

# DateTime

Java menyediakan beberapa kelas untuk menangani DateTime. Kelas-kelas ini menangani segala yang berhubungan dengan tanggal dan waktu. Pada materi ini kita akan belajar bagaimana menggunakan, menentukan, merubah serta menampilkan dengan format waktu belahan dunia yang diinginkan.

Berikut kelas yang sering digunakan untuk mendukung pengelolaan DateTime

Kelas	Deskripsi
System.currentTimeMillis()	Method yang mengembalikan waktu sekarang dalam millisecond
java.util.Date	Kelas yang mewakili tanggal dan waktu.
java.util.Calendar	Kelas yang memiliki spesifik method untuk mengkonversi waktu dan tanggal
java.text.SimpleDateFormat	Kelas yang membantu menterjemahkan format waktu dalam bentuk String ke format date time serta sebaliknya

## CurrentTimeMillis

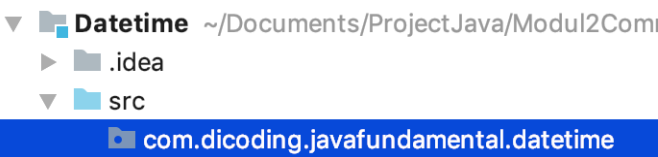
Merupakan method bagian dari kelas System yang menampilkan waktu sekarang dalam satuan milisecond (1000 milisecond = 1 second). CurrentTimeMillis() mengembalikan jumlah waktu mulai dari 1-1-1970 sampai dengan waktu yang sekarang. `System.currentTimeMillis()` biasanya digunakan untuk mengukur suatu waktu yang dibutuhkan dengan lebih spesifik. Ia biasa juga dipakai untuk tambahan sebagai variabel unik.

## Codelab CurrentTimeMillis

Mari kita coba penerapannya di kode berikut:

- 1. Buatlah proyek baru dengan nama Datetime dengan nama package

`com.dicoding.javafundamental.datetime` di dalamnya:



- 2. Buatlah sebuah kelas baru di dalamnya dengan nama `ExampleSystemCurrentMilis`, kemudian tambahkan kode berikut:

```
1. package com.dicoding.javafundamental.datetime;
2.
3. public class ExampleSystemCurrentMillis {
4.     public static void main(String[] args) {
5.         long timeNow = System.currentTimeMillis();
6.
7.         System.out.println("Waktu sekarang adalah " + timeNow + " milisecond");
8.     }
9. }
```

3. Melalui perintah `System.currentTimeMillis();`, saat kode diatas dieksekusi, maka waktu yang didapat saat itu disimpan pada variable timeNow yang bertipe data long. Data yang disimpan pada timeNow dapat diolah untuk hal lain atau ditampilkan seperti pada hasil di bawah ini:

Waktu sekarang adalah 1563792106146 milisecond

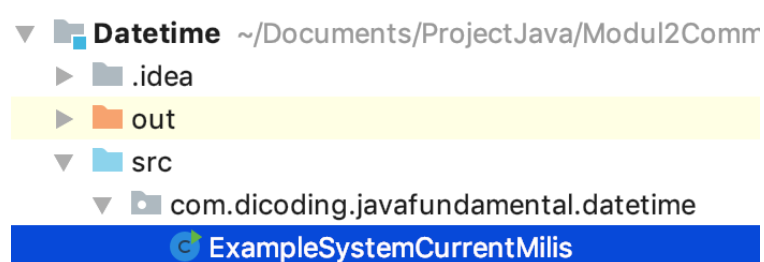
## Date

Merupakan kelas tertua yang menangani Waktu dan Tanggal. Dalam penggunaannya sekarang sudah banyak method yang sudah *deprecated* karena adanya java.util.Calendar. Kelas Calendar memiliki kemampuan penanganan waktu dan tanggal yang lebih baik. Tetapi jangan kuatir, masih ada beberapa kebutuhan kode yang bisa diselesaikan dengan menggunakan Date.

## Codelab Date

Mari kita praktikan codelab berikut:

1. Bukalah kembali proyek Datetime.



2. Buatlah kelas baru dengan nama ExampleDate dan masukkan kode berikut ke dalam kelas tersebut:

```
1. package com.dicoding.javafundamental.datetime;
2.
3. import java.util.Date;
4.
5. public class ExampleDate {
6.     public static void main(String[] args) {
7.         Date date = new Date();
8.         System.out.println("Tanggal sekarang adalah " + date.toString());
9.     }
10. }
```

Untuk menghasilkan tanggal dan waktu sekarang `java.util.Date` kita perlu lakukan inisialisasi lalu panggil dengan method `date.toString()` untuk menampilkan hasil tanggal sekarang setelah dijalankan programnya.

3. Jalankan tersebut maka hasilnya akan jadi seperti ini:

```
Tanggal sekarang adalah Mon Jul 22 17:45:41 WIB 2019
```

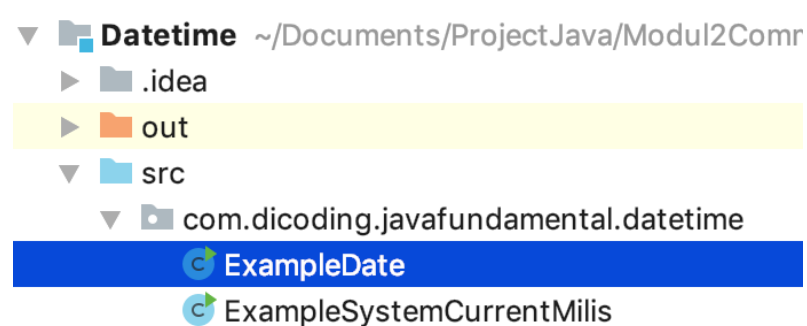
## Calendar

Kelas dari grup bagian dari `Java.util` yang menggantikan fungsi `date` untuk mengkonversi data tanggal dan waktu secara instan. Kelas `Calendar` ini memiliki banyak fungsi untuk parsing dan manipulasi tanggal. Berikut hal penting yang perlu diperhatikan tentang mengapa kita menggunakan kelas `Calendar` ini:

1. Kelas ini menyediakan ruang dan method untuk mengimplementasikan sistem kalender yang berada di luar dari bagian `java.util.Calendar`
2. Kelas ini juga dapat menentukan dan mengukur waktu sesuai yang diinginkan

## Codelab Calendar Menampilkan Waktu Sekarang

1. Bukalah kembali proyek `Datetime`.



2. Buatlah kelas baru dengan nama `ExampleCalendar` dan masukkan kode berikut ke dalam kelas tersebut:

```
1. package com.dicoding.javafundamental.datetime;
2.
3. import java.util.Calendar;
4.
5. public class ExampleCalendar {
6.     public static void main(String[] args) {
7.
8.         // Menampilkan waktu sekarang
9.         Calendar calendar = Calendar.getInstance();
10.        System.out.println("Waktu sekarang adalah " + calendar.getTime());
11.    }
12. }
```

3. Jalankan tersebut maka hasilnya akan jadi seperti ini:

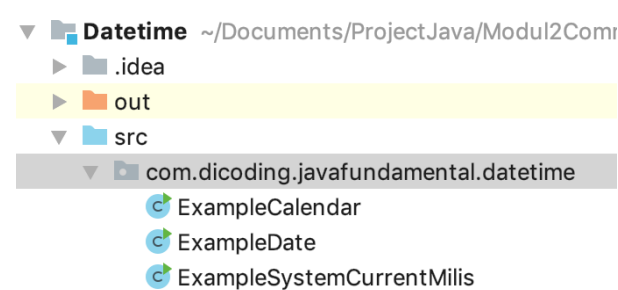
```
Waktu sekarang adalah Mon Jul 22 17:48:25 WIB 2019
```

Seperti halnya pada Date, inisialisasi Calendar dipanggil dengan `Calendar.getInstance()`. Selanjutnya menampilkan waktu yang sekarang dengan memanggil method yang `calendar.getTime()`. Bila kode di atas dijalankan maka akan tampil waktu dan tanggal ketika program dieksekusi.

## Codelab Calendar Menampilkan Secara Terpisah

Calendar memiliki banyak sekali method yang mempermudah untuk mengelola waktu dan tanggal. Contoh di bawah ini memperlihatkan bagaimana Anda dapat mengambil waktu / tanggal spesifik yang diinginkan untuk diolah. Misalnya Anda hanya ingin menampilkan tahun, bulan atau hari saja.

1. Bukalah kembali proyek Datetime.



2. Buatlah kelas baru dengan nama `ExampleSplitCalendar` dan masukkan kode berikut ke dalam kelas tersebut:

```
1. package com.dicoding.javafundamental.datetime;
2.
3. import java.util.Calendar;
4.
5. public class ExampleSplitCalendar {
6.     public static void main(String[] args) {
7.         // Menampilkan waktu sekarang
8.         Calendar calendar = Calendar.getInstance();
9.         System.out.println("Waktu sekarang adalah " + calendar.getTime());
10.        // menampilkan spesifik waktu yang diinginkan
11.        System.out.println("Tanggal : " + calendar.get(Calendar.DATE));
12.        System.out.println("Bulan   : " + calendar.get(Calendar.MONTH));
13.        System.out.println("Tahun   : " + calendar.get(Calendar.YEAR));
14.    }
15. }
```

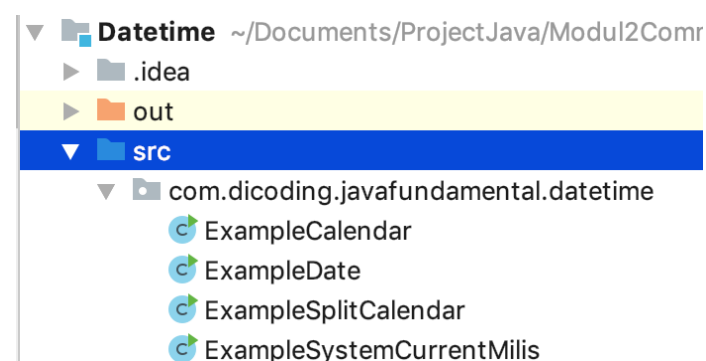
3. Jalankan tersebut maka hasilnya akan jadi seperti ini:

```
Waktu sekarang adalah Mon Jul 22 17:51:37 WIB 2019
Tanggal : 22
Bulan : 6
Tahun  : 2019
```

Calendar memiliki kemampuan untuk menampilkan waktu/tanggal secara spesifik. Method `get(Calendar.DATE)` milik `Calendar` membutuhkan parameter untuk penggunaannya. Seluruh field dapat Anda cek di <https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html>.

## Codelab Calendar Menampilkan Waktu Tertentu yang Diinginkan

1. Bukalah kembali proyek Datetime.



2. Buatlah kelas baru dengan nama `ExampleSpecificCalendar` dan masukkan kode berikut ke dalam kelas tersebut:

```
1. package com.dicoding.javafundamental.datetime;
2.
3. import java.util.Calendar;
4.
5. public class ExampleSpecificCalendar {
6.     public static void main(String[] args) {
7.
8.         // Menampilkan waktu sekarang
9.         Calendar calendar = Calendar.getInstance();
10.        System.out.println("Waktu sekarang adalah " + calendar.getTime());
11.
12.        // Menampilkan waktu 15 hari yang lalu
13.        calendar.add(Calendar.DATE, -15);
14.        System.out.println("15 hari yang lalu: " + calendar.getTime());
15.    }
```

3. Jalankan tersebut maka hasilnya akan jadi seperti ini:

```
Waktu sekarang adalah Mon Jul 22 17:55:36 WIB 2019
15 hari yang lalu: Sun Jul 07 17:55:36 WIB 2019
4 bulan kemudian: Thu Nov 07 17:55:36 WIB 2019
2 tahun kemudian: Sun Nov 07 17:55:36 WIB 2021
```

Salah satu metode yang juga tersedia pada Calendar adalah `add()` yang memiliki 2 parameter yaitu calendar field dan banyaknya waktu yang ingin ditambahkan. Seperti pada kode di atas untuk melihat waktu 15 hari yang lalu, kita bisa gunakan `calendar.add(Calendar.DATE, -15)` di mana -15 adalah 15 hari yang lalu dari waktu saat ini. Begitu juga untuk waktu bulan dan tahun.

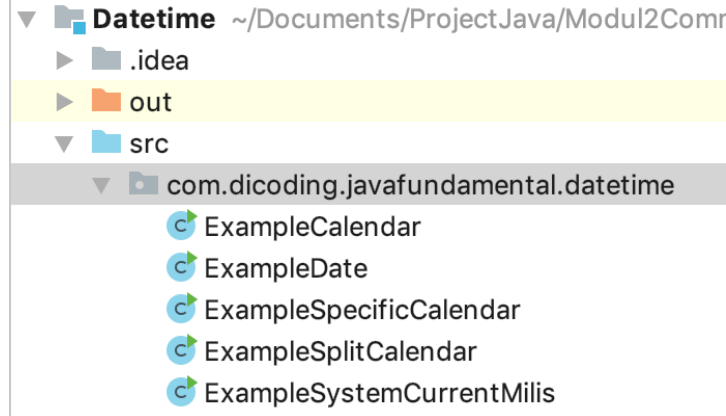
## SimpleDateFormat

Dalam menampilkan suatu tanggal dan waktu, biasanya format tampilan di setiap kebutuhan akan berbeda-beda. Kalau hanya untuk sekedar menampilkan waktu dan tanggal, kita dapat menggunakan date atau calendar.

Bagaimana bila menampilkan format yang tidak biasa? Atau Anda ingin parsing data dari sebuah String waktu dan tanggal dengan format yang tidak biasa? Di sini `SimpleDateFormat` memang berfokus pada memformat dan parsing data tanggal dan waktu serta normalisasi waktu.

## Codelab Menampilkan Waktu Dengan Format yang Ditetapkan.

1. Bukalah kembali proyek Datetime.



2. Buatlah kelas baru dengan nama `ExampleSimpleDateFormat` dan masukkan kode berikut ke dalam kelas tersebut:

```
1. package com.dicoding.javafundamental.datetime;
2.
3. import java.text.SimpleDateFormat;
4. import java.util.Date;
5.
6. public class ExampleSimpleDateFormat {
7.     public static void main(String[] args) {
8.
9.         SimpleDateFormat sdf = new SimpleDateFormat("dd/M/yyyy");
10.        String dateFormatted = sdf.format(new Date());
11.        System.out.println("Format tanggal default : " + new Date());
12.        System.out.println("Format tanggal dengan format : " + dateFormatted);
13.    }
14. }
```

3. Jalankan tersebut maka hasilnya akan jadi seperti ini:

```
Format tanggal default : Mon Jul 22 17:59:24 WIB 2019
Format tanggal dengan format : 22/7/2019
```

Saat inisialisasi `SimpleDateFormat`, kita wajib mengisi parameter pattern. Pattern ini ("dd/M/yyyy") nantinya dijadikan acuan output dari waktu tanggal yang akan diparsing. Kita melakukan inisialisasi atas `SimpleDateFormat new SimpleDateFormat("dd/M/yyyy")`. Artinya output akan menampilkan (tanggal dalam 2 digit) / (Bulan 1 digit) / (Tahun 4 digit).

[< Sebelumnya](#)

[Selanjutnya >](#)