

Nama : Kanaya Dzikra Setiadi

NRP : 2C2230015

Prodi : Sains Data

Program tersebut ditulis dalam bahasa pemrograman C++ dan bertujuan untuk mendemonstrasikan penggunaan pointer

- `#include <iostream>`

Baris ini menyertakan library `iostream` yang menyediakan fungsi untuk input dan output data, seperti `std::cout` yang digunakan untuk mencetak ke konsol.

- `int main() {`

Fungsi `main` adalah titik awal program di C++. Semua kode yang ingin dijalankan program harus ditulis di dalam fungsi `main`.

- `int a = 123;`
- `int b = 456;`

Baris ini mendeklarasikan dua variabel integer, `a` dan `b`, dan memberikan nilai awal masing-masing 123 dan 456.

- `int *ptr_b = &b;`

Baris ini mendeklarasikan pointer `ptr_b` yang menyimpan alamat memori dari variabel `b`. Tanda `*` di depan tipe data `int` menunjukkan bahwa `ptr_b` adalah pointer yang menunjuk ke variabel integer. `&b` adalah operator address-of yang mengembalikan alamat memori dari variabel `b`.

- `// print isi variabel dan alamat memori`
- `std::cout << "Nama Variabel \t Alamat \t Konten" << std::endl;`
- `std::cout << "a \t\t " << &a << "\t" << a << std::endl;`
- `std::cout << "b \t\t " << &b << "\t" << b << std::endl;`
- `std::cout << "ptr_b \t\t " << &ptr_b << "\t" << ptr_b << std::endl;`
- `std::cout << "*ptr_b \t " << ptr_b << "\t" << *ptr_b << std::endl;`

Blok kode ini mencetak isi variabel dan alamat memorinya ke konsol.

- `std::cout` digunakan untuk mencetak ke konsol.
- `\t` digunakan untuk membuat tab pada keluaran.
- `<<` digunakan untuk memasukkan data ke dalam stream output (`std::cout`).
- Baris pertama mencetak header berupa "Nama Variabel", "Alamat", dan "Konten".
- Baris selanjutnya mencetak nilai `a`, alamat memori `a`, nilai `b`, alamat memori `b`, alamat memori `ptr_b` (yang menunjuk ke `b`), dan nilai yang ditunjuk oleh `ptr_b` (yaitu nilai `b`).

keluaran yang dihasilkan:

Nama VariabelAlamatKonten

```
a                0x7fffaaaaaaaa    123
b                0x7fffaaaaaaaa    456
ptr_b            0x7fffaaaaaaaa    0x7fffaaaaaaaa // Alamat memori dari variabel b
*ptr_b           0x7fffaaaaaaaa    456           // Nilai yang ditunjuk oleh ptr_b (yaitu nilai b)
```

Setelah pencetakan, program mengubah nilai yang ditunjuk oleh pointer `ptr_b` menjadi 789:

- `*ptr_b = 789;`

Karena `ptr_b` menunjuk ke variabel `b`, perubahan ini sebenarnya mengubah nilai `b` menjadi 789.

Terakhir, program mencetak isi `b` dan nilai yang ditunjuk oleh `ptr_b` sekali lagi:

- `std::cout << "b \t\t " << &b << "\t" << b << std::endl;`
- `std::cout << "*ptr_b \t\t " << ptr_b << "\t" << *ptr_b << std::endl;`

Keluaran yang dihasilkan setelah perubahan:*

```
b                0x7fffaaaaaaaa    789
*ptr_b           0x7fffaaaaaaaa    789
```

Kesimpulan:

Program ini menunjukkan bagaimana pointer dapat digunakan untuk mengakses dan memodifikasi nilai variabel secara tidak langsung. Dengan menggunakan pointer, kita dapat menghindari kebutuhan untuk selalu menyebutkan nama variabel dan membuat program lebih fleksibel.