

Penerapan Tanggal di Javadengan Memanfaatkan Class Date, Calendar, dan SimpleDateFormat

Mudafiq Riyan Pratama

me@mudafiqriyan.net

http://www.mudafiqriyan.net

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Menerapkan tanggal di Java yaitu dengan memanfaatkan class Date. Sedangkan SimpleDateFormat berguna untuk mengubah format tanggal yang ditampilkan sesuai keinginan apakah **MM-dd-yyyy** (01-08-2014) atau **MM/dd/yyyy** (01/08/2014) atau format tanggal Indonesia yaitu **dd-MM-yyyy** (08-01-2014) dan **dd MMMM yyyy** (08 January 2014). Pemanfaatan tanggal pada berbagai kasus tentu sangat dibutuhkan seperti pada sistem reservasi hotel untuk me-record tanggal *check-in* dan *check-out*. Pada artikel ini dibahas beberapa kasus penerapan tanggal, mulai dari mengambil tanggal hari ini, tanggal esok hari, tanggal kemarin hari, tanggal beberapa hari kedepan, tanggal beberapa minggu kedepan, tanggal beberapa bulan kedepan, dan lain sebagainya.

Pendahuluan

Pada dasarnya “tanggal” merupakan sebuah komponen yang sangat penting dalam pengembangan software. Contohnya: Sistem Rental Movie, Sistem POS (*Point of Sales*), Sistem Perpustakaan, Sistem Reservasi Hotel, Sistem Tiket Pesawat, dan lain-lain. Semua sistem yang disebutkan tadi sangat membutuhkan penerapan tanggal. Sistem Rental Movie membutuhkan *record* untuk tanggal sewa dan tanggal pengembalian, Sistem POS digunakan untuk mencatat tanggal penjualan dan pembelian barang, sedangkan penggunaan tanggal pada Sistem Reservasi Hotel diterapkan untuk tanggal *check-in* dan *check-out*. Dan masih banyak lagi sistem yang membutuhkan penggunaan tanggal.

Java telah menyediakan class-class yang dapat dimanfaatkan oleh developer dalam membuat aplikasi. Salah satunya adalah class **Date**, yaitu sebuah class yang dapat digunakan untuk mendapatkan tanggal. Akan tetapi dalam penerapannya nanti akan membutuhkan class lain sebagai penunjang dalam penerapan tanggal di Java.

Dalam beberapa kasus tentu membutuhkan untuk mendapatkan tanggal hari ini, tanggal esok hari, tanggal kemarin hari, tanggal beberapa hari kedepan, tanggal beberapa minggu kedepan, tanggal beberapa bulan kedepan, dan lain sebagainya. Dan beberapa kasus penerapan tanggal akan dikupas pada artikel ini.

Pembahasan

1.1. Penerapan Date di Java

Untuk menerapkan Date di Java sangat sederhana, yaitu tinggal membuat objek dari class Date.

```
import java.util.Date;

/**
 *
 * @author mudafiq
 */
public class Tanggal {
    public static void main(String[] args) {
        Date tanggal = new Date();
        System.out.println(tanggal);
    }
}
```

Outpunya:

```
Wed Jan 11 11:32:36 ICT 2014
```

Outputnya kok banyak gitu? ya itu format standar dari class Date yang menampilkan hari, bulan, tanggal, jam, menit, detik, format waktu dan tahun. ICT disitu berarti *IndoChina Time*.

1.2. Mengubah Format Tanggal di Java

Sebelumnya telah dapat menampilkan tanggal hari ini, akan tetapi format yang ditampilkan terlalu banyak. Bagaimana kalau hanya ingin mendapatkan tanggalnya saja. Format tanggal itu beragam: **MM-dd-yyyy** (01-08-2014) atau **MM/dd/yyyy** (01/08/2014) atau format tanggal Indonesia yaitu **dd-MM-yyyy** (08-01-2014) dan **dd MMMM yyyy** (08 January 2014). Maka kita harus mengatur format tanggal yang akan ditampilkan. Untuk mengatur format tanggal di Java dibutuhkan class lain yaitu **SimpleDateFormat**.

```
import java.text.SimpleDateFormat;
import java.util.Date;

/**
 *
 * @author mudafiq
 */
public class Tanggal {
    public static void main(String[] args) {
        Date tanggal = new Date();
        SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
        System.out.println(sdf.format(tanggal));
    }
}
```

Outputnya:

11-01-2014

Kalau ingin menampilkan format tanggal seperti ini **08 January 2014** bagaimana? maka anda hanya tinggal mengganti formatnya menjadi dd MMMM yyyy.

```
Date tanggal = new Date();
SimpleDateFormat sdf = new SimpleDateFormat("dd MMMM yyyy");
System.out.println(sdf.format(tanggal));
```

Outputnya:

11 January 2014

Jika hanya ingin menampilkan 3 huruf pertama dari nama-nama bulan, seperti: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec, maka penulisan format bulannya menggunakan 3 huruf M, yaitu dd MMM yyyy

```
Date tanggal = new Date();
SimpleDateFormat sdf = new SimpleDateFormat("dd MMM yyyy");
System.out.println(sdf.format(tanggal));
```

Outputnya:

11 Jan 2014

1.3. Menampilkan Tanggal Pada Beberapa Hari Kemudian dan Sebelumnya

Pada kasus tertentu, membutuhkan untuk mendapatkan tanggal pada hari-hari setelah dan sebelumnya. Contohnya pada sistem rental movie, ketika melakukan record peminjaman pada hari ini, tentu akan secara otomatis tercatat tanggal pengembaliannya pada tanggal berapa. Misalkan ditentukan pada 3 hari setelah peminjaman, maka tanggal hari ini ditambahkan dengan 3 hari kemudian untuk pengembaliannya. Hal tersebut berkaitan dengan denda yang harus dibayar jika pengembaliannya melewati tanggal yang telah ditentukan.

Untuk menampilkan tanggal pada hari sebelum dan setelahnya, membutuhkan class lagi yaitu class **Calendar**. Begini cara penerapannya:

```
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

/**
 *
 * @author mudafiq
 */
public class Tanggal {

    public static void main(String[] args) {
        SimpleDateFormat sdfdate = new SimpleDateFormat("dd-MM-yyyy");
        Date tanggal = new Date();
        System.out.println("Sekarang tanggal: " + sdfdate.format(tanggal));

        Calendar cal = Calendar.getInstance();
        cal.add(Calendar.DATE, 3); //menghitung 3 hari setelah hari ini
        Date tigaharilagi = cal.getTime(); //membuat variabel yang bertipe Date yang
        System.out.println("Tiga hari lagi tanggal: "+sdfdate.format(tigaharilagi));
    }
}
```

Outputnya:

```
Sekarang tanggal: 11-01-2014
Tiga hari lagi tanggal: 14-01-2014
```

Jika mendapatkan hari-hari sebelumnya bagaimana? tinggal mengubah angka pada pemanggilan method **add()**. Misalkan 3 hari sebelumnya, lihat code berikut:

```
public static void main(String[] args) {
    SimpleDateFormat sdfdate = new SimpleDateFormat("dd-MM-yyyy");
    Date tanggal = new Date();
    System.out.println("Sekarang tanggal: " + sdfdate.format(tanggal));

    Calendar cal = Calendar.getInstance();
    cal.add(Calendar.DATE, -3); //menghitung 3 hari sebelum hari ini
    Date tigaharilalu = cal.getTime(); //membuat variabel yang bertipe Date yar
    System.out.println("Tiga hari sebelumnya: "+sdfdate.format(tigaharilalu));
}
```

Outputnya:

```
Sekarang tanggal: 11-01-2014
Tiga hari sebelumnya: 08-01-2014
```

Jika ingin menampilkan tanggal pada minggu depan tinggal memberi nilai 7 pada pemanggilan method **add()**. Begitupula untuk seminggu yang lalu, maka tinggal diberi nilai -7.

1.4. Menampilkan Tanggal Pada Beberapa Bulan Sebelum dan Setelahnya

Kalau sebelumnya menampilkan tanggal pada hari-hari sebelum dan setelahnya, sekarang kita coba untuk menampilkan tanggal pada bulan-bulan sebelum dan setelahnya.

```
public static void main(String[] args) {
    SimpleDateFormat sdfdate = new SimpleDateFormat("dd-MM-yyyy");
    Date tanggal = new Date();
    System.out.println("Sekarang tanggal: " + sdfdate.format(tanggal));

    Calendar cal = Calendar.getInstance();
    cal.add(Calendar.MONTH, 3); //menghitung 3 bulan kemudian
    Date tigabulanlagi = cal.getTime(); //membuat variabel yang bertipe Date yang
    System.out.println("Tiga bulan lagi tanggal: "+sdfdate.format(tigabulanlagi));
}
```

Outputnya:

```
Sekarang tanggal: 11-01-2014
Tiga bulan lagi tanggal: 11-04-2014
```

Jika pada 3 bulan yang lalu, hanya tinggal mengubah nilai dari method **add()** menjadi -3.

```
public static void main(String[] args) {
    SimpleDateFormat sdfdate = new SimpleDateFormat("dd-MM-yyyy");
    Date tanggal = new Date();
    System.out.println("Sekarang tanggal: " + sdfdate.format(tanggal));

    Calendar cal = Calendar.getInstance();
    cal.add(Calendar.MONTH, -3); //menghitung 3 bulan yang lalu
    Date tigabulanlalu = cal.getTime(); //membuat variabel yang bertipe Date yang
    System.out.println("Tiga bulan lagi tanggal: "+sdfdate.format(tigabulanlalu));
}
```

Outputnya:

```
Sekarang tanggal: 11-01-2014  
Tiga bulan lagi tanggal: 11-10-2013
```

1.5. Menampilkan Tanggal Pada Beberapa Tahun Sebelum dan Setelahnya

Konsepnya sama dengan hari dan bulan yang telah dibahas diatas, yang berubah hanya pada pendeklarasian method **add()**. Jika pada tahun, maka code pemanggilan method **add()** nya adalah seperti ini:

```
Calendar cal = Calendar.getInstance();  
cal.add(Calendar.YEAR, 3);  
Date tigatahun = cal.getTime();
```

Jika pada 3 tahun sebelumnya, maka nilai pada method **add()** diganti dengan nilai -3.

=0=0=0=0=0=0=0=0= *Semoga Bermanfaat* =0=0=0=0=0=0=0=0=

Penutup

Semoga tulisan ini dapat bermanfaat. Dan jangan lupa untuk mengunjungi website saya www.mudafiqriyan.net dan follow twitter saya: @mudafiq

Biografi Penulis



Mudafiq Riyan Pratama. Adalah seorang anak laki-laki dari Nurkholis (bapak) dan Susriyanti (ibu) ini terlahir di Jember pada tanggal 9 Mei 1989. Mengawali pendidikan TK dan SD di Jenggawah, Jember. Kemudian menempuh SMP di SMPN 6 Jember, yang kemudian dilanjutkan ke SMAN 2 Jember. Kuliah S1 di Universitas Muhammadiyah Malang dengan mengambil jurusan Teknik Informatika. Saat itulah penulis mengawali masuk ke dunia informatika yang sebelumnya tidak pernah tau tentang dunia tersebut. Saat itulah ketertarikan dengan informatika makin membesar. Penulis mulai meniti karir di bidang IT sebagai *programmer freelance*. Hingga akhirnya saat ini beliau berkarir sebagai tenaga pengajar (dosen) di Universitas Muhammadiyah Jember.