**TUGAS PERTEMUAN 12**

**STRUKTUR DATA**

**BINARY SEARCH**



**Disusun oleh:**

**Rama Pramudya Wibisana 2022320019**

**PROGRAM STUDI SISTEM INFORMASI**

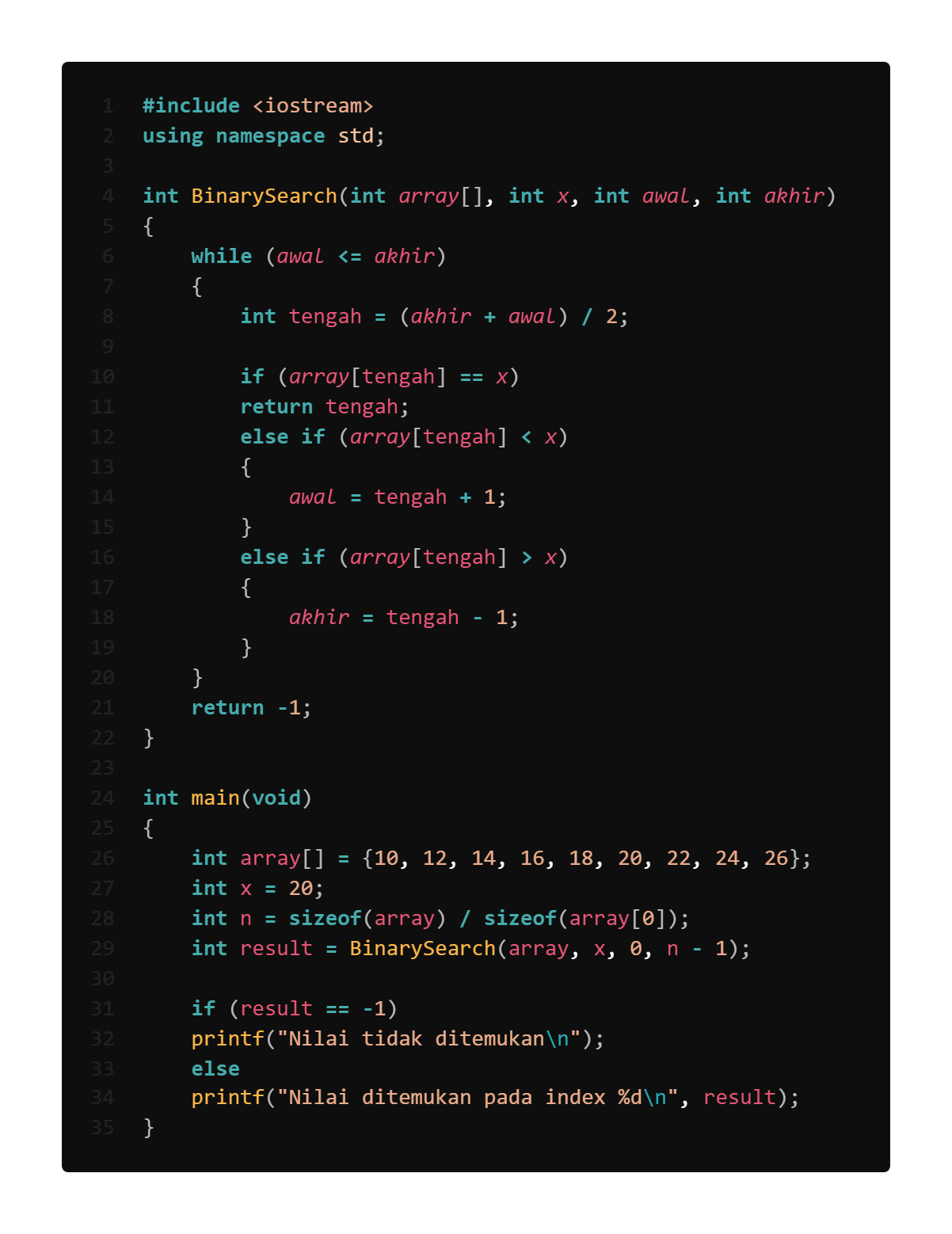
**FAKULTAS INFORMATIKA**

**UNIVERSITAS BINA INSANI**

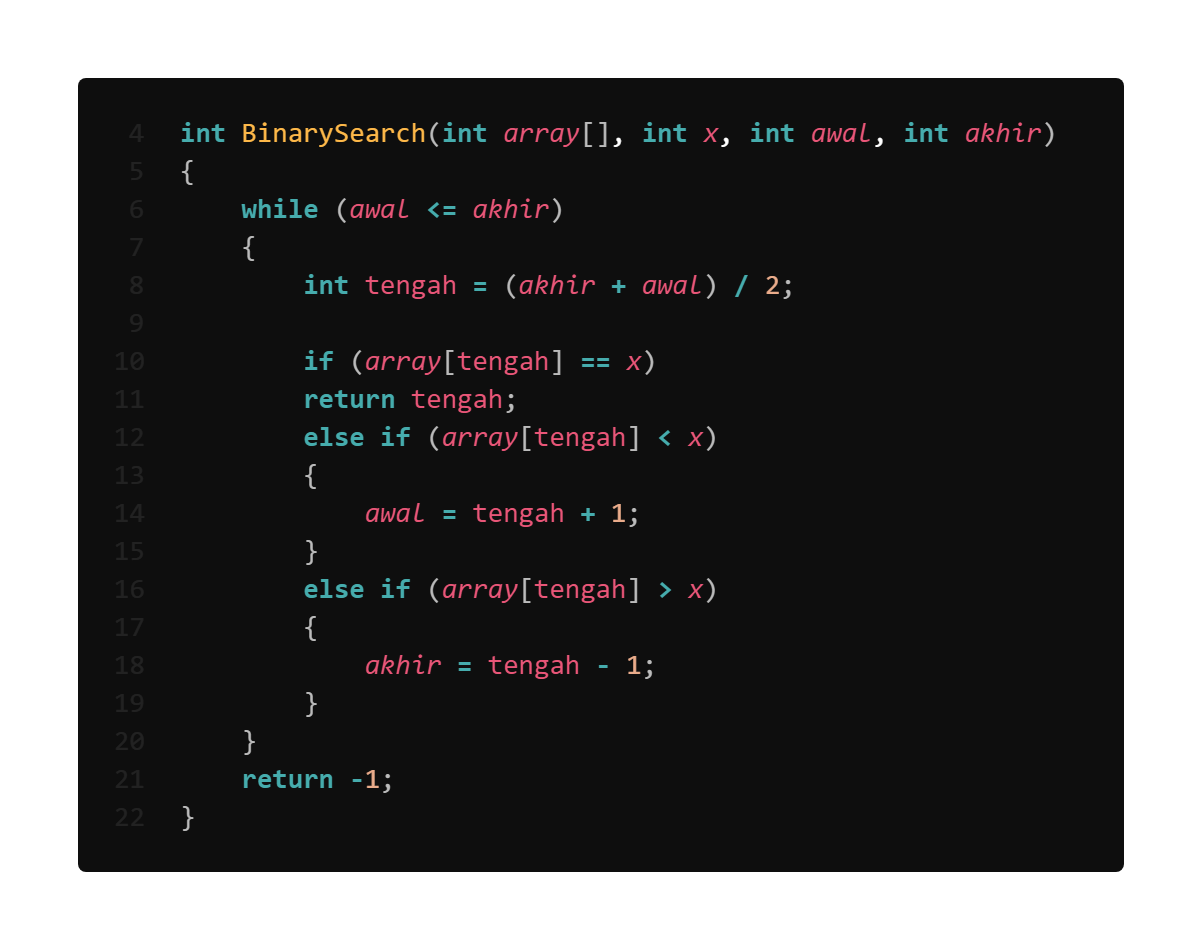
**BEKASI**

**2023**

1. **SYNTAX**

****

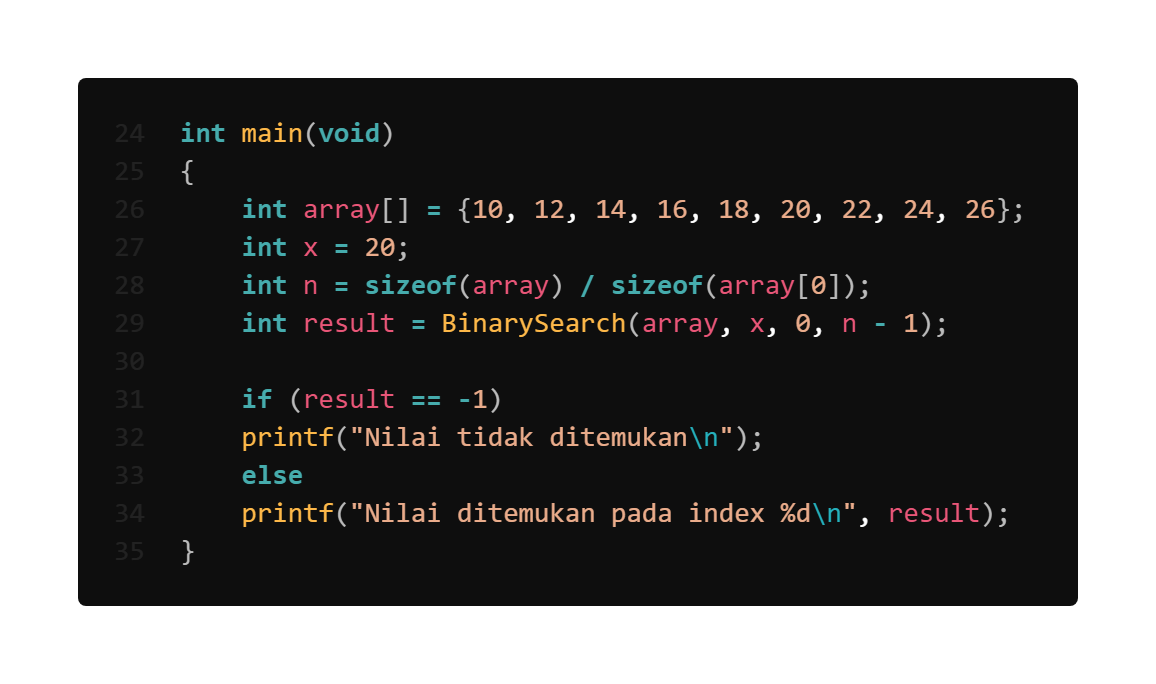
1. **PENJELASAN**

Pada tugas kali ini, kita akan membahas sedikit mengenai struktur searching pada c++. Kali ini yang akan kita gunakan sebagai contoh yakni Binary Search. Mari kita langsung masuk ke pembahasan mengenai Binary Search.

Kita membuat fungsi ***BinarySearch*** yang bertipe data integer, yang memiliki 3 parameter, yaitu ***array[]***, ***x***, ***awal***, dan ***akhir*** yang masing-masing memiliki tipe data integer.

Selanjutnya kita akan melakukan perulangan ***while*** untuk menentukan nilai awal kurang dari sama dengan nilai akhir. Lalu kita membuat variabel ***tengah*** dengan tipe data integer yang melakukan operasi ***(akhir + awal) / 2*** untuk menentukan nilai tengah dari index.

Kemudian kita membuat percabangan, jika ***array[tengah]*** sama dengan ***x*** maka ia akan kembali ke nilai tengah, jika ***array[tengah]*** kurang dari ***x*** maka akan menentukan nilai awal dengan menjalankan operasi ***tengah + 1***, jika ***array[tengah]*** lebih dari ***x*** maka akan menentukan nilai akhir dengan menjalanan operasi ***tengah – 1***.

Kita berlanjut ke fungsi utama, di sini kita akan menentukan array-nya yang menggunakan tipe data integer, memiliki 9 nilai dan 8 index ***{10, 12, 14, 16, 18, 20, 22, 24, 26}***. Lalu kita akan menentukan nilai yang akan kita cari dalam index, di sini kita menentukan ***x = 20***. Kemudian kita akan menentukan nilai ***n***, yaitu jumlah ukuran dari index array di atas ***[8]***. Setelah itu akan menampilkan hasilnya dengan memanggil fungsi ***BinarySearch***.

Maka dapat ditentukan :

* nilai awal = 0
* nilai akhir = n – 1 = 8
* nilai tengah = (8 + 0) / 2 = 4
* nilai tengah terdapat pada index ke-4 yaitu 18

Kemudian bilangan dibandingkan :

* memanggil operasi ***(array[tengah] == x)***, maka 18 = 20 -> false,

nilai belum sesuai

* memanggil operasi ***(array[tengah] < x)***, maka 18 < 20 -> true,

nilai belum sesuai

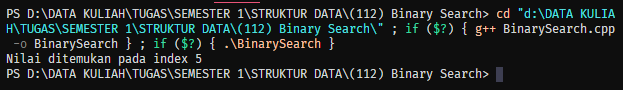
Pencarian dilakukan ke sebelah kiri dengan nilai awal = tengah + 1 yaitu index ke-5, karena nilai awal tadi masih lebih kecil, maka perulangan akan dilakukan kembali, maka :

* memanggil operasi ***(array[tengah] == x)***, maka 20 = 20 -> true,

nilai sudah sesuai

Nilai sudah ditemukan pada index ke-5 yaitu 20, sesuai dengan apa yang kita cari, maka proses pencarian pun dihentikan dan akan menampilkan output seperti ini :

1. **OUTPUT**

****