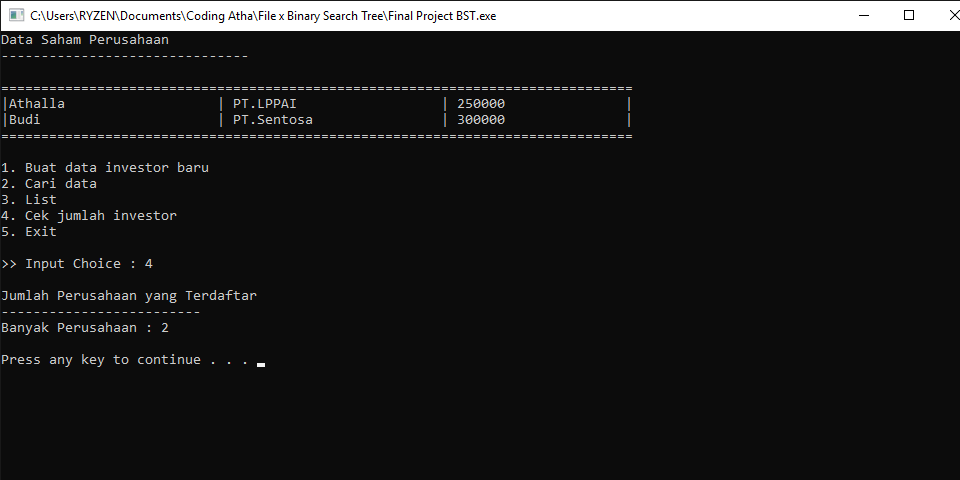
* Menu List (tertinggi)



* Menu List (terendah)



* Menu Cek jumlah investor



**SOURCE CODE**

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <string.h>
4. struct node{
5. char nama[30];
6. char perusahaan[30];
7. int invest;
8. node\*kiri;
9. node\*kanan;
10. };
11. node \*pohon = NULL;
12. clr(){
13. system("cls");
14. }
15. void addNew(node \*\*root, int databaru){
17. if ((\*root)==NULL){
18. node \*baru;
20. baru = new node;
22. baru->invest=databaru;
23. baru->kiri=NULL;
24. baru->kanan=NULL;
25. (\*root) = baru;
26. (\*root) -> kiri = NULL;
27. (\*root) -> kanan = NULL;
28. }
30. else if (databaru<(\*root)->invest)
31. addNew(&(\*root)->kiri,databaru);
33. else if (databaru>(\*root)->invest)
34. addNew(&(\*root)->kanan,databaru);
35. else if (databaru==(\*root)->invest);
36. }
37. void fileread(){
38. char nama[30];
39. char perusahaan[30];
40. int invest;
41. FILE \*in=fopen("test.txt","r");
42. if(!in){
43. printf("\n\t\t\t\tNo File\n");
44. }else{
45. while(!feof(in)){
46. fscanf(in,"%[^#]#%[^#]#%d\n", &nama, &perusahaan, &invest);fflush(stdin);
47. printf("|%-25s | %-25s | %-20d |\n", nama, perusahaan, invest);
48. addNew(&pohon, invest);
49. }
50. fclose(in);
51. }
52. }
53. void fileadd(){
55. struct node \*temp = (struct node\*)malloc(sizeof(struct node));
57. char nama[30];
58. char perusahaan[30];
59. int invest;
60. printf("Masukkan nama : "); gets(nama); fflush(stdin);
61. printf("Masukkan perusahaan : "); gets(perusahaan); fflush(stdin);
62. printf("Masukkan Invest : "); scanf("%d", &invest); fflush(stdin);
63. addNew(&pohon, invest);
64. strcpy(temp->nama, nama);
65. strcpy(temp->perusahaan, perusahaan);
66. FILE \*out=fopen("test.txt","a+");
67. fprintf(out,"%s#%s#%d\n",nama, perusahaan, invest);
68. fclose(out);
69. printf("\n\nSukses menambah data.");
70. getchar();
71. }
72. void filedelete(){
73. char nama[30];
74. char perusahaan[30];
75. int invest;
76. FILE \*in=fopen("test.txt","r");
77. if(!in){
78. printf("\t\t\t\tNo File\n\n");
79. }else{
80. while(!feof(in)){
81. fscanf(in,"%[^#]#%[^#]#%d\n", &nama, &perusahaan, &invest);fflush(stdin);
82. printf("%-25s | %-25s | %-20d |\n", nama, perusahaan, invest);
83. }
84. fclose(in);
85. }
86. }
87. int counter(node \*root){
88. if(root==NULL)
89. return 0;
90. else
91. return counter(root->kiri)+ counter(root->kanan)+1;
92. }
93. void hitung(){
94. printf("\nJumlah Perusahaan yang Terdaftar");
95. printf("\n-------------------------");
96. printf("\nBanyak Perusahaan : %d\n\n", counter(pohon));
97. system("pause");
98. }
99. void menu(){
100. printf("Data Saham Perusahaan\n");
101. printf("-------------------------------\n\n");
102. printf("===============================================================================\n");
103. fileread();
104. printf("===============================================================================\n\n");
105. printf("1. Buat data investor baru\n");
106. printf("2. Cari data\n");
107. printf("3. List\n");
108. printf("4. Cek jumlah investor\n");
109. printf("5. Exit\n");
110. }
111. int inOrder(node \*root){
112. char nama[30];
113. char perusahaan[30];
114. if(root!=NULL)
115. {
116. inOrder(root->kiri);
117. if(root->invest!=NULL)
118. {
119. printf("%d ",root->invest);
120. }
121. inOrder(root->kanan);
122. }
123. return 0;
124. }
125. void postOrder(node \*root){
126. char nama[30];
127. char perusahaan[30];
128. if(root!=NULL){
129. postOrder(root->kiri);
130. postOrder(root->kanan);
131. if(root->invest!=NULL)
132. {
133. printf("%d ",root->invest);
134. }
135. }
136. }
137. void search(node \*\*root, int cari){
138. if((\*root) == NULL){
139. printf("Data tidak ditemukan!");
140. }
141. else if(cari < (\*root)->invest)
142. search(&(\*root)->kiri,cari);
143. else if(cari > (\*root)->invest)
144. search(&(\*root)->kanan,cari);
145. else if(cari == (\*root)->invest)
146. printf("Data ditemukan!");
148. }
149. void searchmenu(){
150. int cari;
151. printf("\nMasukkan jumlah invest yang ingin dicari : ");
152. scanf("%d", &cari);
153. if(pohon!=NULL){
154. search(&pohon, cari);
155. }else{
156. printf("\nMasih kosong!");
157. }
158. printf("\n\n");
159. system("pause");
160. }
161. void listmenu(){
162. int list\_menu;
163. clr();
164. printf("Daftar Perusahaan\n\n");
165. printf("1. Invest Tertinggi\n");
166. printf("2. Invest Terendah\n");
167. printf("Input Choice : ");
168. scanf("%d", &list\_menu);
169. switch(list\_menu){
170. case 1 :
171. postOrder(pohon);
172. break;
173. case 2 :
174. inOrder(pohon);
175. break;
176. }
177. printf("\n\n");
178. system("pause");
179. }
180. main(){
181. int input;
182. do{
183. do{
184. clr();
185. menu();
186. printf("\n>> Input Choice : ");
187. scanf("%d", &input);
188. }while(input > 5|| input < 1);
190. switch(input){
191. case 1 :
192. fileadd();
193. break;
194. case 2 :
195. searchmenu();
196. break;
197. case 3 :
198. listmenu();
199. break;
200. case 4:
201. hitung();
202. break;
203. case 5:
204. break;
205. }
206. }while(input<6);
207. }

**DAFTAR PUSTAKA**

* <https://www.it-jurnal.com/pengertian-struktur-data/>
* <https://www.mahirkoding.com/struktur-data-binary-search-tree-bst/>
* <https://socs.binus.ac.id/2017/05/10/implementasi-insert-pada-binary-search-tree-dengan-single-dan-double-pointer/>