

LAPORAN
ALGORITMA DAN PEMROGRAMAN I



**SOFTWARE ARCHITECTURE
AND QUALITY LABORATORY**

NAMA : Berliana Sela Ryani

NIM : 202531007

KELAS : A

DOSEN : Dr.Meilia Nur Indah Susanti, ST, M.Kom

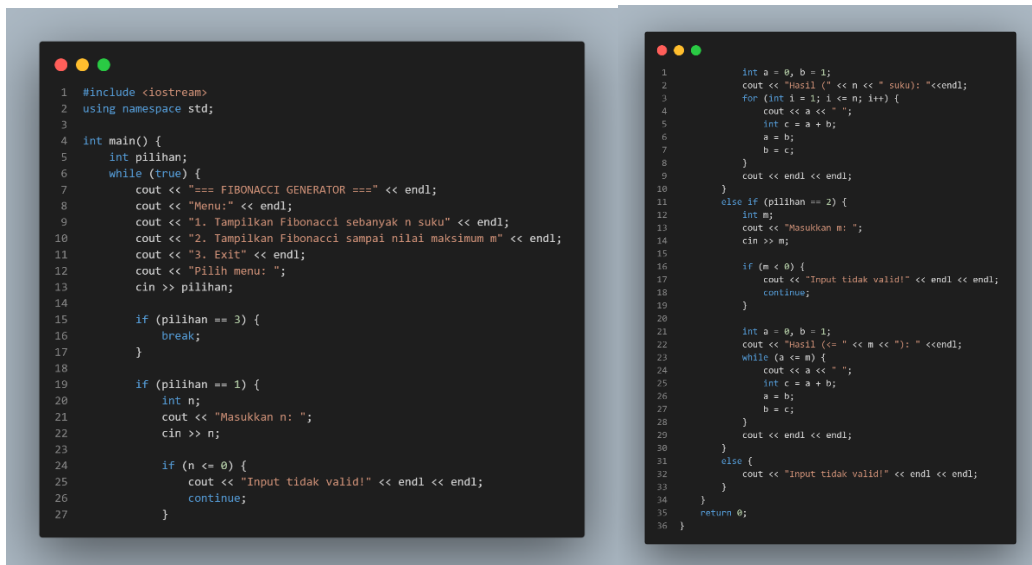
NO.PC : 11

ASISTEN : 1. Adhiwira Wildan Pratama
2. Petra Manullang
3. Rafi Indra Pramudhito Zuhayr
4. Tasya Kartika Sari

FAKULTAS TELEMATIKA ENERGI
TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI PLN – JAKARTA

2024/2025

SOAL 1



PENJELASAN : Program C++ ini menampilkan menu interaktif pembangkit deret Fibonacci yang dirancang agar mahasiswa dapat menggunakan program dengan mudah dan terstruktur, di mana pengguna diberikan beberapa pilihan menu untuk menampilkan deret Fibonacci berdasarkan jumlah suku (n) atau hingga batas nilai maksimum tertentu (m) sesuai kebutuhan. Program ini juga dilengkapi dengan pengecekan input sehingga dapat mencegah kesalahan saat pengguna memasukkan data yang tidak valid, serta menggunakan perulangan agar menu dapat ditampilkan kembali dan program bisa digunakan berulang kali sampai pengguna memilih menu Exit untuk mengakhiri program.

OUTPUT :

```

=== FIBONACCI GENERATOR ===
Menu:
1. Tampilkan Fibonacci sebanyak n suku
2. Tampilkan Fibonacci sampai nilai maksimum m
3. Exit
Pilih menu: 1
Masukkan n: 8
Hasil (8 suku):
0 1 1 2 3 5 8 13

```

Penjelasan : Tampilan awal program Fibonacci Generator yang menampilkan menu pilihan. Pengguna memilih menu 1, yaitu menampilkan deret Fibonacci berdasarkan jumlah suku (n). Setelah itu pengguna memasukkan nilai n = 8, sehingga program menghitung dan menampilkan 8 suku pertama deret Fibonacci secara berurutan, dimulai dari 0, 1, 1, 2, 3, 5, 8, hingga 13. Hasil ini menunjukkan bahwa program berhasil menampilkan deret sesuai jumlah suku yang diminta dan perhitungan berjalan dengan benar.

```
=== FIBONACCI GENERATOR ===
Menu:
1. Tampilkan Fibonacci sebanyak n suku
2. Tampilkan Fibonacci sampai nilai maksimum m
3. Exit
Pilih menu: 2
Masukkan m: 20
Hasil (<= 20):
0 1 1 2 3 5 8 13
```

Penjelasan : Menampilkan menu karena menggunakan perulangan. Pengguna kemudian memilih **menu 2**, yaitu menampilkan deret Fibonacci hingga **nilai maksimum tertentu (m)**. Setelah pengguna memasukkan **m = 20**, program menampilkan semua bilangan Fibonacci yang nilainya **tidak melebihi 20**. Deret yang ditampilkan berhenti pada angka 13 karena angka berikutnya sudah lebih besar dari batas yang ditentukan. Hal ini menunjukkan bahwa program mampu membatasi output sesuai nilai maksimum yang dimasukkan oleh pengguna.

```
=== FIBONACCI GENERATOR ===
Menu:
1. Tampilkan Fibonacci sebanyak n suku
2. Tampilkan Fibonacci sampai nilai maksimum m
3. Exit
Pilih menu: -3
Input tidak valid!
```

Penjelasan : Menunjukkan pilihan menu yang tidak tersedia, yaitu angka -3. Karena pilihan tersebut tidak sesuai dengan menu yang ada, program menampilkan pesan "Input tidak valid!". Kondisi ini menunjukkan bahwa program telah dilengkapi dengan validasi input, sehingga dapat menangani kesalahan pengguna dengan memberikan peringatan, bukan langsung berhenti atau menghasilkan output yang salah..

SOAL 2

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int kode = 7;
6      int tebak;
7
8      for (int i = 1; i <= 3; i++) {
9          cout << "percobaan " << i << "/" << 3 << " - Tebak : ";
10         cin >> tebak;
11
12         if (tebak == kode) {
13             cout << "Menang!" << endl;
14             return 0;
15         } else {
16             cout << "Salah!" << endl;
17         }
18     }
19
20     cout << "Kalah! Angka = 7" << endl;
21     return 0;
22 }
```

PENJELASAN : Program C++ ini mensimulasikan permainan tebak angka sederhana yang dirancang untuk melatih pemahaman mahasiswa terhadap konsep percabangan dan perulangan, di

mana pengguna diberikan maksimal 3 kali kesempatan untuk menebak angka rahasia yaitu 7; setiap tebakan akan langsung dicek, jika tebakan benar maka permainan langsung berakhir dengan kemenangan, namun jika tebakan salah maka pengguna dapat mencoba lagi hingga kesempatan habis, dan apabila ketiga percobaan tersebut gagal maka program akan menampilkan pesan kalah serta memberitahukan angka yang sebenarnya.

OUTPUT :

```
percobaan 1/3 - Tebak : 5
Salah!
percobaan 2/3 - Tebak : 7
Menang!
```

Penjelasan : Menunjukkan kondisi ketika pengguna melakukan percobaan pertama dengan menebak angka 5, namun tebakan tersebut tidak sesuai dengan angka yang ditentukan sehingga program menampilkan pesan “Salah!”. Pada percobaan kedua, pengguna memasukkan angka 7 yang sama dengan angka rahasia, sehingga program langsung menampilkan pesan “Menang!” dan permainan berakhir tanpa melanjutkan ke percobaan berikutnya.

```
percobaan 1/3 - Tebak : 1
Salah!
percobaan 2/3 - Tebak : 2
Salah!
percobaan 3/3 - Tebak : 3
Salah!
Kalah! Angka = 7
```

Penjelasan : Menunjukkan kondisi ketika pengguna menggunakan seluruh kesempatan menebak sebanyak tiga kali, yaitu dengan angka 1, 2, dan 3, namun semua tebakan tersebut salah. Karena tidak ada tebakan yang benar hingga percobaan terakhir, program menampilkan pesan “Kalah! Angka = 7” sebagai tanda bahwa permainan telah selesai dan menunjukkan angka yang seharusnya ditebak.

SOAL 3

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     cout << "Masukkan jumlah mahasiswa: ";
7     cin >> n;
8
9     int hadir = 0, tidakHadir = 0, data;
10
11     for (int i = 1; i <= n; i++) {
12         cout << "Mahasiswa ke-" << i << " (1=hadir, 0=tidak): ";
13         cin >> data;
14
15         if (data == 1)
16             hadir++;
17         else
18             tidakHadir++;
19     }
20     cout<<endl;
21
22     double persen = (double)hadir / n * 100;
23     persen = (int)(persen * 100 + 0.5) / 100.0;
24
25     cout << "Jumlah hadir: " << hadir << endl;
26     cout << "Jumlah tidak hadir: " << tidakHadir << endl;
27     cout << "Persentase hadir: " << persen << "%" << endl;
28
29     return 0;
30 }
31
```

PENJELASAN : Program C++ ini digunakan untuk menghitung jumlah serta persentase kehadiran mahasiswa dengan cara meminta pengguna memasukkan status kehadiran setiap mahasiswa (hadir atau tidak hadir) satu per satu sesuai jumlah yang ditentukan, kemudian data tersebut diolah untuk mengetahui total mahasiswa yang hadir dan tidak hadir, serta dihitung persentase kehadiran secara otomatis, sehingga hasil akhirnya dapat ditampilkan dengan jelas, rapi, dan mudah dipahami sebagai informasi kehadiran kelas.

OUTPUT :

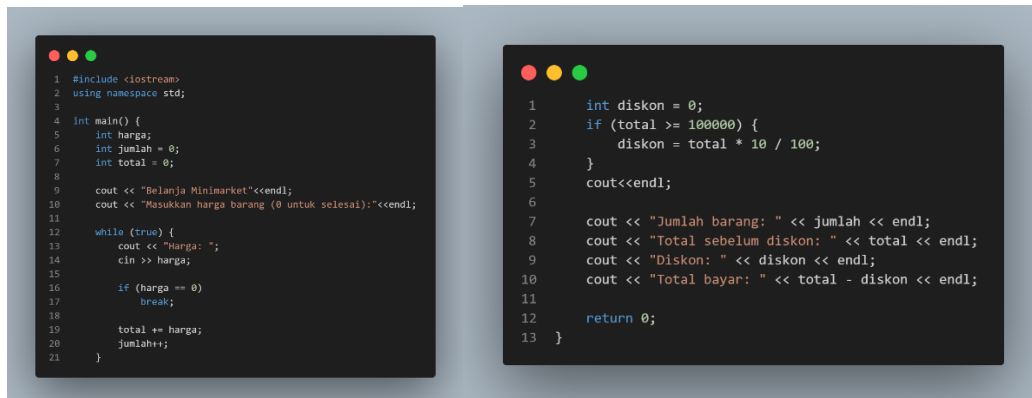
```
Masukkan jumlah mahasiswa: 6
Mahasiswa ke-1 (1=hadir, 0=tidak): 1
Mahasiswa ke-2 (1=hadir, 0=tidak): 0
Mahasiswa ke-3 (1=hadir, 0=tidak): 1
Mahasiswa ke-4 (1=hadir, 0=tidak): 1
Mahasiswa ke-5 (1=hadir, 0=tidak): 0
Mahasiswa ke-6 (1=hadir, 0=tidak): 1

Jumlah hadir: 4
Jumlah tidak hadir: 2
Persentase hadir: 66.67%
```

Penjelasan : Menunjukkan bahwa pengguna memasukkan jumlah mahasiswa sebanyak 6 orang, kemudian program meminta pengguna untuk mengisi data kehadiran setiap mahasiswa satu per satu dengan kode 1 untuk hadir dan 0 untuk tidak hadir. Dari data yang dimasukkan, terdapat 4 mahasiswa yang hadir dan 2 mahasiswa yang tidak hadir, sehingga program secara otomatis menghitung total kehadiran serta jumlah mahasiswa yang tidak hadir, lalu menghitung persentase kehadiran berdasarkan perbandingan jumlah hadir dengan total mahasiswa, dan menampilkan hasil

akhir berupa persentase kehadiran sebesar 66,67% yang sudah dibulatkan agar lebih mudah dibaca dan dipahami.

SOAL 4



PENJELASAN : Program C++ ini mensimulasikan kasir minimarket sederhana yang dirancang untuk membantu pengguna mencatat proses belanja, di mana pengguna dapat memasukkan harga barang satu per satu dan menghentikan input ketika selesai berbelanja, kemudian program akan menghitung jumlah barang yang dibeli serta total harga keseluruhan, memberikan diskon sebesar 10% secara otomatis jika total belanja mencapai atau melebihi Rp100.000, dan akhirnya menampilkan ringkasan pembayaran secara lengkap, jelas, dan mudah dipahami agar pengguna mengetahui total yang harus dibayar.

OUTPUT :

```
Belanja Minimarket

Masukkan harga barang (0 untuk selesai):
Harga: 20000
Harga: 30000
Harga: 60000
Harga: 0

Jumlah barang: 3
Total sebelum diskon: 110000
Diskon: 11000

Total bayar: 99000
```

Penjelasan : Menunjukkan proses simulasi belanja di minimarket, di mana pengguna memasukkan harga barang satu per satu, yaitu Rp20.000, Rp30.000, dan Rp60.000, kemudian mengakhiri proses input dengan memasukkan angka 0 sebagai tanda bahwa tidak ada barang tambahan. Berdasarkan input tersebut, program menghitung jumlah barang yang dibeli sebanyak 3 item dan menjumlahkan seluruh harga sehingga diperoleh total belanja sebesar Rp110.000. Karena total belanja melebihi Rp100.000, program secara otomatis memberikan diskon sebesar 10%, yaitu Rp11.000, lalu menghitung dan menampilkan total pembayaran akhir sebesar Rp99.000, sehingga pengguna dapat melihat dengan jelas rincian belanja, potongan harga, dan jumlah uang yang harus dibayarkan.

SOAL 5

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n;
6      cout << "Masukkan jumlah hari: ";
7      cin >> n;
8
9      int uang, total = 0;
10     for (int i = 1; i <= n; i++) {
11         cout << "Uang saku hari ke-" << i << ": ";
12         cin >> uang;
13         total += uang;
14     }
15
16     cout << "Total uang saku: " << total << endl;
17     return 0;
18 }
19
```

PENJELASAN : Program C++ ini digunakan untuk menghitung dan menjumlahkan total uang saku selama beberapa hari, di mana pengguna terlebih dahulu memasukkan jumlah hari, kemudian program meminta input uang saku untuk setiap hari secara berurutan. Setiap nilai uang saku yang dimasukkan akan dijumlahkan menggunakan proses perulangan, sehingga pada akhir program ditampilkan total uang saku keseluruhan, yang disajikan secara sederhana, jelas, dan mudah dipahami, sehingga membantu mahasiswa memahami konsep penjumlahan data menggunakan perulangan dalam pemrograman C++.

OUTPUT :

```
Masukkan jumlah hari: 4
Uang saku hari ke-1: 10000
Uang saku hari ke-2: 15000
Uang saku hari ke-3: 12000
Uang saku hari ke-4: 8000
Total uang saku: 45000
```

Penjelasan : Menunjukkan proses penghitungan total uang saku secara bertahap selama beberapa hari, di mana pengguna terlebih dahulu memasukkan jumlah hari sebanyak 4 hari. Setelah itu, program meminta pengguna untuk menginput uang saku setiap hari satu per satu, yaitu Rp10.000 pada hari pertama, Rp15.000 pada hari kedua, Rp12.000 pada hari ketiga, dan Rp8.000 pada hari keempat. Setiap nilai uang saku yang dimasukkan akan langsung dijumlahkan oleh program, sehingga pada akhir proses ditampilkan total uang saku sebesar Rp45.000, yang merupakan hasil penjumlahan dari seluruh uang saku harian dan ditampilkan dengan jelas agar mudah dipahami oleh pengguna.