



PERULANGAN

Pertemuan ke-5

U N I V E R S I T A S B U N D A M U L I A

Sub-CPMK

- *Mahasiswa dapat menggunakan perulangan dalam konsep Pemrograman Berorientasi Objek (PBO). (C3, A3).*

Materi

1. Jenis-Jenis Perulangan
2. Jenis-Jenis Controlled Jump
3. Nested Loop
4. Rekursif



1. Jenis-Jenis Perulangan

- Cara kerjanya perulangan dibedakan menjadi dua yaitu: ***counted loop*** (jumlah perulangannya sudah pasti) dan ***uncounted loop*** (jumlah perulangannya tidak tentu).
- Jenis perulangan pada Java adalah: **for, while, do...while.**
- Perulangan for merupakan jenis perulangan *counted loop*, dan while, do...while merupakan jenis perulangan *uncounted loop*.

1.1 Perulangan For

- Perulangan **for** digunakan untuk mengulang sejumlah nilai yang sudah pasti. Sintaks untuk perulangan for seperti berikut:

Sintaks: for()

```
for (nilai awal; banyak pengulangan; penambahan nilai)
```

Contoh:

```
for (i=0; i<=10; i++)  
{  
    Blok statement yang diulang;  
}
```

1.1 Perulangan For (Lanj..)

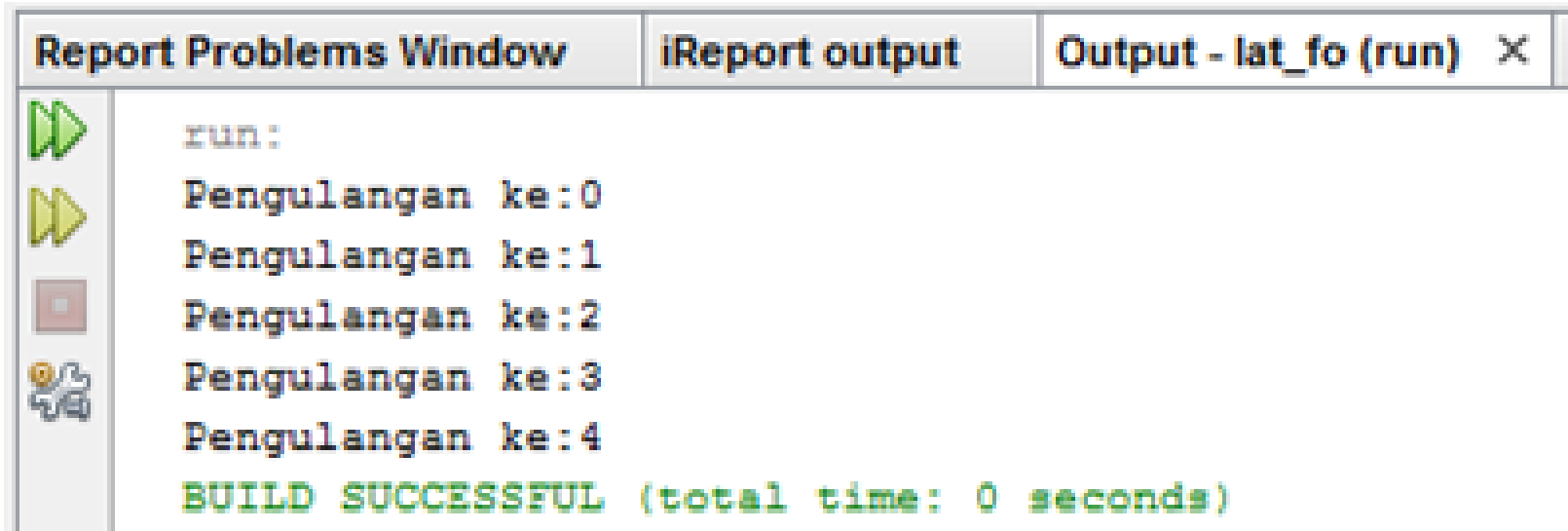
- Contoh perulangan for pada bahasa Java.

Contoh: perulangan for()

```
1.  public class LatihanFor
2.  {
3.      public static void main (String[] args)
4.      {
5.          int i;
6.          for (i = 0; i < 5; i++)
7.          {
8.              System.out.println( "Pengulangan ke:"+i);
9.          }
10.     }
11. }
```

1.1 Perulangan For (Lanj..)

- Contoh keluaran program.



```
run:
Pengulangan ke:0
Pengulangan ke:1
Pengulangan ke:2
Pengulangan ke:3
Pengulangan ke:4
BUILD SUCCESSFUL (total time: 0 seconds)
```

1.2 Perulangan While

- Perulangan yang nilai perulangannya belum diketahui, maka dapat menggunakan perintah while.
- Sintaks perulangan while seperti berikut:

Sintaks: while()

```
while (kondisi)
{
    Statement yang akan diulang;
    Pertambahan_nilai;
}
```


1.2 Perulangan While (Lanj..)

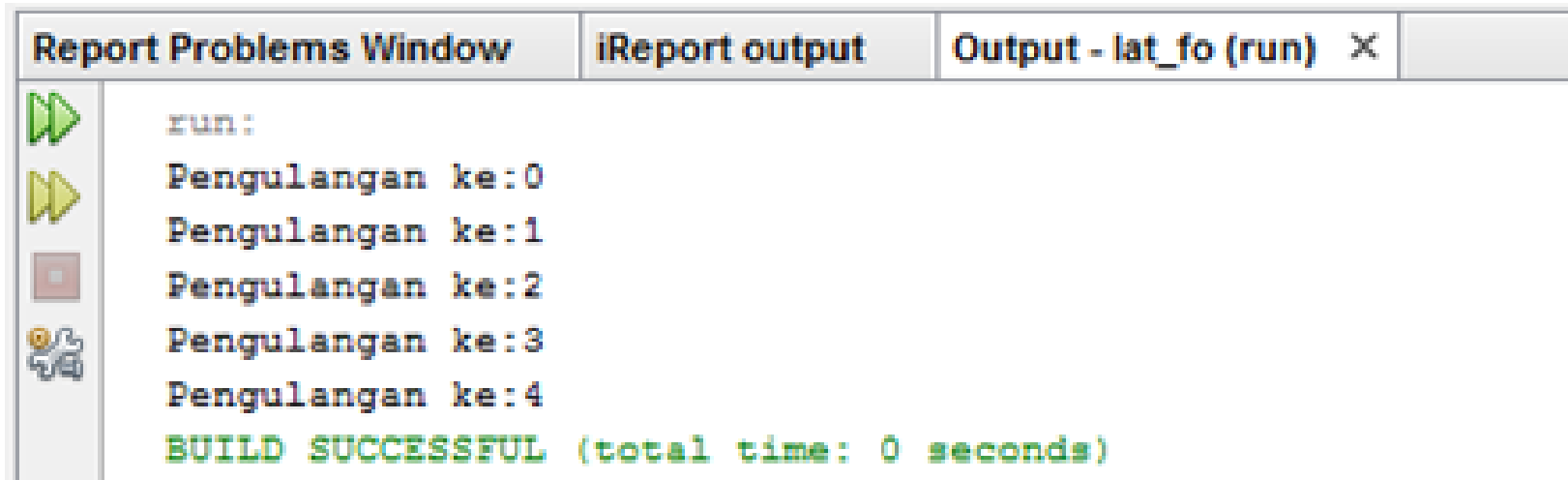
- Contoh perulangan while pada Java

Contoh: while()

```
1.  public class PerulanganWhile
2.  {
3.      public static void main (String[] args)
4.      {
5.          int i = 0;
6.          while (i < 5)
7.          {
8.              System.out.println( "Pengulangan ke:"+i);
9.              i = i+1;
10.         }
11.     }
12. }
```

1.2 Perulangan While (Lanj..)

- Contoh keluaran program.



The screenshot shows an IDE window with three tabs: "Report Problems Window", "iReport output", and "Output - lat_fo (run)". The "Output - lat_fo (run)" tab is active, displaying the following text:

```
run:  
Pengulangan ke:0  
Pengulangan ke:1  
Pengulangan ke:2  
Pengulangan ke:3  
Pengulangan ke:4  
BUILD SUCCESSFUL (total time: 0 seconds)
```

1.3 Perulangan Do..While

- Perulangan yang dilakukan tidak diketahui kapan berakhirnya, perulangan akan berhenti apabila sudah memenuhi syarat untuk berhenti. Sintaks-nya seperti berikut:

Sintaks: do...while()

```
do
{
    Statement yang diulang;
}
while (kondisi);
```

1.3 Perulangan Do..While (Lanj..)

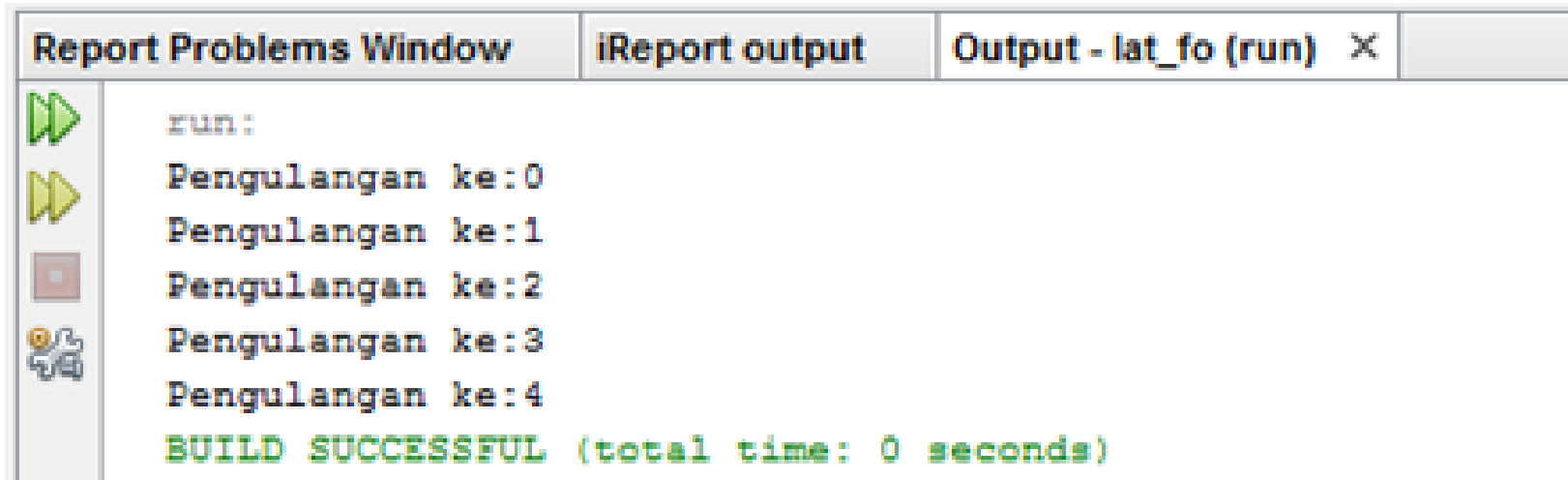
- Contoh program Do..While pada Java.

Contoh: do...while()

```
1.  public class PerulanganDoWhile
2.  {
3.      public static void main (String[] args)
4.      {
5.          int i = 0;
6.          do
7.          {
8.              System.out.println( "Pengulangan ke:"+i);
9.              i = i+1;
10.         }
11.         while (i<5);
12.     }
13. }
```

1.3 Perulangan Do..While (Lanj..)

- Contoh keluaran program.



The screenshot shows a window titled "Output - lat_fo (run)" with a close button. On the left, there is a vertical toolbar with icons for running (green play button), stepping through (yellow play button), stopping (red square), and debugging (wrench and gear). The output text is as follows:

```
run:  
Pengulangan ke:0  
Pengulangan ke:1  
Pengulangan ke:2  
Pengulangan ke:3  
Pengulangan ke:4  
BUILD SUCCESSFUL (total time: 0 seconds)
```



2. Jenis-jenis Controlled Jump

- *Controlled jump* mengubah eksekusi langsung ke pernyataan tertentu.
- Jenis-jenis *controlled jump* pada bahasa java seperti:
 - break
 - continue
 - return

2.1 Break

- Controlled jump seperti break, continue, dan return mengubah eksekusi langsung ke pernyataan tertentu.
- Berikut contoh penerapan break pada program.

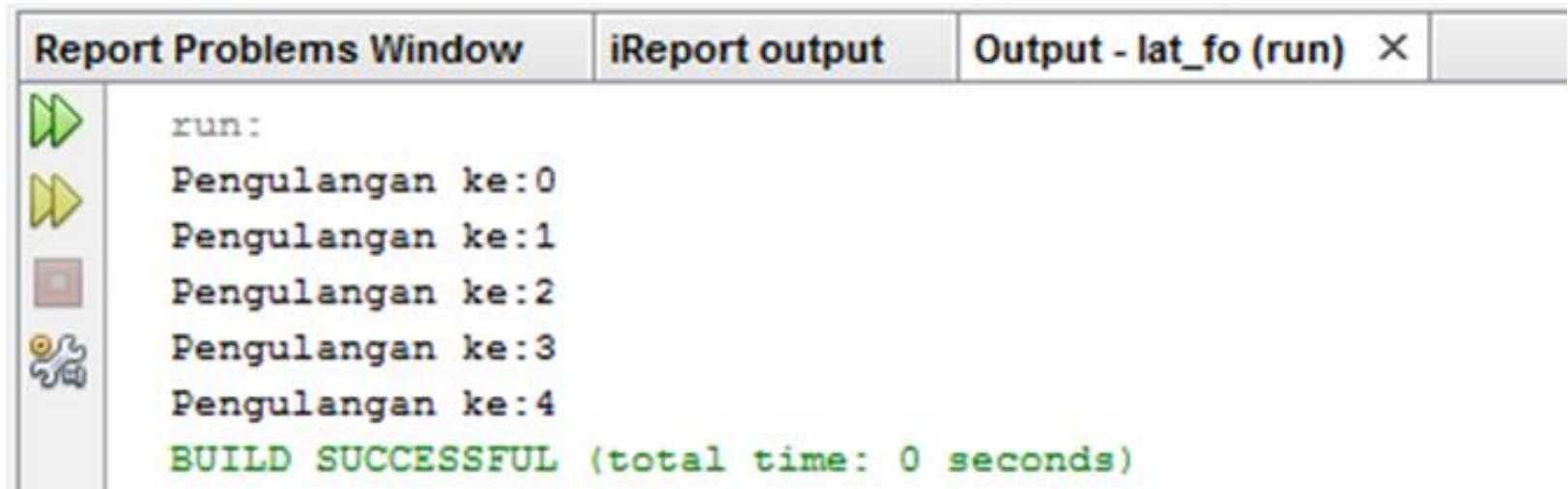
2.1 Break (Lanj..)

Contoh: break

```
1.  public class PernyataanBreak
2.  {
3.      public static void main (String[] args)
4.      {
5.          int i;
6.          for (i = 0; i < 10; i++)
7.          {
8.              System.out.println( "Pengulangan ke:"+i);
9.              if (i==5)
10.                 break;
11.          }
12.      }
13. }
```

2.1 Break (Lanj..)

- Seharusnya dilakukan pengulangan sebanyak 10 kali, namun ada statement jika $i==5$ maka break, sehingga program langsung berhenti pada $i==5$. Contoh keluaran program.



The screenshot shows a window titled 'Report Problems Window' with a sub-tab 'iReport output'. The output text is as follows:

```
run:
Pengulangan ke:0
Pengulangan ke:1
Pengulangan ke:2
Pengulangan ke:3
Pengulangan ke:4
BUILD SUCCESSFUL (total time: 0 seconds)
```

2.2 Continue

- Penggunaan *continue* digunakan untuk melewati semua pernyataan yang tidak dieksekusi dalam perulangan, dan langsung melanjutkan perulangan.
- Berikut contoh program dengan menggunakan *continue*.

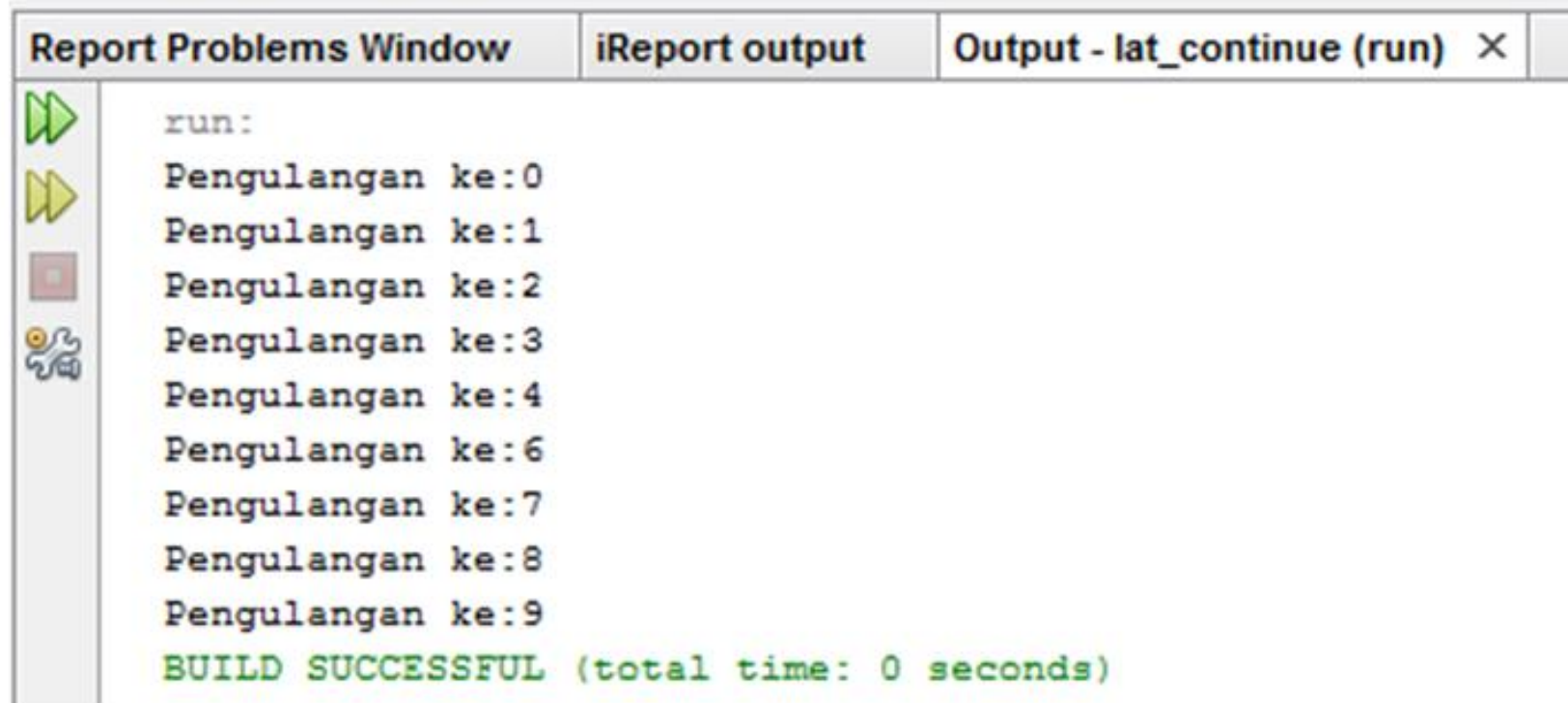
2.2 Continue (Lanj..)

Contoh: continue

```
1.  public class LatihanContinue
2.  {
3.      public static void main (String[] args)
4.      {
5.          int i;
6.          for (i = 0; i < 10; i++)
7.          {
8.              if (i==5)
9.                  continue;
10.             System.out.println( "Pengulangan ke:"+i);
11.         }
12.     }
13. }
```

2.2 Continue (Lanj..)

- Contoh keluaran program.



The screenshot shows a window titled "Output - lat_continue (run)" with a close button (X). The window contains the following text:

```
run:  
Pengulangan ke:0  
Pengulangan ke:1  
Pengulangan ke:2  
Pengulangan ke:3  
Pengulangan ke:4  
Pengulangan ke:6  
Pengulangan ke:7  
Pengulangan ke:8  
Pengulangan ke:9  
BUILD SUCCESSFUL (total time: 0 seconds)
```

On the left side of the window, there is a vertical toolbar with icons for running (green play button), stepping through (yellow play button), stopping (red square), and a search icon (magnifying glass).

2.3 Return

- Untuk segera keluar dari fungsi perulangan dapat dilakukan dengan perintah *return*.
- Contoh penerapan pada program seperti berikut ini.

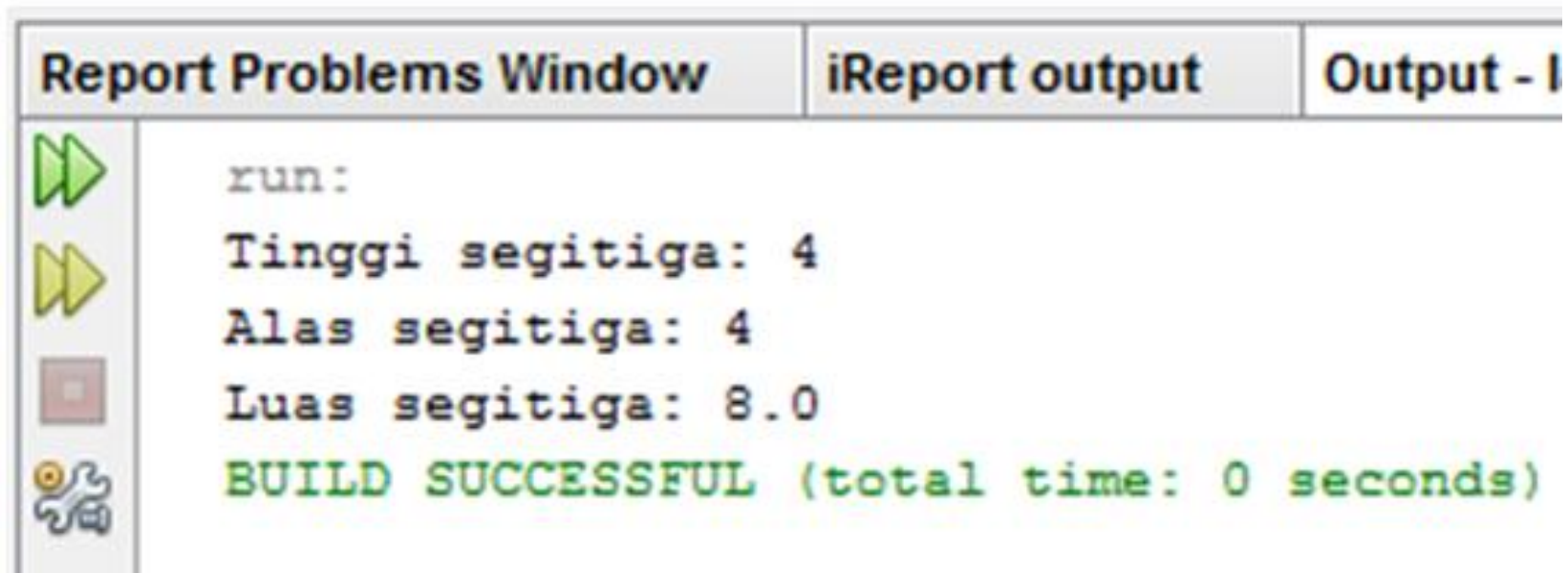
2.3 Return (Lanj..)

Contoh: return

```
1. package latreturn;
2. public class Latreturn {
3.
4.     static double LuasSegitiga(int alas, int tinggi)
5.     {
6.         double luas = 0.5 * alas * tinggi;
7.         return luas;
8.     }
9.     public static void main(String[] args)
10.    {
11.        // TODO code application logic here
12.        int Alas = 4;
13.        int Tinggi = 4;
14.        System.out.println("Alas segitiga: "+ Alas);
15.        System.out.println("Tinggi segitiga: "+ Tinggi);
16.        System.out.println("Luas segitiga: "+
17.            LuasSegitiga(Alas,Tinggi));
18.    }
```

2.3 Return (Lanj..)

- Contoh keluaran program seperti berikut.



The screenshot shows a window titled 'Report Problems Window' with a tab labeled 'iReport output'. The output text is as follows:

```
run:  
Tinggi segitiga: 4  
Alas segitiga: 4  
Luas segitiga: 8.0  
BUILD SUCCESSFUL (total time: 0 seconds)
```




3. Nested Loop

- Perulangan di dalam perulangan disebut dengan perulangan bersarang (*nested loop*).
- Pada *nested loop* maka *loop* paling dalam dikerjakan terlebih dahulu, bila *loop* dalam sudah selesai dikerjakan baru mengerjakan *loop* dibagian luar.
- Berikut contoh nested loop (perulangan dalam perulangan).

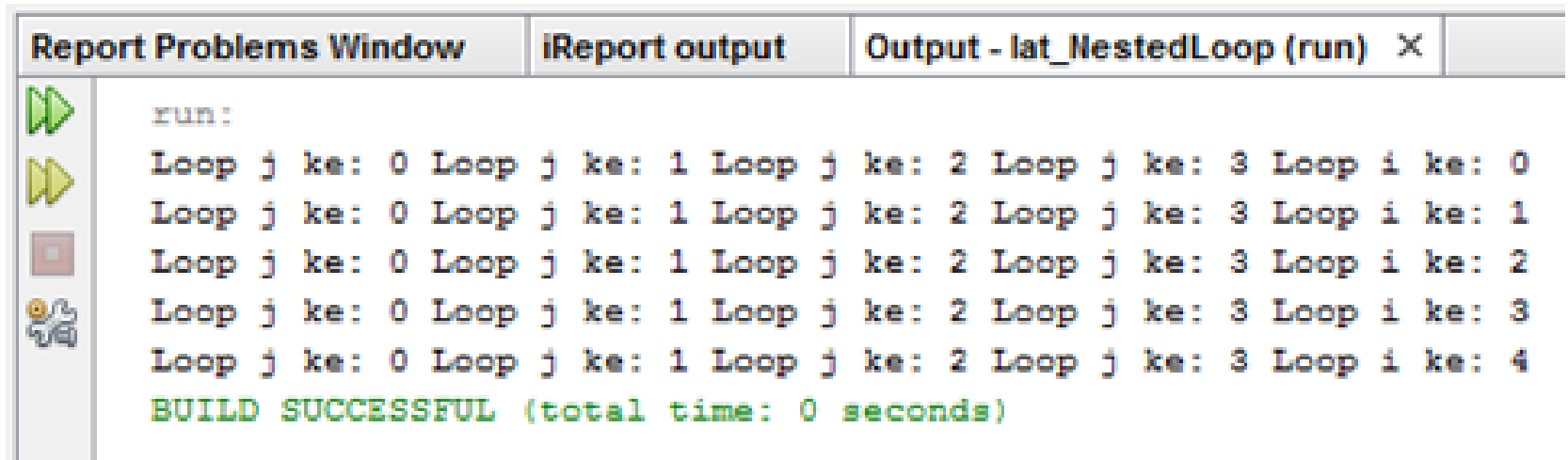
Lanj...

Contoh: nested loop

```
1.  package lat_nestedloop;
2.
3.  public class Lat_NestedLoop {
4.
5.      public static void main(String[] args)
6.      {
7.          // TODO code application logic here
8.          for (int i=0; i<5; i++)
9.          {
10.             for (int j=0; j<4; j++ )
11.             {
12.                 System.out.print("Loop j ke: "+j+" ");
13.             }
14.             System.out.println("Loop i ke: "+i+" ");
15.         }
16.     }
17. }
```

Lanj...

- Keluaran program seperti berikut.



```
run:  
Loop j ke: 0 Loop j ke: 1 Loop j ke: 2 Loop j ke: 3 Loop i ke: 0  
Loop j ke: 0 Loop j ke: 1 Loop j ke: 2 Loop j ke: 3 Loop i ke: 1  
Loop j ke: 0 Loop j ke: 1 Loop j ke: 2 Loop j ke: 3 Loop i ke: 2  
Loop j ke: 0 Loop j ke: 1 Loop j ke: 2 Loop j ke: 3 Loop i ke: 3  
Loop j ke: 0 Loop j ke: 1 Loop j ke: 2 Loop j ke: 3 Loop i ke: 4  
BUILD SUCCESSFUL (total time: 0 seconds)
```



4. Rekursif

- Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri.
- Ada kalanya proses yang berulang kali (loop) dapat juga diselesaikan dengan proses rekursif. Hati-hati dalam penerapan proses rekursif, bila tidak ada kapan berhentinya maka akan terjadi rekursif tanpa batas.
- Penggunaan loop biasa dapat dikatakan sebagai proses iteratif (iterasi).

Lanj...

- Untuk mencari nilai faktorial dapat dilakukan secara iteratif dan rekursif. Faktorial dari bilangan bulat tak negatif n didefinisikan sebagai berikut:

$$A = \begin{cases} 1 & , n = 0 \\ 1 \times 2 \times \dots \times (n - 1) \times n & , n > 0 \end{cases}$$

- Sebagai contoh:

- $0! = 1$
- $1! = 1$
- $2! = 1 \times 2$
- $3! = 1 \times 2 \times 3$

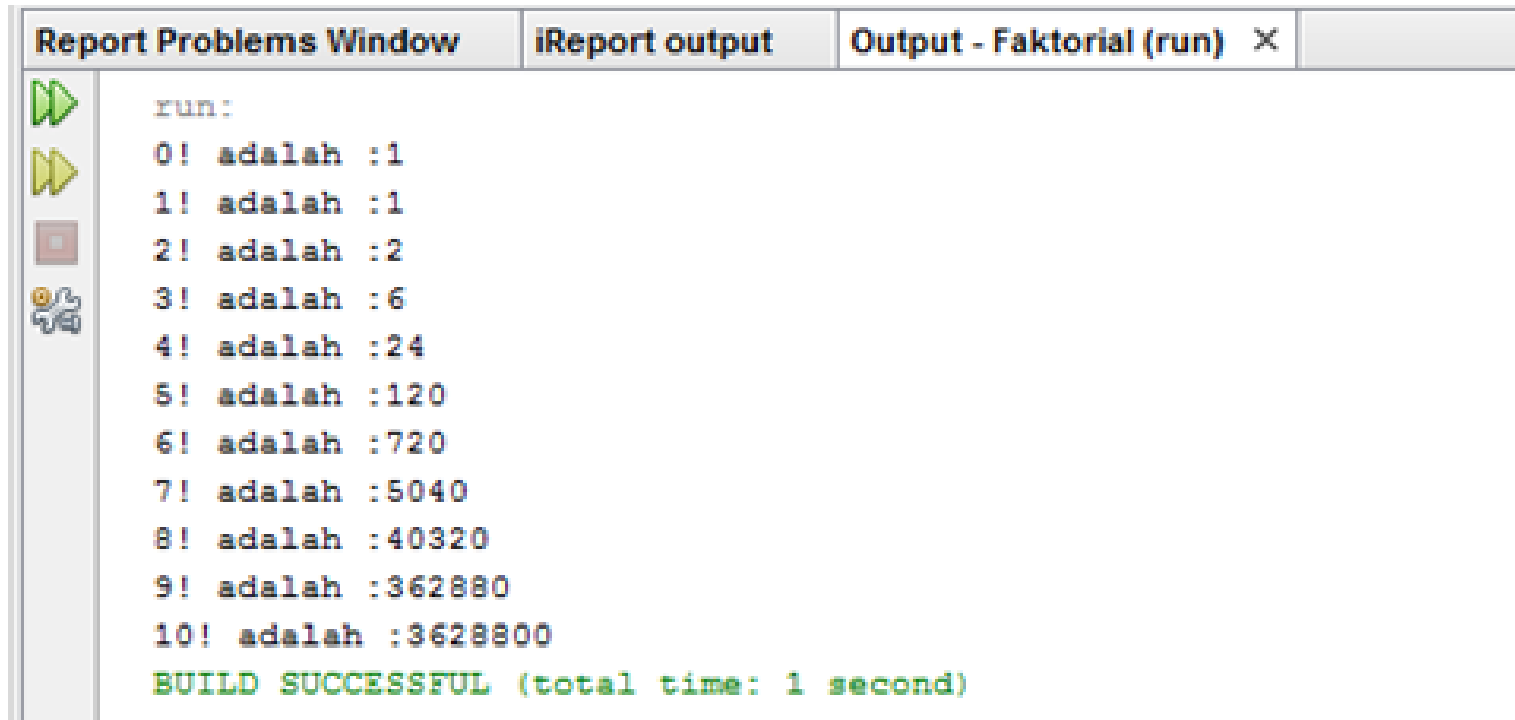
Lanj...

Contoh: program faktorial secara iteratif

```
1.  public class Faktorial
2.  {
3.      public static void main (String[] args)
4.      {
5.          long batas = 10;
6.          long mfaktorial = 1;
7.
8.          for (int i = 0; i <= batas; i++)
9.          {
10.             mfaktorial = 1;
11.             for (int faktor = 2; faktor <= i; faktor
12. ++))
13.                 mfaktorial *= faktor;
14.             System.out.println(i + "!" + " adalah :
15. " + mfaktorial);
16.         }
17.     }
```


Lanj...

- Contoh keluaran program seperti berikut.



```
run:
0! adalah :1
1! adalah :1
2! adalah :2
3! adalah :6
4! adalah :24
5! adalah :120
6! adalah :720
7! adalah :5040
8! adalah :40320
9! adalah :362880
10! adalah :3628800
BUILD SUCCESSFUL (total time: 1 second)
```

Lanj...

Function Faktorial (n: integer) integer

DEKLARASI:

i, F: integer;

ALGORITMA:

F = 1

i = 1

while i ≤ n do

 F = F * i

 i = i + 1

end while

Lanj...

- Berdasarkan algoritma faktorial buatlah program faktorial dengan menggunakan pendekatan secara rekursif yang diimplementasikan dengan konsep pemrograman berorientasi objek.

Lanj...

- Buatlah program untuk mencetak deret bilangan fibonanci.
- Deret bilangan fibonanci adalah sebuah deret bilangan dengan syarat harus diketahui dua buah bilangan sebelumnya, kemudian ditambahkan.
- Contoh: 0,1,1,2,3,5,8,13,21,dan seterusnya.

Lanj...

Contoh: deret fibonacci dengan rekursif



```
1. package lat_fibonanci;
2. import java.util.Scanner;
3.
4. public class Lat_fibonanci {
5.
6.     public static int Fibbo(int n)
7.     {
8.         int c;
9.         if (n==0)
10.            return 0;
11.         if (n==1)
12.            return(1);
13.         else
14.            c = Fibbo(n-1) + Fibbo(n-2);
15.         return c;
16.     }
```

Lanj...

```
17.     public static void main(String[] args) {
18.         // TODO code application logic
19.         int n, t,i;
20.
21.         Scanner input = new Scanner(System.in);
22.         System.out.print("Batas deret fibonacci : ");
23.         n  = input.nextInt();
24.         for (i = 0; i <=n; i++)
25.         {
26.             t = Fibbo(i);
27.             System.out.print(t+" ");
28.         }
29.     }
30. }
```

Lanj...

- Contoh keluaran program.

Report Problems Window	iReport output	Output - lat_
	<code>run:</code>	
	<code>Batas deret fibbonanci : 10</code>	
	<code>0 1 1 2 3 5 8 13 21 34 55</code>	<code>BUILD SUCCESSFUL</code>

Ringkasan:

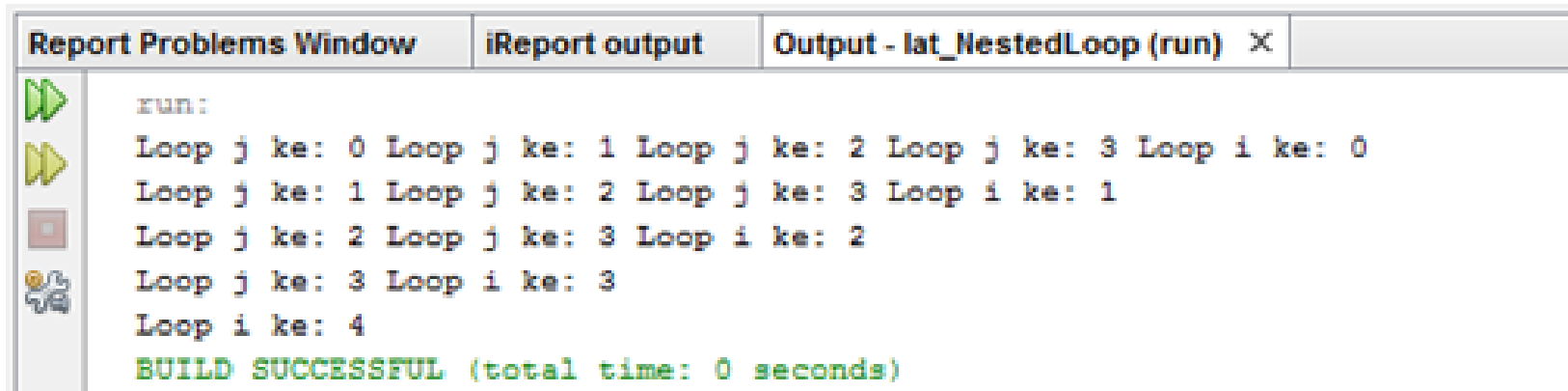
- Cara kerja perulangan dapat dibedakan menjadi dua yaitu: *counted loop* dan *uncounted loop*.
- Jenis pengulangan yang dikenal pada java yaitu: for, while, do...while.
- Perulangan dalam perulangan dikenal juga dengan *nested loop*.
- Prose rekursif merupakan proses memanggil dirinya sendiri (bisa prosedur / fungsi).

Latihan Mandiri:

- Jelaskan apa yang dimaksud dengan *counted loop* dan *uncounted loop*.
- Jelaskan secara singkat cara kerja masing-masing perulangan `for()`, `while()`, `do...while()`.
- Jelaskan secara singkat apa yang dimaksud *nested loop*.
- Jelaskan cara kerja rekursif dan iterasi, apa perbedaan kedua proses tersebut.

Latihan Mandiri (Lanj...)

- Buatlah sebuah program dengan konsep berorientasi objek dengan keluaran program seperti pada gambar.



```
run:  
Loop j ke: 0 Loop j ke: 1 Loop j ke: 2 Loop j ke: 3 Loop i ke: 0  
Loop j ke: 1 Loop j ke: 2 Loop j ke: 3 Loop i ke: 1  
Loop j ke: 2 Loop j ke: 3 Loop i ke: 2  
Loop j ke: 3 Loop i ke: 3  
Loop i ke: 4  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Latihan Mandiri (Lanj...)

- Buatlah sebuah program untuk mengkonversi suhu dari Celcius ke Fahrenheit, nilai batas Celcius dimasukkan dari keyboard. Program akan menanyakan berapa batas Celcius, bila dimasukkan 10 maka program akan membuat tabel konversi dari 0 s/d 10 beserta nilai konversi ke Fahrenheit. Buat programnya dengan konsep berorientasi objek. **Celcius ke Fahrenheit = $(5.0/9) * (Celcius - 32)$.**

Latihan Mandiri (Lanj...)

- Buatlah sebuah program untuk membuat daftar bilangan prima. Program akan menanyakan ke pengguna untuk memasukkan nilai batas bilangan prima, misalkan 100 maka program akan memeriksa angka yang merupakan bilangan prima dari 0 s/d 100. Yang merupakan bilangan prima dari 0 s/d 100 akan ditampilkan. Buat program dengan konsep berorientasi objek.

Latihan Mandiri (Lanj...)

- Buatlah sebuah program (bebas) dengan konsep berorientasi objek, dengan menerapkan perulangan.
- Buatlah program deret fibonacci dengan konsep berorientasi objek secara rekursif.
- Buatlah sebuah program (bebas) dengan konsep berorientasi objek, dengan menerapkan proses rekursif.



TERIMA KASIH

U N I V E R S I T A S B U N D A M U L I A