

---

# String Class



# String Class

- Seperti yang pernah dibahas di materi Java Dasar, String adalah object, artinya dia memiliki representasi class nya
- Ada banyak sekali method yang bisa kita gunakan di String, kita bisa melihat detail method apa aja yang tersedia di halaman dokumentasi javadoc nya
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/String.html>



## Method di String Class

Method	Keterangan
String toLowerCase()	Membuat string baru dengan format lower case
String toUpperCase()	Membuat string baru dengan format upper case
int length()	Mendapatkan panjang string
boolean startsWith(value)	Mengecek apakah dimulai dengan string value
boolean endsWith(value)	Mengecek apakah diakhiri dengan string value
String[] split(value)	Memotong string dengan string value

---

# StringBuffer dan StringBuilder



# Immutable String

- String adalah tipe data immutable, artinya tidak bisa berubah isinya, saat kita mengubah string, sebenarnya yang dilakukan di Java adalah membuat String baru.
- Jika kita ingin memanipulasi String dalam jumlah banyak, sangat tidak disarankan menggunakan String, karena akan memakan memory yang cukup besar, untuk kasus seperti ini, disarankan menggunakan StringBuffer atau StringBuilder



# StringBuffer vs StringBuilder

- Kemampuan StringBuffer dan StringBuilder cukup sama, bisa digunakan untuk memanipulasi String
- Yang membedakan adalah, StringBuffer itu thread safe, sedangkan StringBuilder tidak thread safe
- Jika kita ingin memanipulasi String secara paralel bersamaan, disarankan menggunakan StringBuffer, namun jika tidak butuh paralel, cukup gunakan StringBuilder
- Karena StringBuffer dibuat agar thread safe, maka secara otomatis performanya lebih lambat dibandingkan StringBuilder
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/StringBuffer.html>
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/StringBuilder.html>



## Kode : Menggunakan StringBuilder

```
StringBuilder builder = new StringBuilder();  
builder.append("Eko");  
builder.append(" ");  
builder.append("Kurniawan");  
builder.append(" ");  
builder.append("Khannedy");  
  
String fullName = builder.toString();  
  
}  
}
```

---

# StringJoiner Class





# StringJoiner Class

- StringJoiner adalah class yang bisa digunakan untuk membuat String sequence yang dipisahkan dengan delimiter
- StringJoiner juga mendukung prefix dan suffix jika kita ingin menambahkannya
- Ini sangat bagus ketika ada kasus misal kita ingin mem-print Array dengan format yang kita mau misalnya
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/StringJoiner.html>



## Kode : String Joiner

```
String[] names = {"Eko", "Kurniawan", "Khannedy"};
StringJoiner joiner = new StringJoiner(
    delimiter: "||", prefix: "[", suffix: "]"
);

for (var name : names) {
    joiner.add(name);
}

System.out.println(joiner.toString());
```

---

# StringTokenizer Class



# StringTokenizer Class

- StringTokenizer class adalah class yang bisa digunakan untuk memotong String menjadi token atau string yang lebih kecil
- Kita bisa memotong String dengan delimiter yang kita mau
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/StringTokenizer.html>



## Kode : StringTokenizer Class

```
String name = "Eko Kurniawan Khannedy";
StringTokenizer tokenizer = new StringTokenizer(name, " ");

while (tokenizer.hasMoreTokens()) {
    String token = tokenizer.nextToken();
    System.out.println(token);
}
}
```

---

# Number Class



# Number Class

- Semua number class yang bukan primitif memiliki parent class yang sama, yaitu class Number
- Class number memiliki banyak method yang bisa digunakan untuk mengkonversi ke tipe number lain
- Hal ini memudahkan kita untuk mengkonversi object number dari satu tipe ke tipe number lainnya
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/Number.html>



## Method di Number Class

Method	Keterangan
byte byteValue()	Mengubah menjadi tipe byte
double doubleValue()	Mengubah menjadi tipe double
float floatValue()	Mengubah menjadi tipe float
int intValue()	Mengubah menjadi int value
long longValue()	Mengubah menjadi long value
short shortValue()	Mengubah menjadi short value





# Konversi String ke Number

- Long, Integer, Short dan Byte memiliki static method untuk melakukan konversi dari String ke number
- `parseXxx(string)` digunakan untuk mengkonversi dari string ke tipe data number primitif
- `valueOf(string)` digunakan untuk mengkonversi dari string ke tipe data number non primitif
- Method ini akan throw `NumberFormatException` jika ternyata gagal melakukan konversi String ke number



# Math Class



# Math Class

- Class Math merupakan class utilities yang berisikan banyak sekali static method untuk operasi numerik, seperti trigonometric, logarithm, akar pangkat, dan lain-lain
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/Math.html>



## Method di Math Class

Method	Keterangan
<code>double cos(double)</code>	Menghitung cos di trigonometric
<code>double sin(double)</code>	Menghitung sin di trigonometric
<code>double tan(double)</code>	Menghitung tan di trigonometric
<code>min(number1, number2)</code>	Mengambil nilai terkecil
<code>max(number1, number1)</code>	Mengambil nilai terbesar
...dan masih banyak	

---

# Big Number



# Big Number

- Jika kita ada kebutuhan untuk menggunakan angka yang besar sehingga melebihi kapasitas Long dan Double, di Java sudah disediakan class untuk handle data besar tersebut
- BigInteger adalah class untuk handle tipe data Integer, dan
- BigDecimal adalah class untuk handle tipe data floating point
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/math/BigInteger.html>
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/math/BigDecimal.html>



## Method di BigInteger & BigDecimal

Method	Operator
add	+
subtract	-
multiply	*
divide	/
mod	%
...dan masih banyak lagi	



## Kode : BigInteger

```
BigInteger a = new BigInteger(val: "10000000000000000000");  
BigInteger b = new BigInteger(val: "10000000000000000000");  
BigInteger result = a.add(b);  
  
System.out.println(result);  
  
}  
}
```



---

# Scanner Class



# Scanner Class

- Scanner sebenarnya bagian dari Java IO (Input Output), dan ini akan dibahas di materi terpisah
- Namun sekarang kita akan bahas sekilas tentang class Scanner
- Class Scanner hadir sejak Java 5
- Class Scanner adalah class yang bisa digunakan untuk membaca input, entah dari file, console, dan lain-lain
- Class Scanner ini cocok untuk dijadikan object untuk membaca input user saat kita belajar membuat program Java menggunakan console / terminal
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/Scanner.html>



## Method di Scanner Class

Method	Keterangan
nextLine()	Membaca string
nextInt()	Membaca int
nextLong()	Membaca long
nextBoolean()	Membaca boolean
...dan masih banyak	



## Kode : Menggunakan Scanner

```
import java.util.Scanner;

public class ScannerApp {
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Nama : ");
        String nama = scanner.nextLine();

        System.out.println("Hello " + nama);
    }
}
```

---

# Date & Calendar Class



# Date & Calendar

- Tiap bahasa pemrograman biasanya memiliki representasi tanggal, di Java juga sama, ada class Date & Calendar yang bisa kita gunakan sebagai representasi tanggal
- Sebenarnya di Java 8 sudah ada cara manipulasi tanggal yang baru menggunakan Java Date Time API, namun itu akan kita bahas di course terpisah
- Sekarang kita akan fokus menggunakan class Date dan Calendar



# Hubungan Date dan Calendar

- Class Date adalah class representasi tanggal sampai presisi milisecond
- Namun di class Date sudah banyak method-method yang di deprecated, sehingga untuk memanipulasi date tanggal, kita sekarang harus melakukan kombinasi antara class Date dan Calendar
- Sederhananya Date untuk representasi tanggal, dan Calendar untuk memanipulasi tanggal
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/Date.html>
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/Calendar.html>



## Kode : Menggunakan Date

```
Date date = new Date();  
long millisecond = date.getTime();  
  
System.out.println(date);  
System.out.println(millisecond);  
}  
}
```







## Kode : Menggunakan Calendar

```
Calendar calendar = Calendar.getInstance();
calendar.set(Calendar.YEAR, 2000);
calendar.set(Calendar.MONTH, Calendar.JANUARY);
calendar.set(Calendar.DAY_OF_MONTH, 3);
calendar.set(Calendar.HOUR_OF_DAY, 0);
calendar.set(Calendar.MINUTE, 0);
calendar.set(Calendar.SECOND, 0);
calendar.set(Calendar.MILLISECOND, 0);

Date date = calendar.getTime();
long millisecond = date.getTime();
```

---

# System Class



# System Class

- Class System adalah class yang berisikan banyak utility static method di Java, contohnya sebelumnya kita sudah sering menggunakan method println milik field out di class System.
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/System.html>



## Method di System Class

Method	Keterangan
String getenv(key)	Mendapatkan environment variable sistem operasi
void exit(status)	Menghentikan program Java
long currentTimeMillis()	Mendapatkan waktu saat ini dalam milisecond
long nanoTime()	Mendapatkan waktu saat ini dalam nanosecond
void gc()	Menjalankan Java garbage collection
... dan masih banyak	

---

# Runtime Class



# Runtime Class

- Ketika aplikasi Java kita berjalan, kita bisa melihat informasi environment tempat aplikasi Java berjalan
- Informasi itu terdapat di class Runtime.
- Class Runtime tidak bisa dibuat, secara otomatis Java akan membuat single object. Kita bisa mengakses object tersebut menggunakan static method `getRuntime()` milik class Runtime
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/Runtime.html>



## Method di Runtime Class

Method	Keterangan
<code>int availableProcessors()</code>	Mendapatkan jumlah core cpu
<code>long freeMemory()</code>	Mendapatkan jumlah memory bebas di JVM
<code>long totalMemory()</code>	Mendapatkan jumlah total memory di JVM
<code>long maxMemory()</code>	Mendapatkan jumlah maksimum memory di JVM
<code>void gc()</code>	Menjalankan garbage collector untuk menghilangkan data di memory yang sudah tidak terpakai

---

# UUID Class





# UUID Class

- Saat membuat aplikasi, kadang kita ada kasus ingin membuat data unique, misal untuk kebutuhan data primary key misalnya
- Java menyediakan sebuah class UUID atau singkatan dari Universally Unique Identifier.
- UUID adalah format standard untuk membuat unique value yang telah terjamin
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/UUID.html>
- <https://www.ietf.org/rfc/rfc4122.txt>



## Kode : UUID

```
for (int i = 0; i < 100; i++) {  
    UUID uuid = UUID.randomUUID();  
    System.out.println(uuid);  
}
```

```
}
```

```
}
```

```
|
```

---

# Base64 Class



# Base64 Class

- Sejak Java 8, Java sudah menyediakan class untuk melakukan encoding base64
- Buat programmer web pasti tahu tentang base64, yaitu encoding yang bisa digunakan untuk mengubah binary data ke text yang aman
- Aman disini bukan dari sisi security, tapi aman dari kesalahan parsing
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/Base64.html>
- <https://www.ietf.org/rfc/rfc4648.txt>



## Kode : Base64

```
String query = "belajar()    pemrograman()    java";

String encode = Base64.getEncoder().encodeToString(query.getBytes());
System.out.println(encode);

String decode = new String(Base64.getDecoder().decode(encode));
System.out.println(decode);

}

}
```

---

# Objects Class



# Objects Class

- Awas jangan tertukar, ini class Objects, bukan Object
- Objects adalah class utility yang berisikan banyak static method yang bisa kita gunakan untuk operasi object atau melakukan pengecekan sebelum operasi nya dilakukan
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/Objects.html>



## Kode : Tanda Objects

```
if (product != null) {  
    product.toString();  
}  
  
if (product != null) {  
    product.hashCode();  
}  
  
if (product != null) {  
    product.equals(product2);  
}
```





## Kode : Dengan Objects

```
var string = Objects.toString(product);  
var hashCode = Objects.hashCode(product);  
var equals = Objects.equals(product, product2);
```

```
}
```

```
}
```

```
|
```

---

# Random Class



# Random Class

- Random class adalah class yang bisa kita gunakan untuk men-generate random number
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/Random.html>



## Kode : Random

```
Random random = new Random();

for (int i = 0; i < 100; i++) {
    var value = random.nextInt( bound: 1000);
    System.out.println(value);
}
}
```

---

# Properties Class

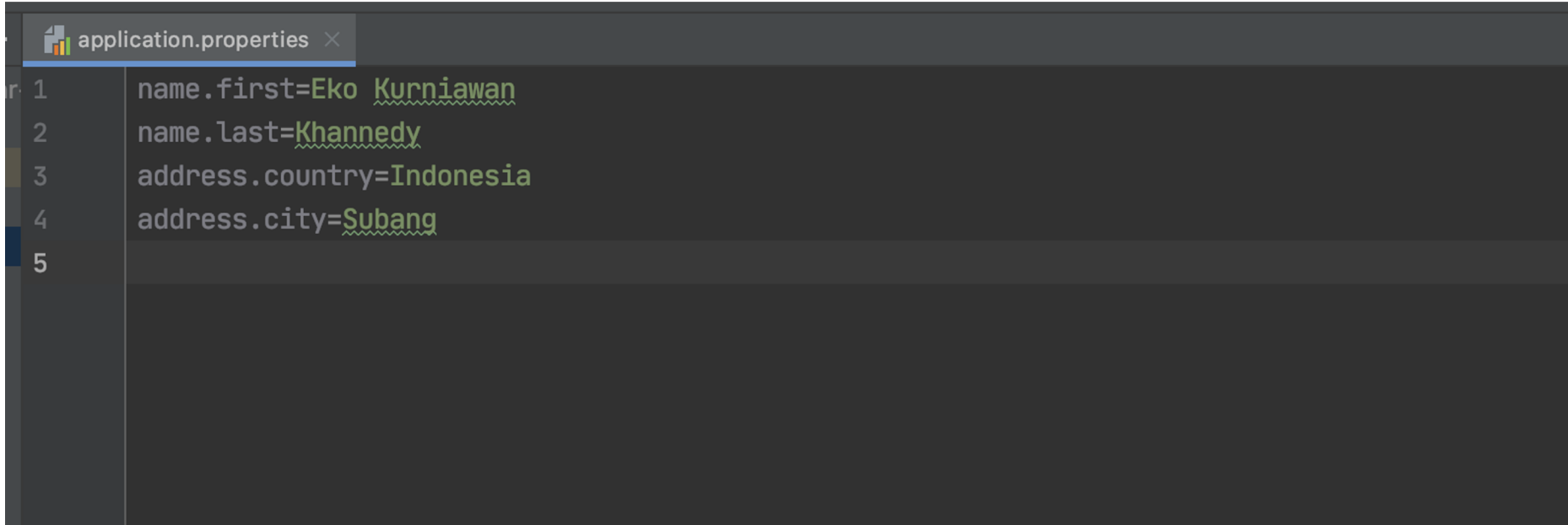


# Properties File

- Kebanyakan aplikasi Java akan menyimpan konfigurasi file dalam bentuk properties file
- Properties file adalah file yang berisi key value yang dipisahkan dengan tanda sama dengan (=)
- Properties file bisa kita gunakan untuk menyimpan konfigurasi aplikasi kita



# Contoh Properties File



```
application.properties x
1 name.first=Eko Kurniawan
2 name.last=Khannedy
3 address.country=Indonesia
4 address.city=Subang
5
```



# Properties Class

- Properties Class adalah class yang bisa kita gunakan untuk mengambil atau menyimpan informasi ke file properties
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/Properties.html>





## Kode : Properties

```
try {  
    Properties properties = new Properties();  
    properties.load(new FileInputStream( name: "application.properties"));  
  
    System.out.println(properties.getProperty("name.first"));  
    System.out.println(properties.getProperty("name.last"));  
  
    properties.put("hobby", "Coding");  
    properties.store(new FileOutputStream( name: "application.properties"), comments: "Komentar");  
} catch (IOException exception) {
```

---

# Arrays Class



# Arrays Class

- Arrays class adalah class yang berisikan static method yang bisa kita gunakan untuk memanipulasi data array, seperti pencarian dan pengurutan
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/Arrays.html>



## Method di Arrays Class

Method	Keterangan
<code>binarySearch(array, value)</code>	Mencari value di array
<code>copyOf(...)</code>	Menyalin data array
<code>equals(array1, array2)</code>	Membandingkan array1 dan array2
<code>sort(array)</code>	Mengurutkan array
<code>toString(array)</code>	Mengembalikan representasi string
... dan masih banyak	

---

# Regular Expression



# Regular Expression

- Regular Expression atau disingkat regex adalah cara untuk melakukan pola pencarian
- Biasanya dilakukan untuk pencarian dalam data String
- Secara sederhana, kita mungkin sudah sering melakukan pencarian text, entah di text editor atau di aplikasi word
- Regex adalah pencarian yang lebih advanced dibanding pencarian text biasanya, misal kita ingin mencari semua kata yang mengandung diawali huruf a dan diakhiri huruf a, dan lain-lain



# Regex Package

- Java sudah menyediakan package `java.util.regex` yang berisikan utilitas untuk melakukan proses regular expression
- Secara garis besar terdapat 2 class yang dapat kita gunakan, yaitu `Pattern` class dan `Matcher` class
- `Pattern` class adalah representasi hasil kompilasi dari pola regular expression yang kita buat
- `Matcher` class adalah engine untuk melakukan pencarian dari pattern yang sudah kita buat



# Aturan Regular Expression

- Aturan regular expression sangat kaya, sehingga kemungkinan tidak bisa dibahas dalam satu materi
- Kita bisa lihat detail aturan-aturannya di halaman javadoc class Pattern
- <https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/regex/Pattern.html>





## Kode : Regular Expression

```
String name = "Eko Kurniawan Khanendy Programmer Zaman Now";

Pattern pattern = Pattern.compile("[a-zA-Z]*[a][a-zA-Z]*");

Matcher matcher = pattern.matcher(name);

while (matcher.find()) {
    System.out.println(matcher.group());
}

}
```