Nama : Andri Firman Saputra

NIM : 201011402125

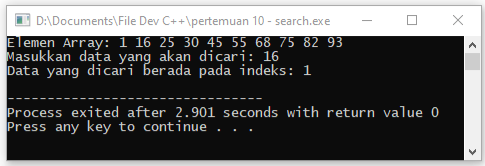
Kelas : 02TPLP023

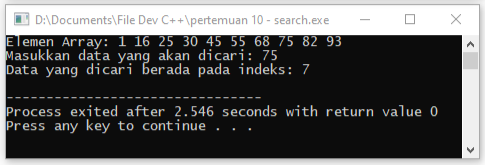
Tugas : Algoritma II – Pertemuan 10

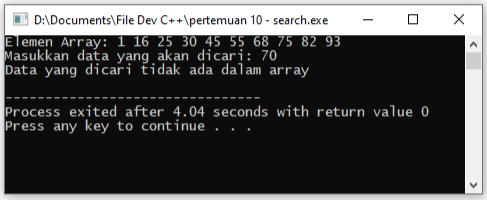
1.

pertemuan 10 - search.cpp

```cpp
#include<iostream>
using namespace std;

void BinSearch(int data[], int n, int x, int *idx);

int main()
{
    int data[] = {1, 16, 25, 30, 45, 55, 68, 75, 82, 93};
    int idx, x, i, jmlData = 10;

    cout<<"Elemen Array: ";
    for(i = 0; i < jmlData; i++)
    {
        cout<<data[i]<<" ";
    }
    cout<<endl;

    cout<<"Masukkan data yang akan dicari: ";
    cin>>x;

    BinSearch(data, jmlData, x, &idx);
    if(idx != -1)
    {
        cout<<"Data yang dicari berada pada indeks: "<<idx<<endl;
    }
    else
    {
        cout<<"Data yang dicari tidak ada dalam array"<<endl;
    }

}

void BinSearch(int data[], int n, int x, int *idx)
{
    bool ketemu = false;
    int top = n-1, bottom = 0, mid;
    int i = 0;
    while(bottom <= top && !ketemu)
    {
        mid = (top + bottom) / 2;

        if(data[mid] == x)
        {
            ketemu = true;
        }
        else
        {
            if(data[mid] > x)
            {
                top = mid - 1;
            }
            else
            {
                bottom = mid + 1;
            }
        }
    }

    if(ketemu)
    {
        *idx = mid;
    }
    else
    {
        *idx = -1;
    }
}
```







Source code:

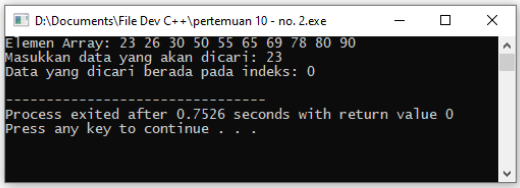
```
#include<iostream>
using namespace std;

void BinSearch(int data[], int n, int x, int *idx);

int main()
{
   int data[] = {1, 16, 25, 30, 45, 55, 68, 75, 82, 93};
   int idx, x, i, jmlData = 10;

   cout<<"Elemen Array: ";
   for(i = 0; i < jmlData; i++)
   {
          cout<<data[i]<<" ";
   }
   cout<<endl;

   cout<<"Masukkan data yang akan dicari: ";
   cin>>x;

   BinSearch(data, jmlData, x, &idx);
   if(idx != -1)
   {
          cout<<"Data yang dicari berada pada indeks: "<<idx<<endl;
   }
   else
   {
          cout<<"Data yang dicari tidak ada dalam array"<<endl;
   }

}

void BinSearch(int data[], int n, int x, int *idx)
{
   bool ketemu = false;
   int top = n-1, bottom = 0, mid;
   int i = 0;
   while(bottom <= top && !ketemu)
   {
          mid = (top + bottom) / 2;

          if(data[mid] == x)
          {
                 ketemu = true;
          }
          else
          {
                 if(data[mid] > x)
                 {
                        top = mid - 1;
                 }
                 else
                 {
                        bottom = mid + 1;
                 }
          }
   }
```

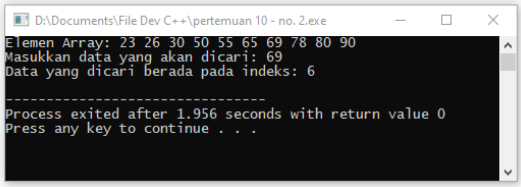
```
    if(ketemu)
    {
            *idx = mid;
    }
    else
    {
            *idx = -1;
    }
}
```

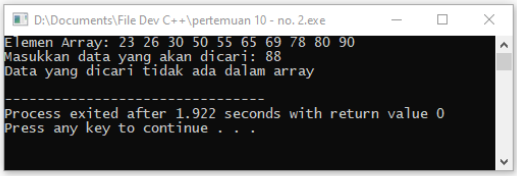
2.

pertemuan 10 - no. 2.cpp

```
#include<iostream>
using namespace std;

void BinSearch(int data[], int n, int x, int *idx);

int main()
{
    int data[] = {23, 26, 30, 50, 55, 65, 69, 78, 80, 90};
    int idx, x, i, jmlData = 10;

    cout<<"Elemen Array: ";
    for(i = 0; i < jmlData; i++)
    {
        cout<<data[i]<<" ";
    }
    cout<<endl;

    cout<<"Masukkan data yang akan dicari: ";
    cin>>x;

    BinSearch(data, jmlData, x, &idx);
    if(idx != -1)
    {
        cout<<"Data yang dicari berada pada indeks: "<<idx<<endl;
    }
    else
    {
        cout<<"Data yang dicari tidak ada dalam array"<<endl;
    }

}

void BinSearch(int data[], int n, int x, int *idx)
{
    bool ketemu = false;
    int top = n-1, bottom = 0, mid;
    int i = 0;
    while(bottom <= top && !ketemu)
    {
        mid = (top + bottom) / 2;

        if(data[mid] == x)
        {
            ketemu = true;
        }
        else
        {
            if(data[mid] > x)
            {
                top = mid - 1;
            }
            else
            {
                bottom = mid + 1;
            }
        }
    }

    if(ketemu)
    {
        *idx = mid;
    }
    else
    {
        *idx = -1;
    }
}
```







Source Code:

```
#include<iostream>
using namespace std;

void BinSearch(int data[], int n, int x, int *idx);

int main()
{
   int data[] = {23, 26, 30, 50, 55, 65, 69, 78, 80, 90};
   int idx, x, i, jmlData = 10;

   cout<<"Elemen Array: ";
   for(i = 0; i < jmlData; i++)
   {
          cout<<data[i]<<" ";
   }
   cout<<endl;

   cout<<"Masukkan data yang akan dicari: ";
   cin>>x;

   BinSearch(data, jmlData, x, &idx);
   if(idx != -1)
   {
          cout<<"Data yang dicari berada pada indeks: "<<idx<<endl;
   }
   else
   {
          cout<<"Data yang dicari tidak ada dalam array"<<endl;
   }

}

void BinSearch(int data[], int n, int x, int *idx)
{
   bool ketemu = false;
   int top = n-1, bottom = 0, mid;
   int i = 0;
   while(bottom <= top && !ketemu)
   {
```

```
            mid = (top + bottom) / 2;

            if(data[mid] == x)
            {
                  ketemu = true;
            }
            else
            {
                  if(data[mid] > x)
                  {
                        top = mid - 1;
                  }
                  else
                  {
                        bottom = mid + 1;
                  }
            }
      }

      if(ketemu)
      {
            *idx = mid;
      }
      else
      {
            *idx = -1;
      }
}
```

3.

pertemuan 10 - no. 3.cpp

```cpp
#include<iostream>
using namespace std;

void BinSearch(int data[], int n, int x, int *idx);

int main()
{
    int data[] = {90, 80, 78, 65, 54, 45, 32, 23, 19, 4};
    int idx, x, i, jmlData = 10;

    cout<<"Elemen Array: ";
    for(i = 0; i < jmlData; i++)
    {
        cout<<data[i]<<" ";
    }
    cout<<endl;

    cout<<"Masukkan data yang akan dicari: ";
    cin>>x;

    BinSearch(data, jmlData, x, &idx);
    if(idx != -1)
    {
        cout<<"Data yang dicari berada pada indeks: "<<idx<<endl;
    }
    else
    {
        cout<<"Data yang dicari tidak ada dalam array"<<endl;
    }

}

void BinSearch(int data[], int n, int x, int *idx)
{
    bool ketemu = false;
    int top = n-1, bottom = 0, mid;
    int i = 0;
    while(bottom <= top && !ketemu)
    {
        mid = (top + bottom) / 2;

        if(data[mid] == x)
        {
            ketemu = true;
        }
        else
        {
            if(data[mid] < x)
            {
                top = mid - 1;
            }
            else
            {
                bottom = mid + 1;
            }
        }
    }

    if(ketemu)
    {
        *idx = mid;
    }
    else
    {
        *idx = -1;
    }
}
```

```
D:\Documents\File Dev C++\pertemuan 10 - no. 3.exe
Elemen Array: 90 80 78 65 54 45 32 23 19 4
Masukkan data yang akan dicari: 80
Data yang dicari berada pada indeks: 1

--------------------------------
Process exited after 2.67 seconds with return value 0
Press any key to continue . . .
```

```
D:\Documents\File Dev C++\pertemuan 10 - no. 3.exe
Elemen Array: 90 80 78 65 54 45 32 23 19 4
Masukkan data yang akan dicari: 45
Data yang dicari berada pada indeks: 5

--------------------------------
Process exited after 7.83 seconds with return value 0
Press any key to continue . . .
```

```
D:\Documents\File Dev C++\pertemuan 10 - no. 3.exe
Elemen Array: 90 80 78 65 54 45 32 23 19 4
Masukkan data yang akan dicari: 69
Data yang dicari tidak ada dalam array

--------------------------------
Process exited after 4.486 seconds with return value 0
Press any key to continue . . .
```

Source Code:

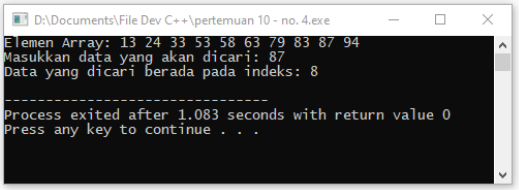
```
#include<iostream>
using namespace std;

void BinSearch(int data[], int n, int x, int *idx);

int main()
{
	int data[] = {90, 80, 78, 65, 54, 45, 32, 23, 19, 4};
	int idx, x, i, jmlData = 10;

	cout<<"Elemen Array: ";
	for(i = 0; i < jmlData; i++)
	{
		cout<<data[i]<<" ";
	}
	cout<<endl;

	cout<<"Masukkan data yang akan dicari: ";
	cin>>x;

	BinSearch(data, jmlData, x, &idx);
	if(idx != -1)
	{
		cout<<"Data yang dicari berada pada indeks: "<<idx<<endl;
	}
	else
	{
		cout<<"Data yang dicari tidak ada dalam array"<<endl;
	}

}

void BinSearch(int data[], int n, int x, int *idx)
{
```

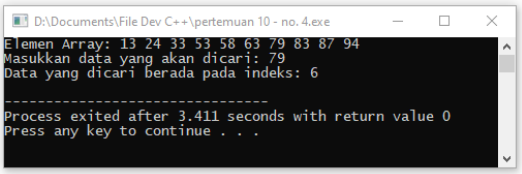
```
	bool ketemu = false;
	int top = n-1, bottom = 0, mid;
	int i = 0;
	while(bottom <= top && !ketemu)
	{
		mid = (top + bottom) / 2;

		if(data[mid] == x)
		{
			ketemu = true;
		}
		else
		{
			if(data[mid] < x)
			{
				top = mid - 1;
			}
			else
			{
				bottom = mid + 1;
			}
		}
	}

	if(ketemu)
	{
		*idx = mid;
	}
	else
	{
		*idx = -1;
	}
}
```

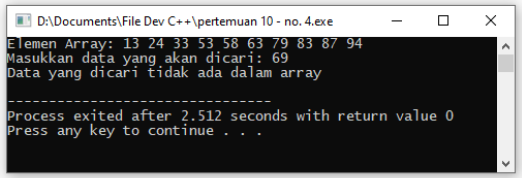
4.

pertemuan 10 - no. 4.cpp

```cpp
#include<iostream>
using namespace std;

void BinSearch(int data[], int n, int x, int *idx);

int main()
{
    int data[] = {13, 24, 33, 53, 58, 63, 79, 83, 87, 94};
    int idx, x, i, jmlData = 10;

    cout<<"Elemen Array: ";
    for(i = 0; i < jmlData; i++)
    {
        cout<<data[i]<<" ";
    }
    cout<<endl;

    cout<<"Masukkan data yang akan dicari: ";
    cin>>x;

    BinSearch(data, jmlData, x, &idx);
    if(idx != -1)
    {
        cout<<"Data yang dicari berada pada indeks: "<<idx<<endl;
    }
    else
    {
        cout<<"Data yang dicari tidak ada dalam array"<<endl;
    }

}

void BinSearch(int data[], int n, int x, int *idx)
{
    bool ketemu = false;
    int top = n-1, bottom = 0, mid;
    int i = 0;
    while(bottom <= top && !ketemu)
    {
        mid = (top + bottom) / 2;

        if(data[mid] == x)
        {
            ketemu = true;
        }
        else
        {
            if(data[mid] > x)
            {
                top = mid - 1;
            }
            else
            {
                bottom = mid + 1;
            }
        }
    }

    if(ketemu)
    {
        *idx = mid;
    }
    else
    {
        *idx = -1;
    }
}
```







Source Code:

```
#include<iostream>
using namespace std;

void BinSearch(int data[], int n, int x, int *idx);

int main()
{
	int data[] = {13, 24, 33, 53, 58, 63, 79, 83, 87, 94};
	int idx, x, i, jmlData = 10;

	cout<<"Elemen Array: ";
	for(i = 0; i < jmlData; i++)
	{
		cout<<data[i]<<" ";
	}
	cout<<endl;

	cout<<"Masukkan data yang akan dicari: ";
	cin>>x;

	BinSearch(data, jmlData, x, &idx);
	if(idx != -1)
	{
		cout<<"Data yang dicari berada pada indeks: "<<idx<<endl;
	}
	else
	{
		cout<<"Data yang dicari tidak ada dalam array"<<endl;
	}

}

void BinSearch(int data[], int n, int x, int *idx)
{
```

```
	bool ketemu = false;
	int top = n-1, bottom = 0, mid;
	int i = 0;
	while(bottom <= top && !ketemu)
	{
		mid = (top + bottom) / 2;

		if(data[mid] == x)
		{
			ketemu = true;
		}
		else
		{
			if(data[mid] > x)
			{
				top = mid - 1;
			}
			else
			{
				bottom = mid + 1;
			}
		}
	}

	if(ketemu)
	{
		*idx = mid;
	}
	else
	{
		*idx = -1;
	}
}
```