**Report of Programming Practicum VIII**

Nurrizky Imani

18/429294/PA/18685

**Exercise 2 : Binary Search**

#include <iostream>

#include <string>

using namespace std;

struct Student {

string name;

int NISN;

int value;

};

Student data[100];

int main () {

data[0].name = "Handi Ramadhan" ;

data[0].NISN = 699 ;

data[0].value = 90 ;

data[1].name = "Rio Alfandra" ;

data[1].NISN = 682 ;

data[1].value = 55 ;

data[2].name = "Ronaldo Valentino Uneputty" ;

data[2].NISN = 962 ;

data[2].value = 80 ;

data[3].name = "Achmad Yaumil Fadjri R." ;

data[3].NISN = 750 ;

data[3].value = 60 ;

data[4].name = "Alivia Rahma Pramesti" ;

data[4].NISN = 945 ;

data[4].value = 70 ;

data[5].name = "Ari Lutfianto" ;

data[5].NISN = 180 ;

data[5].value = 65 ;

data[6].name = "Arief Budiman" ;

data[6].NISN = 989 ;

data[6].value = 60 ;

int temp, temp3, min; string temp2;

for (int i=0; i<7-1; i++){

min=i;

for (int j=i+1; j<7; j++){

if (data[j].NISN < data[min].NISN)

{

min=j;

}

}

temp=data[i].NISN;

temp2=data[i].name;

temp3=data[i].value;

data[i].NISN=data[min].NISN;

data[i].name=data[min].name;

data[i].value=data[min].value;

data[min].NISN=temp;

data[min].name=temp2;

data[min].value=temp3 ;

}

for (int k=0; k<7; k++){

cout << data[k].NISN << " " << data[k].name << " " << data[k].value << endl;

}

// binary search

int i = 0, j = 6, k , query = 962;

bool found = false;

//searchalgos

cout<< endl ;

while(!found&&i<=j){

k=(i+j)/2;

if(data[k].NISN<query){

i=k+1;

}else if(data[k].NISN == query){

found=true;

}else {

j=k-1;

} }

//print

if(!found){

cout<<"Tidak ditemukan.";

}else{

cout<<"Ditemukan.";

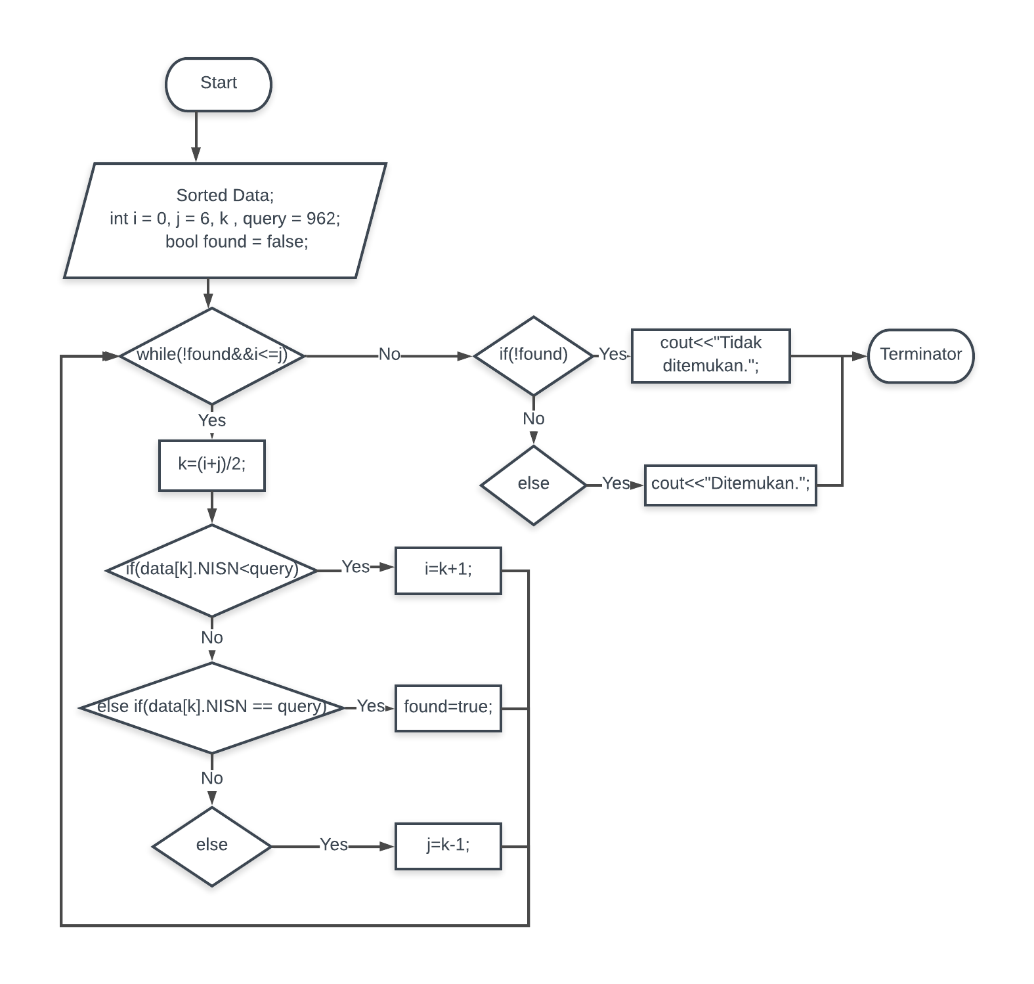
}

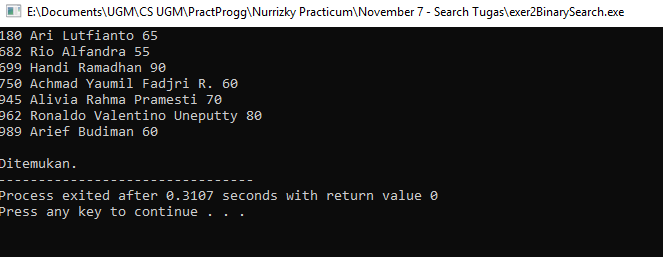
}

**Description:**

1. We start by sorting the data first according to the nisn. Without sorting, the data cannot be process by the search algorithm. Any sorting algorithm is allowed as long as it is sort the data from the minimal to the maximal.
2. Import the main library
3. Initialize the struct of data in the name of Student with the data inside : name as string, nisn as integer and value as integer. We initialize the student under the name data[100].
4. Inside the main function, we declare all the data the user have with struct syntax. Every data are ordered with index inside the data[i].
5. Then we initialize several integer : temp, temp3, min and string temp2. We declare it for storing temporary data for swaping number in sorting.
6. The sorting algorithm started. First, we would like to declare for loop with variable i = 0 and constraint of 7, increment i. inside it, we creat local variable min = i. Then we create another loop, nested loop, with variable j is i+1, constraint 7 and increment j which execute task to find min or max value and store it in min. Notice we store the index of j in min.
7. The next algorithm is sorting algorithm which swaping all different data name, nisn, value, to temporary variable. And fill those index with the min index variable. And store the temp data to the min index. Next algorithm is print the algorithm.
8. The program start the search algorithm. Starting with initialize variable such as I = 0, j= 6, k and query which is the number we search and found assign as Boolean.
9. Using while with requirement condition of Not found and i must less than j. The reccurent execute algorithm. Notice that we use control statement inside the whileloop to manipulate variable i or j. The loop will always execute until the condition is False. In which it bring to the next control statement which will print “Tidak Ditemukan” or “Ditemukan”.

**Flowchart:**



**Screenshot:**

**Exercise 3 Sequential Search**

**Code:**

#include <iostream>

#include <string>

using namespace std;

struct Student {

string name;

int NISN;

int value;

};

Student data[100];

int main () {

data[0].name = "Handi Ramadhan" ;

data[0].NISN = 699 ;

data[0].value = 90 ;

data[1].name = "Rio Alfandra" ;

data[1].NISN = 682 ;

data[1].value = 55 ;

data[2].name = "Ronaldo Valentino Uneputty" ;

data[2].NISN = 962 ;

data[2].value = 80 ;

data[3].name = "Achmad Yaumil Fadjri R." ;

data[3].NISN = 750 ;

data[3].value = 60 ;

data[4].name = "Alivia Rahma Pramesti" ;

data[4].NISN = 945 ;

data[4].value = 70 ;

data[5].name = "Ari Lutfianto" ;

data[5].NISN = 180 ;

data[5].value = 65 ;

data[6].name = "Arief Budiman" ;

data[6].NISN = 989 ;

data[6].value = 60 ;

int temp, temp3, min; string temp2;

for (int i=0; i<7-1; i++){

min=i;

for (int j=i+1; j<7; j++){

if (data[j].NISN < data[min].NISN)

{

min=j;

}

}

temp=data[i].NISN;

temp2=data[i].name;

temp3=data[i].value;

data[i].NISN=data[min].NISN;

data[i].name=data[min].name;

data[i].value=data[min].value;

data[min].NISN=temp;

data[min].name=temp2;

data[min].value=temp3 ;

}

for (int k=0; k<7; k++){

cout << data[k].NISN << " " << data[k].name << " " << data[k].value << endl;

}

//search

int query = 60;

bool found = false;

string temp4 = "Joko";

//searchalgos

cout<< endl <<"Change All name's with value of 60 into : " << endl << endl;

for( int i = 0; i < 7; i++ ){

if (data[i].value == query){

data[i].name = temp4;

cout << data[i].NISN << " " << data[i].name << " " <<data[i].value << endl;

found = true;

}

}

if (!found){

cout<< "Tidak Ditemukan.";

}

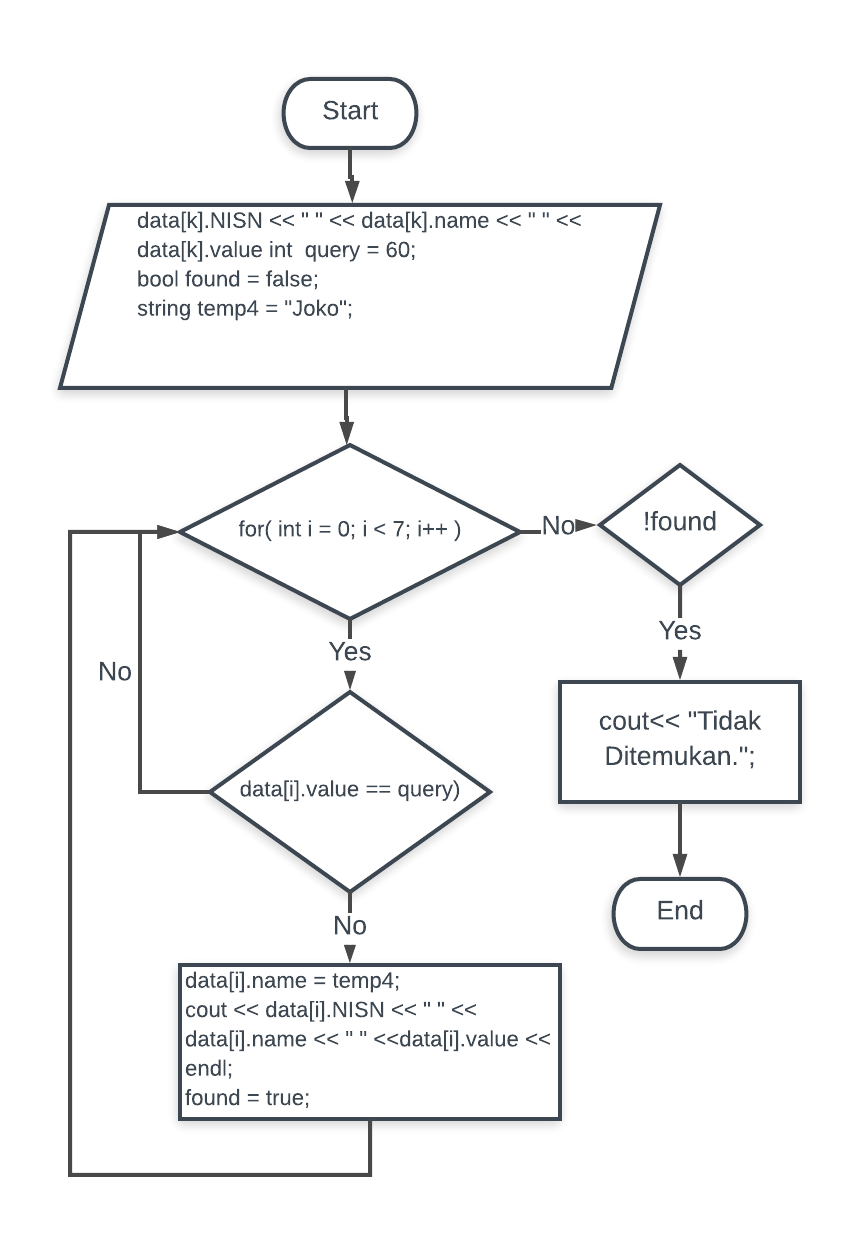
return 0;

}

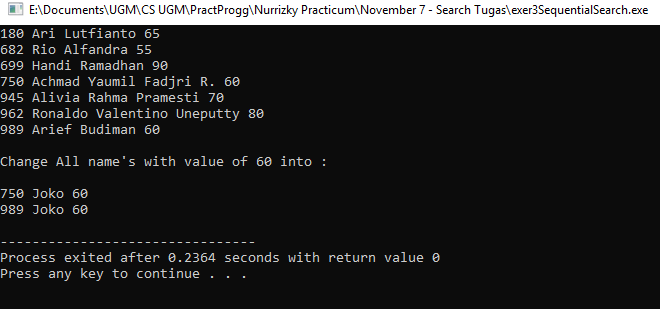
**Description:**

1. Import the main library
2. Initialize the struct of data in the name of Student with the data inside : name as string, nisn as integer and value as integer. We initialize the student under the name data[100].
3. The program depict sorting algorithm in which it is optional to include. Here, to include in case of fulfilling more detail information.
4. The search algorithm start by initialize several variable: query as variable in which we look upon into. Found assign as Boolean and temp4 as string for manipulate the name in which have 60 as the value.
5. Using forloop with constraint of 7, conditional statement is set in place inside it with requerment if the value in data for each index is equal to query. If it is true, the program will change the name into “Joko”. Then, it will print the nisn, name, and value of the data. Also, we add another if for catch another condition which is negation found which will print “Tidak Ditemukan”.

**Flowchart:**



**Screenshot:**

****