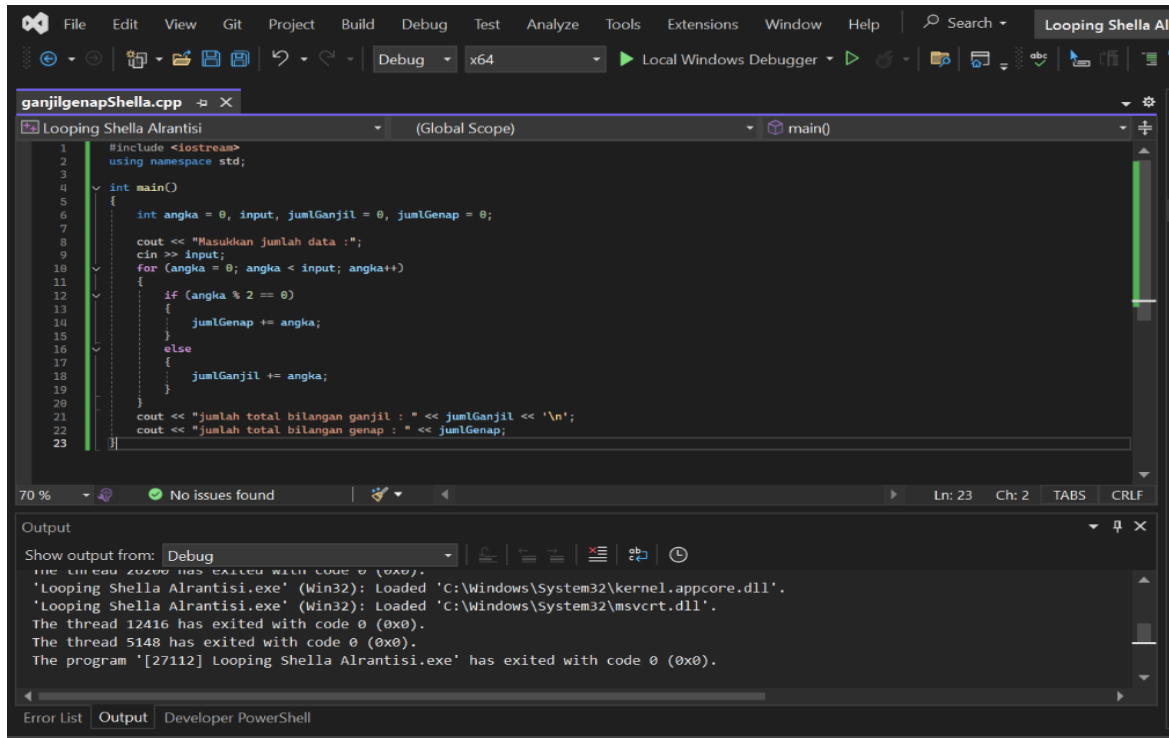


Nama : Shella Alrantisi

NIM : 24104410060

Prodi : Teknik Informatika

PENJELASAN



```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int angka = 0, input, jumlGanjil = 0, jumlGenap = 0;
7
8     cout << "Masukkan jumlah data :";
9     cin >> input;
10    for (angka = 0; angka < input; angka++)
11    {
12        if (angka % 2 == 0)
13        {
14            jumlGenap += angka;
15        }
16        else
17        {
18            jumlGanjil += angka;
19        }
20    }
21    cout << "jumlah total bilangan ganjil : " << jumlGanjil << '\n';
22    cout << "jumlah total bilangan genap : " << jumlGenap;
23 }
```

Output

Show output from: Debug

The thread 4200 has exited with code 0 (0x0).
'Looping Shella Alrantisi.exe' (Win32): Loaded 'C:\Windows\System32\kernel.appcore.dll'.
'Looping Shella Alrantisi.exe' (Win32): Loaded 'C:\Windows\System32\msvcrt.dll'.
The thread 12416 has exited with code 0 (0x0).
The thread 5148 has exited with code 0 (0x0).
The program '[27112] Looping Shella Alrantisi.exe' has exited with code 0 (0x0).

1. #include <iostream>

- Direktif `#include` digunakan untuk menyertakan header file yang berisi deklarasi untuk objek dan fungsi yang digunakan untuk input/output standar.
- `<iostream>` adalah header file yang memungkinkan penggunaan objek `cin` (untuk input) dan `cout` (untuk output).

2. Using namespace std;

- Ini digunakan untuk menghindari penulisan `std::` sebelum objek atau fungsi dari namespace standar, seperti `std::cout`, `std::cin`, dan lainnya.
- Misalnya, kita bisa menulis `cout` tanpa perlu menulis `std::cout`.

3. Fungsi Utama (int main)

Titik awal eksekusi program. Fungsi `main` akan dijalankan ketika program dieksekusi.

4. Deklarasi Variabel

```
6     int angka = 0, input, jumlGanjil = 0, jumlGenap = 0;
```

`int angka = 0, input, jumlGanjil = 0, jumlGenap = 0;`

- Angka: Variabel untuk menghitung iterasi dalam loop.
- input: Variabel untuk menyimpan jumlah data yang akan diinput oleh pengguna.
- jumlahGanjil: Menyimpan total jumlah bilangan ganjil.
- jumlahGenap: Menyimpan total jumlah bilangan genap.

5. Input dari User

```
8      cout << "Masukkan jumlah data :";
9      cin >> input;
```

cout << "Masukkan jumlah data : "; cin >> input;

Program meminta pengguna untuk memasukkan jumlah data yang ingin dihitung. Nilai ini disimpan dalam variabel input.

6. Loop untuk Menghitung Bilangan Genap dan Ganjil

```
11
12      if (angka % 2 == 0)
13      {
14          jumlahGenap += angka;
15      }
16      else
17      {
18          jumlahGanjil += angka;
19      }
20
```

For (angka = 0; angka < input; angka++)

Loop ini berjalan dari 0 hingga kurang dari input.

• Pengkondisian dalam loop

- ☐ Jika angka genap (diperiksa menggunakan `angka % 2 == 0`), maka angka ditambahkan ke jumlahGenap.
- ☐ Jika angka ganjil (sebaliknya), maka angka ditambahkan ke jumlahGanjil.
- ☐ Variabel tambah dihitung tetapi tidak digunakan lebih lanjut dalam program ini.

7. Output Hasil

Setelah loop selesai, program menampilkan total bilangan ganjil dan genap yang telah dihitung.

```
ganjilGenapShella.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int input;
7     cout << "Masukkan jumlah data :";
8     cin >> input;
9
10    int angka = 0;
11    int jumlahGanjil = 0;
12    int jumlahGenap = 0;
13    while (angka < input)
14    {
15        if (angka % 2 == 0)
16        {
17            jumlahGenap += angka;
18        }
19        else
20        {
21            jumlahGanjil += angka;
22        }
23        angka++;
24    }
25    cout << "jumlah total bilangan ganjil : " << jumlahGanjil << endl;
26    cout << "jumlah total bilangan genap : " << jumlahGenap << endl;
27    return 0;
28 }
```

Output:

```
Masukkan jumlah data :10
jumlah total bilangan ganjil : 25
jumlah total bilangan genap : 20
C:\Users\LENOVO\source\repos\Looping Shella Alrantisi\x64\Debug\Looping Shella Alrantisi.exe (process 24288) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Kesimpulan :

- a. Kita membuat looping dengan batas yang disesuaikan (sesuai dengan yang di-inputkan) dan hasilnya akan ditampilkan di konsol.
- b. Membuat if else dengan format, jika sebuah angka dapat dibagi dengan angka 2, maka akan masuk ke dalam true/genap. Sebaliknya, jika hasilnya false, akan dimasukkan ke dalam ganjil.
- c. Terakhir, hasil penjumlahan dari semua nilai genap dan nilai ganjil sesuai dengan inputan akan ditampilkan.