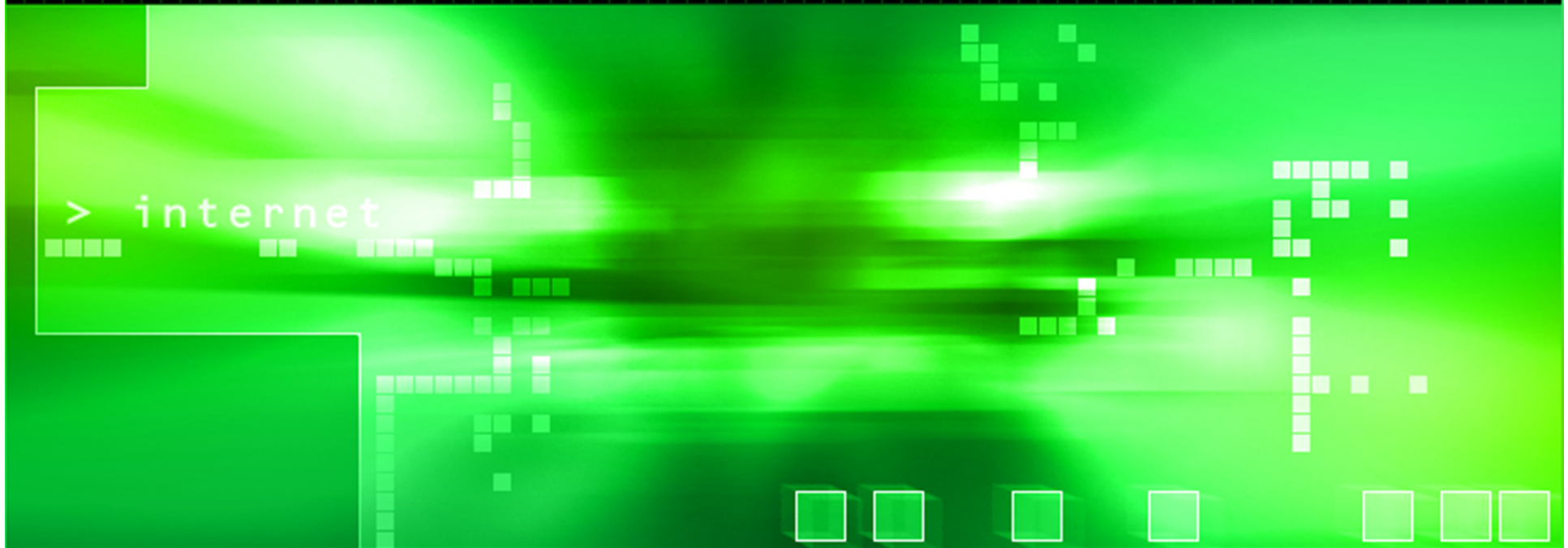


Pemrograman I

Bab VI – Pengulangan



Adam Mukharil Bachtiar, S.Kom.
Universitas Komputer Indonesia

Latar Belakang Pengulangan

1. Tampilkan angka **1 sampai 1000!**
2. Lari keliling lapangan **selama kamu belum pingsan!**
3. Makan baso “pak de” **sampai kamu kenyang!**



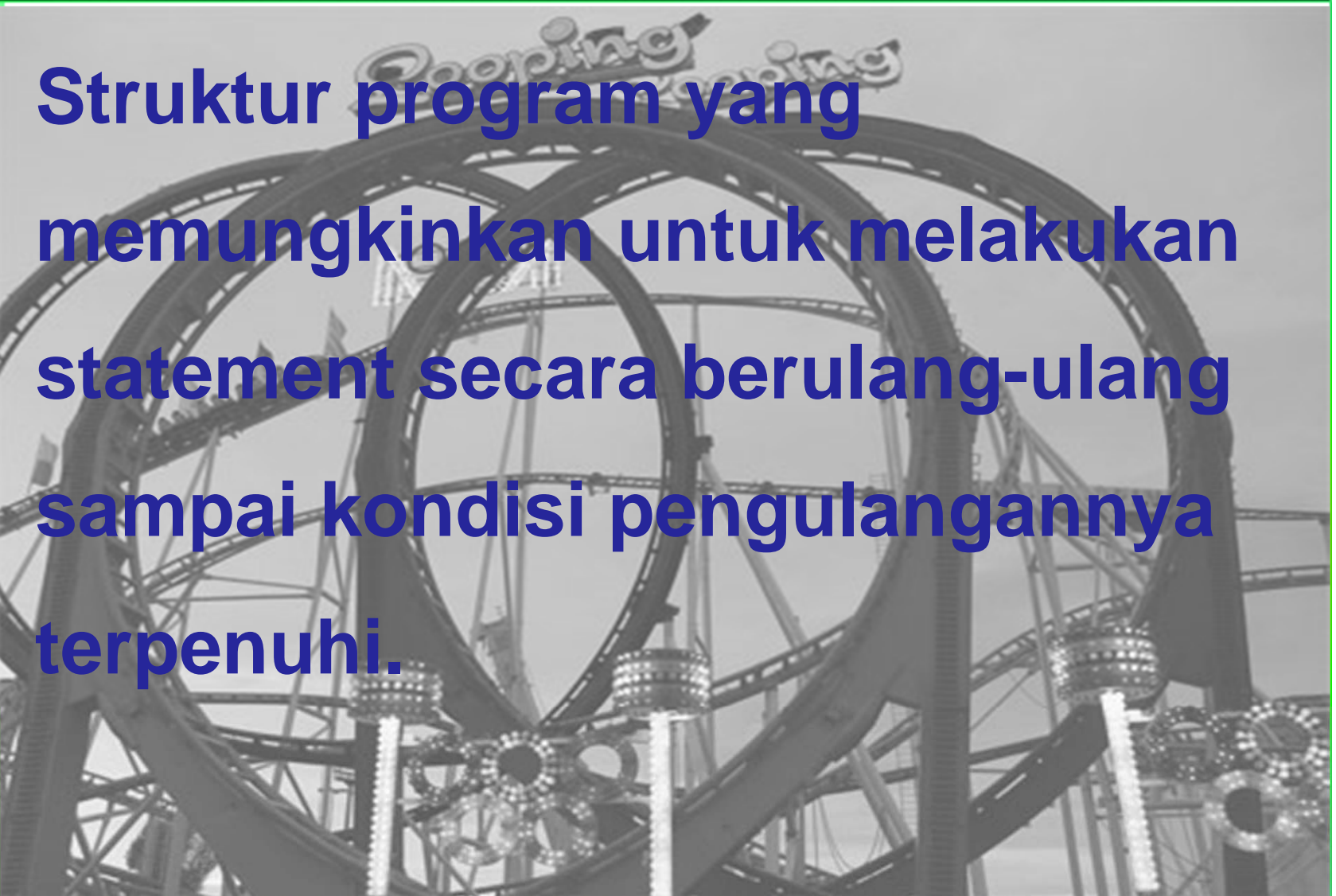
Kelebihan Komputer

1. Tidak bosan.
2. Penurut
3. Tidak cepat lelah.



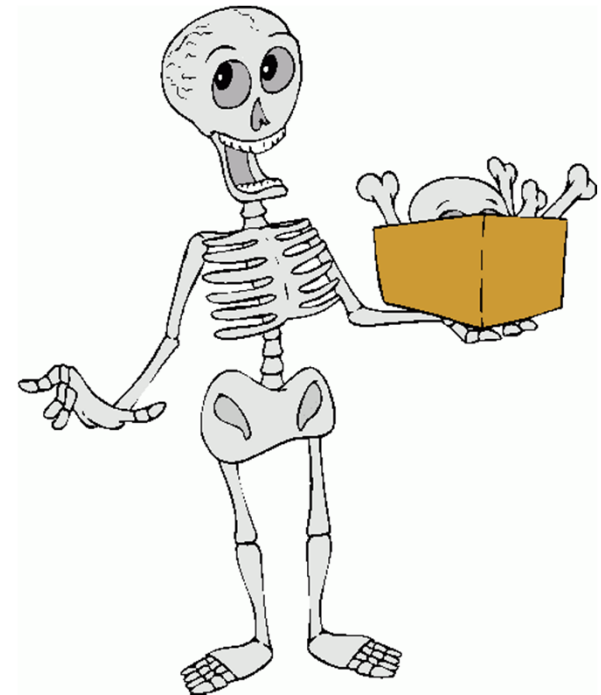
Struktur Pengulangan

Struktur program yang memungkinkan untuk melakukan statement secara berulang-ulang sampai kondisi pengulangannya terpenuhi.



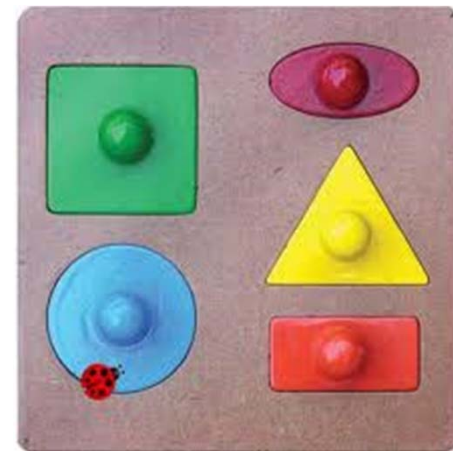
Bagian dalam Struktur Pengulangan

1. **Kondisi Pengulangan**
2. **Badan Pengulangan**
3. Inisialisasi
4. Terminasi



Bentuk Pengulangan

1. For
2. While
3. Do While
4. Label



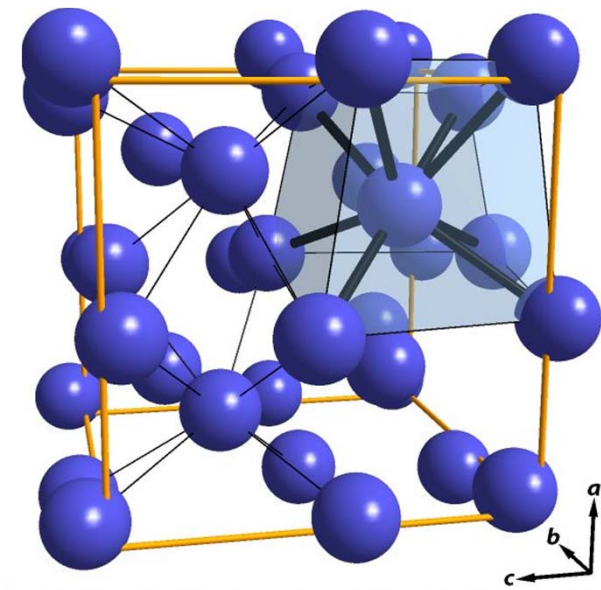
Pernyataan FOR

1. FOR digunakan untuk pengulangan yang **akhir pengulangannya** telah **dispesifikasikan**.
2. **Jumlah pengulangan** sudah **diketahui** di awal.
3. Bisa **ASCENDING** dan **DESCENDING**.



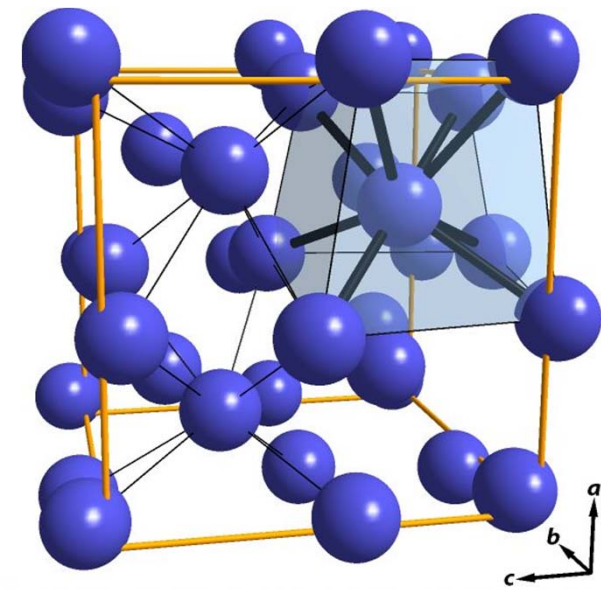
Format FOR

```
for(inisialisasi counter;kondisi pengulangan;statement)
{
    pernyataan;
}
```



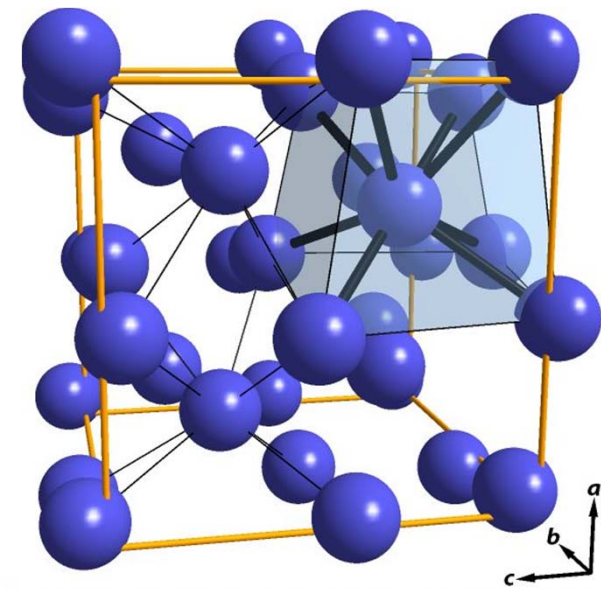
Format FOR Ascending

```
for(variabel=nilai_awal;kondisi;variabel++)  
{  
    pernyataan;  
}
```



Format FOR Descending

```
for(variabel=nilai_akhir;kondisi;variabel--)  
{  
    pernyataan;  
}
```



Contoh FOR C

```
1  /*
2    Program 6.2
3    Nama File   : Lat-6.2.c
4    Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     int i;
13     for(i=1;i<=5;i++)
14     {
15         printf("%i\n",i);
16     }
17     printf("\n");
18     for(i=5;i>=1;i--)
19     {
20         printf("%i\n",i);
21     }
22     printf("\n");
23     system("PAUSE");
24     return 0;
25 }
```

Contoh FOR C++

```
1  /*
2    Program 6.2
3    Nama File   : Lat-6.2.cpp
4    Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int main(int argc, char *argv[])
13 {
14     int i;
15     for(i=1;i<=5;i++)
16     {
17         cout<<i<<endl;
18     }
19     cout<<endl;
20     for(i=5;i>=1;i--)
21     {
22         cout<<i<<endl;
23     }
24     cout<<endl;
25     system("PAUSE");
26     return EXIT_SUCCESS;
27 }
```

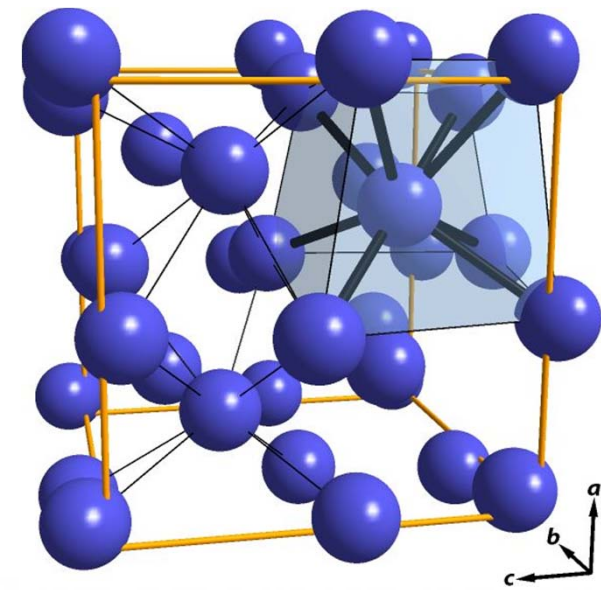
Pernyataan WHILE

1. While digunakan untuk melakukan pengulangan selama kondisi pengulangannya bernilai **true**.
2. Apabila kondisi pengulangan sudah bernilai **false** maka pengulangan **berhenti**.
3. WHILE melakukan **pemeriksaan kondisi di awal** pengulangan.



Format WHILE

```
while(kondisi)
{
    pernyataan;
}
```



Contoh WHILE C

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[])
5  {
6      int baris,kolom;
7      printf("Masukkan jumlah baris : ");scanf("%i",&baris);
8      while(baris>=1)
9      {
10         kolom=1;
11         while(kolom<=baris)
12         {
13             printf("%2i  ",baris*kolom);
14             kolom++;
15         }
16         printf("\n");
17         baris--;
18     }
19     printf("\n");
20     system("PAUSE");
21     return 0;
22 }
```

Contoh WHILE C++

```
1  #include <cstdlib>
2  #include <iostream>
3  #include <iomanip>
4
5  using namespace std;
6
7  int main(int argc, char *argv[])
8  {
9      int baris,kolom;
10     cout<<"Masukkan jumlah baris : ";cin>>baris;
11     while(baris>=1)
12     {
13         kolom=1;
14         while(kolom<=baris)
15         {
16             cout<<setw(2)<<baris*kolom<<" ";
17             kolom++;
18         }
19         cout<<endl;
20         baris--;
21     }
22     cout<<endl;
23     system("PAUSE");
24     return EXIT_SUCCESS;
25 }
```

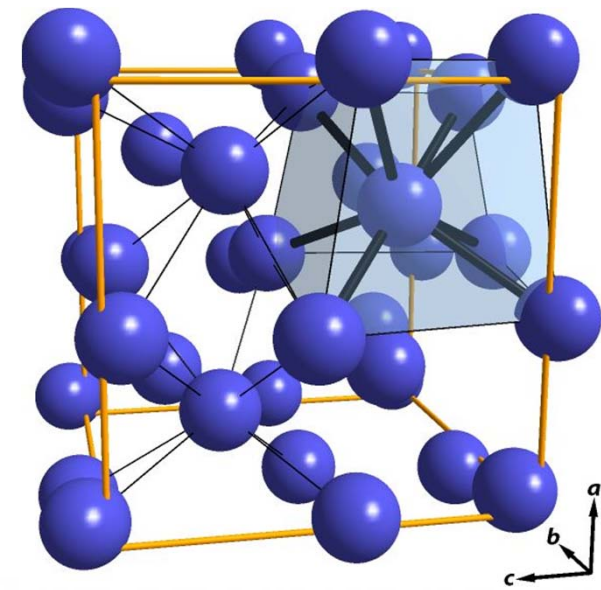
Pernyataan DO WHILE

1. Struktur DO WHILE memeriksa kondisi pengulangannya di **akhir pengulangan**.
2. Ekuivalen dengan perintah **REPEAT UNTIL NOT**.
3. **Minimal** menjalankan statement sebanyak **1 kali**.



Format WHILE

```
do  
{  
    pernyataan;  
}while(kondisi);
```



Contoh DO WHILE C

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <conio.h>
4
5  #define pass 1234
6  int main(int argc, char *argv[])
7  {
8      int password,i;
9      i=0;
10     do
11     {
12         printf("Masukkan password yang benar : ");
13         scanf("%i",&password);
14         if(password==pass)
15             printf("Password anda benar!\n");
16         else
17         {
18             printf("Password anda salah\n");getch();
19             system("cls");
20         }
21         i++;
22     }while((password!=pass)&&(i!=3));
23     system("PAUSE");
24     return 0;
25 }
```

Contoh DO WHILE C++

```
1  #include <cstdlib>
2  #include <iostream>
3  #include <conio.h>
4
5  #define pass 1234
6  using namespace std;
7
8  int main(int argc, char *argv[])
9  {
10     int password,i;
11     i=0;
12     do
13     {
14         cout<<"Masukkan password yang benar : ";cin>>password;
15         if(password==pass)
16             cout<<"Password anda benar!"<<endl;
17         else
18         {
19             cout<<"Password anda salah"<<endl;getch();
20             system("cls");
21         }
22         i++;
23     }while((password!=pass)&&(i!=3));
24     system("PAUSE");
25     return EXIT_SUCCESS;
26 }
```

Pernyataan LABEL

1. Struktur LABEL merupakan **struktur pengulangan yang pertama kali** digunakan.
2. Mempunyai banyak kelebihan karena **fleksibilitasnya**.
3. Dianjurkan **tidak digunakan**.




Label C (Baris Tidak Dieksekusi)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[])
5  {
6      int i;
7      i=0;
8      awal:
9          i=i+1;
10         printf("%i\n",i);
11         if(i<10)
12             goto awal;
13         else
14             goto akhir;
15         printf("Perintah ini tak akan dieksekusi\n");
16         printf("Perintah ini juga tak akan dieksekusi\n");
17
18     akhir:
19         printf("\n");
20         system("PAUSE");
21         return 0;
22 }
```

Label C++ (Baris Tidak Dieksekusi)

```
1  #include <cstdlib>
2  #include <iostream>
3
4  using namespace std;
5
6  int main(int argc, char *argv[])
7  {
8      int i;
9      i=0;
10     awal:
11         i=i+1;
12         cout<<i<<endl;
13         if(i<10)
14             goto awal;
15         else
16             goto akhir;
17         cout<<"Perintah ini tak akan dieksekusi"<<endl;
18         cout<<"Perintah ini juga tak akan dieksekusi"<<endl;
19
20     akhir:
21         cout<<endl;
22         system("PAUSE");
23         return EXIT_SUCCESS;
24 }
```

Label C (Logik Program Rusak)



```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[])
5  {
6      int i;
7      for(i=1;i<=5;i++)
8      {
9          goto free_pass;
10         if(i%2==0)
11             printf("%i bilangan genap!\n\n",i);
12         else
13         {
14             free_pass:
15                 printf("%i bilangan ganjil!\n\n",i);
16         }
17     }
18     system("PAUSE");
19     return 0;
20 }
```

Label C++ (Logik Program Rusak)

```
1  #include <cstdlib>
2  #include <iostream>
3
4  using namespace std;
5
6  int main(int argc, char *argv[])
7  {
8      int i;
9      for(i=1;i<=5;i++)
10     {
11         goto free_pass;
12         if(i%2==0)
13             cout<<i<<" bilangan genap!"<<endl<<endl;
14         else
15         {
16             free_pass:
17                 cout<<i<<" bilangan ganjil!"<<endl<<endl;
18         }
19     }
20     system("PAUSE");
21     return EXIT_SUCCESS;
22 }
```

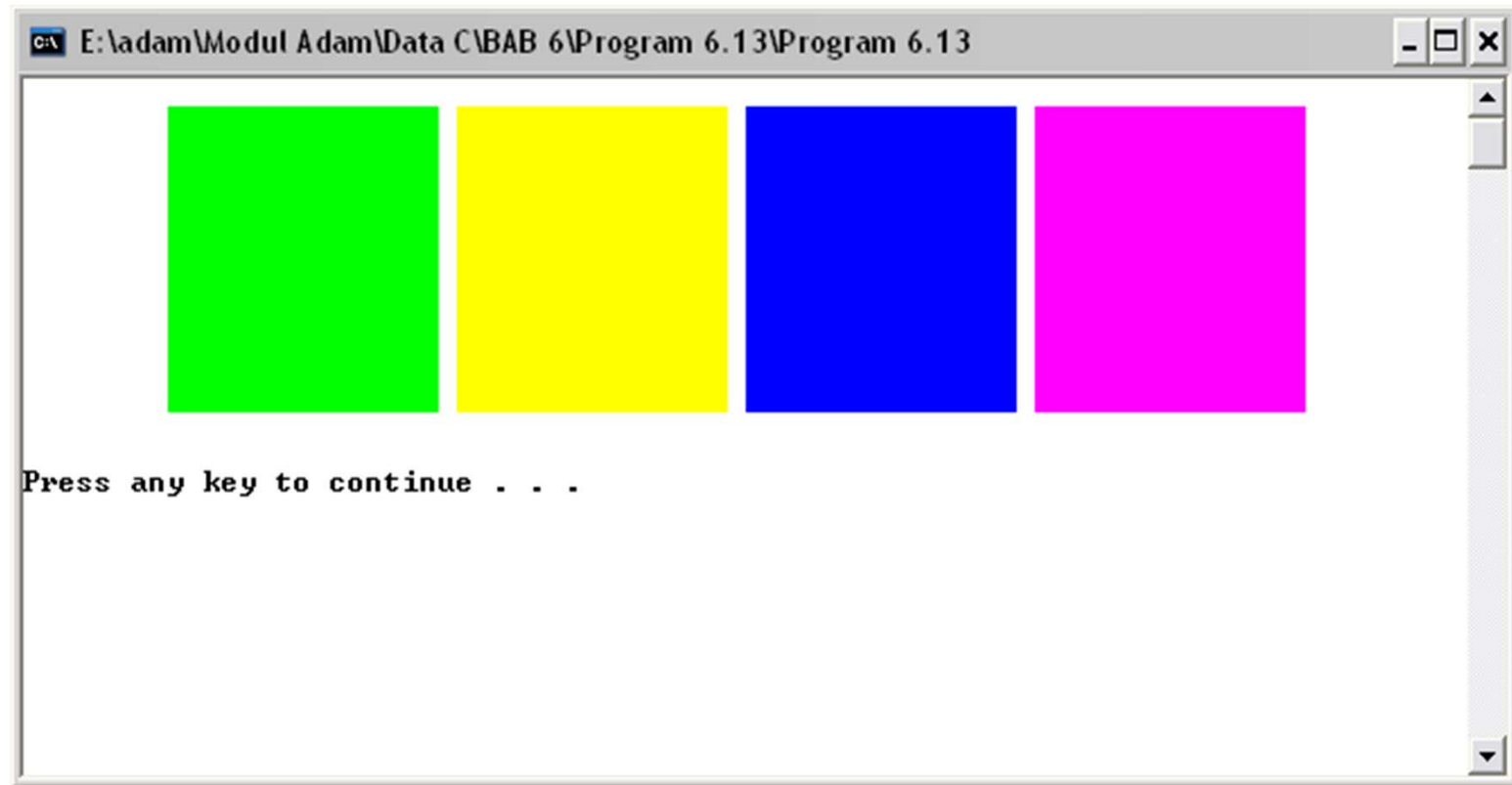
Break VS Continue

Break digunakan untuk menghentikan eksekusi pengulangan dan melanjutkannya ke blok berikutnya.

Continue digunakan untuk meneruskan pengulangan berikutnya tanpa melanjutkan sisa statementnya.



Kombinasi Pengulangan dengan Conio2



Kasus

1. Buatlah program C/C++ untuk menyelesaikan kasus seperti di bawah ini (warna bintang setiap baris dibedakan):

N=5

*

* *

* * *

* * * *

* * * * *

Kasus

2. Buatlah program C++ untuk menyelesaikan kasus seperti di bawah ini (warna bintang setiap baris dibedakan):

N=3

*


* *

* * *

* *

*

Kasus

- 
3. Modifikasi project menu pada struktur pemilihan agar menunya bisa terus ditampilkan sampai user menginput pilihan keluar.
 4. Buat algoritma dan program untuk menghitung:
 - a. $s = 1 - 2/3 + 3/5 - 4/7$
 - b. Jumlah deret bilangan prima sampai suku ke-n.
 6. Buat Algoritma dan program untuk menghitung nilai terbesar dan nilai rata-rata dari 10 orang mahasiswa.

