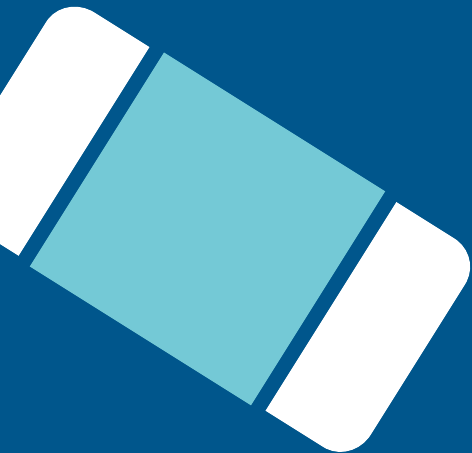
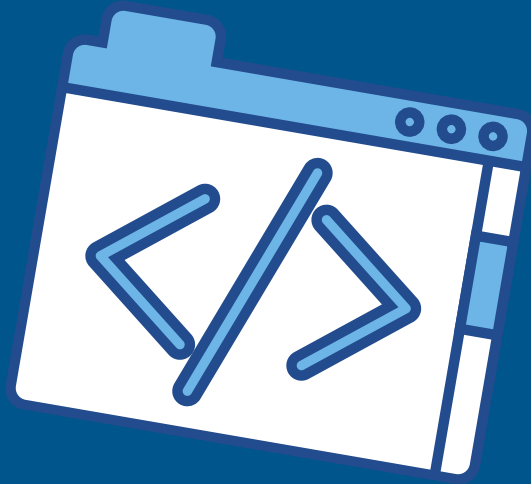
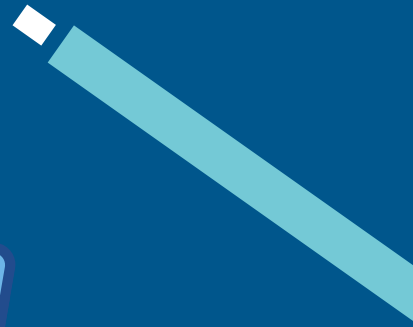
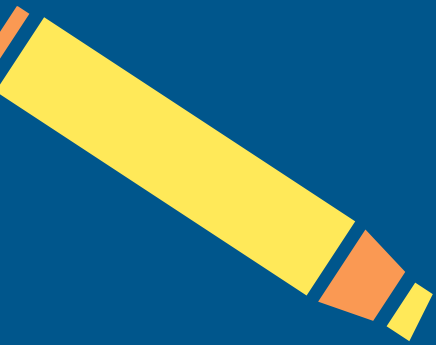


PROSEDUR



Perhatikan contoh berikut!

Terdapat sebuah program yang menampilkan menu pada ATM sebagai berikut:

```
int main() {  
    cout<< " 1. CEK SALDO"<<endl;  
    cout<< " 2. PENARIKAN TUNAI"<<endl;  
    cout<< " 3. SETOR TUNAI"<<endl;  
    cout<< " 4. KELUAR"<<endl;  
}
```

Sebagai contoh, menu ATM tersebut harus dicetak secara berulang kali pada beberapa tempat yang berbeda yaitu pada **halaman awal** serta **halaman pemilihan menu**

```
int main() {  
    //halaman awal  
  
    cout<< " 1. CEK SALDO"<<endl;  
    cout<< " 2. PENARIKAN TUNAI"<<endl;  
    cout<< " 3. SETOR TUNAI"<<endl;  
    cout<< " 4. KELUAR"<<endl;  
  
    //halaman pemilihan menu  
  
    cout<< " 1. CEK SALDO"<<endl;  
    cout<< " 2. PENARIKAN TUNAI"<<endl;  
    cout<< " 3. SETOR TUNAI"<<endl;  
    cout<< " 4. KELUAR"<<endl;  
}
```

Kode program untuk mencetak menu tersebut harus dituliskan secara **berulang-ulang**.

Berdasarkan contoh tersebut dalam program komputer seringkali terdapat bagian program yang ingin digunakan kembali sehingga perlu **dituliskan secara berulang**. Pada contoh program yang bertugas menampilkan menu dituliskan secara berulang padahal melakukan aktivitas yang sama.

Ketika program semakin besar dan kompleks, maka jika kode program yang sama dituliskan secara berulang-ulang membuat program menjadi tidak efektif karena semakin besar program maka akan semakin kompleks dan sulit melacak kesalahan.

Pada program, suatu proses utama yang kompleks dapat dibagi ke dalam beberapa **prosedur** berdasarkan tugasnya. Dengan menggunakan prosedur kode program yang sama tidak perlu dituliskan secara berulang, program hanya perlu dikelompokkan ke dalam prosedur dan prosedur dapat dipanggil secara berulang kali sesuai dengan kebutuhan.

Contoh:

```
int main(){  
    //halaman awal  
    menu();  
  
    //halaman pemilihan menu  
    menu();  
}
```

Contoh:

```
int main(){  
    string str1, str2;  
    str1 = "Maya";  
    cout << "Hello " << str1 << endl;  
  
    cout << "Hello " << "Joko" << endl;  
  
    str2 = "Dini";  
    cout << "Hello " << str2 << endl;  
}
```

Diganti dengan sub program (**prosedur**)

```
int main(){  
    string str1, str2;  
    str1 = "Maya";  
    cetak(str1);  
  
    cetak("Joko");  
  
    str3 = "Dini";  
    cetak(str3);  
}
```

Perhatikan ilustrasi berikut!

Ketika seseorang akan pergi ke luar negeri maka perlu mengikuti serangkaian prosedur berikut:

Prosedur Pergi ke luar Negeri

A. Prosedur Urus Paspor

- Isi formulir permintaan paspor dengan lampiran fotocopy KTP, Kartu Keluarga dan Pas Foto
- Serahkan formulir yang sudah diisi beserta biaya pembuatan paspor
- Wawancara dengan petugas imigrasi
- Terima paspor

B. Prosedur Urus Visa

- Isi formulir permohonan visa dengan lampiran fotocopy KTP, paspor, Pas Foto, dan tiket pesawat terbang
- Serahkan formulir yang sudah diisi beserta biaya pembuatan visa
- Terima Visa

Pada contoh tersebut, dalam prosedur pergi ke luar negeri terdapat beberapa prosedur didalamnya. Sehingga instruksi tersebut dikelompokkan ke dalam prosedur berdasarkan tugasnya. Yaitu prosedur urus paspor yang bertugas untuk mengurus paspor dan prosedur urus visa yang bertugas untuk mengurus visa

```
ALGORITMA pergi ke luar negeri  
urusVisa();  
urusPaspor();
```

PROSEDUR

Berdasarkan beberapa contoh dan ilustrasi tersebut dapat didefinisikan bahwa

Prosedur adalah modul atau sub program yang merupakan bagian dari program utama yang bertugas untuk mengerjakan tugas tertentu dan tidak mengembalikan suatu nilai.

Penggunaan prosedur pada program dapat **mengurangi duplikasi kode**. Karena untuk aktivitas yang harus dilakukan lebih dari sekali, cukup ditulis sekali sehingga dapat mengurangi panjang program.

Selain itu dapat mempermudah **melacak kesalahan (*debug program*)**, karena hanya perlu memeriksa kesalahan pada prosedur.



PROSEDUR PADA PROGRAM

Untuk membuat prosedur pada program maka perlu dilakukan **pendefinisian prosedur** agar prosedur dapat dikenali oleh program, lalu dilakukan **pemanggilan prosedur** yang telah didefinisikan pada saat dibutuhkan.



PENDEFINISIAN PROSEDUR

Agar prosedur dikenali oleh program maka perlu dilakukan **pendefinisian prosedur**. Biasanya prosedur didefinisikan pada bagian atas program sebelum program utama. Adapun sintaks pendefinisian prosedur adalah sebagai berikut:

```
void nama Prosedur (deklarasi parameter, jika  
ada) {  
  
    DEKLARASI  
  
    (semua nama yang dipakai dalam prosedur,  
    variabel maupun konstanta)  
  
    ALGORITMA  
  
    (badan prosedur berisi instruksi-instruksi  
    yang dikerjakan oleh prosedur)  
  
}
```

Keterangan:

- **void**: menspesifikasikan tipe data nilai kembalian prosedur, karena prosedur tidak memiliki nilai kembalian maka ditulis dengan keyword **void** yang berarti kosong.
- **Nama prosedur**: aturan pemberian nama pada prosedur secara bebas, dengan mengikuti aturan penamaan variabel
- **Parameter**: variabel yang bertindak untuk menerima nilai masukan pada prosedur, daftar parameter diletakan alam tanda kurung " () ", dibelakang nama fungsi. Parameter pada suatu prosedur boleh ada ataupun tidak.
- **Badan prosedur**: berisi deklarasi dan instruksi yang dikerjakan oleh prosedur. Badan prosedur ditulis dalam tanda kurung kurawal " { } "

CONTOH

Program menampilkan menu ATM pada contoh didefinisikan ke dalam sebuah prosedur

Pendefinsian prosedur

```
void menu() {  
    cout<< " 1. CEK SALDO"<<endl;  
    cout<< " 2. PENARIKAN TUNAI"<<endl;  
    cout<< " 3. SETOR TUNAI"<<endl;  
    cout<< " 4. KELUAR"<<endl;  
}
```

Pada program tersebut didefinisikan prosedur bernama prosedur menu yang bertugas untuk mencetak menu pada program. Prosedur menu () tidak mengembalikan suatu nilai, sehingga tipe data yang diberikan pada nilai kembalian prosedur adalah **void**.

Namun meskipun sudah dilakukan pendefinisian prosedur, ketika program dieksekusi prosedur tersebut tidak dapat langsung dijalankan. Sehingga, agar prosedur tersebut dapat dijalankan maka perlu dilakukan **pemanggilan prosedur**.



PEMANGGILAN PROSEDUR

Prosedur bukan merupakan program yang dapat berdiri sendiri, jadi ia tidak dapat dieksekusi secara langsung. Ini berarti, instruksi-instruksi di dalam prosedur baru dapat dilaksanakan ketika prosedur tersebut diakses.

Prosedur dapat diakses dengan melakukan **pemanggilan prosedur** dari program pemanggil (dapat dari program utama atau dari sub program lainnya). Adapun sintaks pemanggilan prosedur adalah sebagai berikut:

```
namaProsedur (daftar parameter, jika ada);
```

Pemanggilan prosedur dilakukan dengan menuliskan nama prosedur, dan daftar parameter yang ada pada prosedur. Parameter adalah variabel yang terdapat pada prosedur (pembahasan mengenai parameter akan dibahas dalam sub materi selanjutnya)

Pemanggilan prosedur menu

```
void menu() {  
    cout<< " 1. CEK  
    SALDO"<<endl;  
    cout<< " 2. PENARIKAN  
    TUNAI"<<endl;  
    cout<< " 3. SETOR  
    TUNAI"<<endl;  
    cout<< " 4. KELUAR"<<endl;  
}
```

Pendefinisian
Prosedur

```
int main() {  
    menu();  
}
```

Pemanggilan
Prosedur

Karena menu tersebut pada program dicetak dalam beberapa tempat, maka pada program utama kita hanya perlu melakukan pemanggilan prosedur sebanyak yang dibutuhkan.

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

void menu() {
    cout<< " 1. CEK SALDO"<<endl;
    cout<< " 2. PENARIKAN TUNAI"<<endl;
    cout<< " 3. SETOR TUNAI"<<endl;
    cout<< " 4. KELUAR"<<endl;
}

int main() {
    menu();
    menu();
    menu();
}
```

Pada program dengan prosedur, jika terdapat instruksi atau kode program yang sama yang akan digunakan kembali maka tidak perlu dituliskan secara berulang. Cukup dilakukan **pemanggilan pada prosedur tempat kode program tersebut dituliskan secara berulang kali** sesuai dengan kebutuhan.

Program di atas sekaligus menjelaskan bahwa suatu prosedur cukup **didefinisikan satu kali tetapi bisa digunakan beberapa kali**. Sehingga pemakaian prosedur dapat menghindari duplikasi kode dan menghemat penulisan program maupun kode dalam memori.