

---

# Pengenalan Spring Logging



# Pengenalan Logging

- Logging sudah kita bahas dengan detail pada materi Java Logging, jadi diharapkan sudah mengikuti materi Java Logging terlebih dahulu sebelum mengikuti materi Spring Logging



# Spring Logging

- Spring sendiri sebenarnya tidak memiliki fitur khusus untuk Logging
- Spring menggunakan library Logging yang sudah tersedia di Java
- Secara default Spring menggunakan library Logback untuk melakukan logging, dan materi Logback sudah kita bahas secara lengkap di kelas Java Logging
- Namun Spring Boot memiliki fitur tambahan untuk mempermudah kita, ketika akan menggunakan Logging di Spring, dan fitur-fitur ini akan kita bahas di kelas ini

---

# Membuat Project



# Membuat Project

<https://start.spring.io/>

---

# Default Logging Setting



# Default Logging Setting

- Saat kita menggunakan Spring Boot, secara default, Spring Boot sudah memiliki setting atau pengaturan default untuk Logging
- Secara default Spring Boot melakukan logging ke Console, dan selain itu Spring Boot juga memiliki Log Format

## Kode : Sample Logging

```
@Slf4j
@SpringBootTest
public class LoggingTest {

    @Test
    void testNothing() {
        log.info("Hello World");
        log.warn("Hello Spring");
        log.error("Hello Programmer Zaman Now");
    }
}
```

You, Moments ago • Uncommitted changes





# Kode : Sample Logging Output

✓ Tests passed: 1 of 1 test – 509 ms

ms	2022-06-10 21:37:56.051	INFO 75653 --- [	main]	p.belajarspringlogging.LoggingTest	: Hello World
ms	2022-06-10 21:37:56.052	WARN 75653 --- [	main]	p.belajarspringlogging.LoggingTest	: Hello Spring
	2022-06-10 21:37:56.052	ERROR 75653 --- [	main]	p.belajarspringlogging.LoggingTest	: Hello Programmer Zaman Now

---

# Logging Level



# Logging Level

- Pada materi Java Logging, kita sudah bahas tentang Logging Level
- Saat menggunakan Logback, biasanya kita bisa ubah level dari logging di file logback.xml, namun Spring Boot memiliki fitur dimana kita bisa menggunakan file Spring application.properties untuk mengubah pengaturan Logging nya

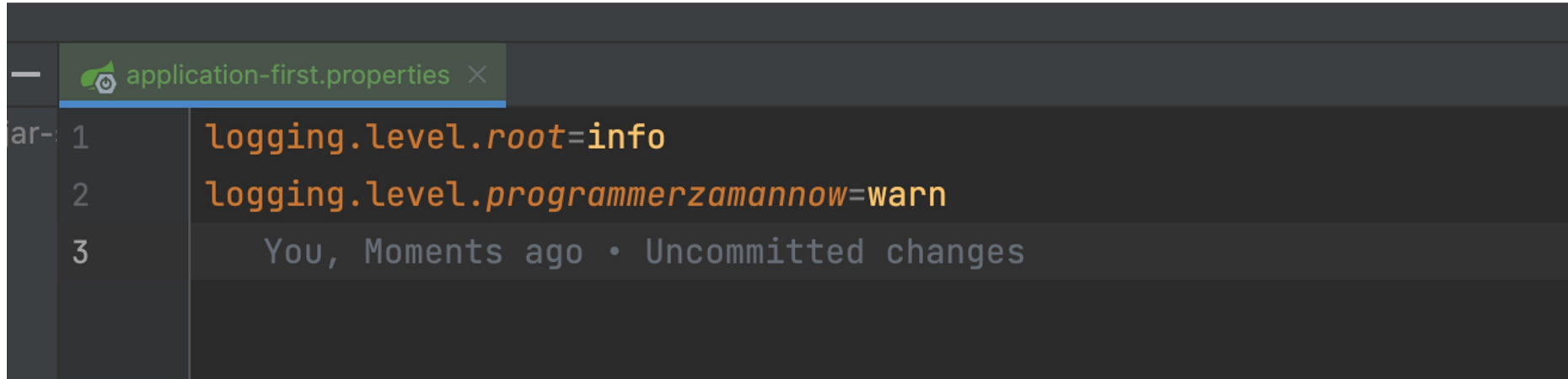


# Logging Properties

- Kita bisa menggunakan prefix `logging.level` untuk mengubah pengaturan logging level, misal:
- `logging.level.root=info`
- `logging.level.nama.package.nya=warn`



## Kode : Application Properties



The screenshot shows an IDE window titled "application-first.properties" with a close button. The code is as follows:

```
1 logging.level.root=info
2 logging.level.programmerzamannow=warn
3 You, Moments ago • Uncommitted changes
```

## Kode : Logging

```
7
8  @Slf4j
9  @SpringBootTest
10 @TestPropertySource("classpath:application-first.properties")
11 public class LoggingTest {
12
13     @Test
14     void testNothing() {
15         log.info("Hello World");
16         log.warn("Hello Spring");
17         log.error("Hello Programmer Zaman Now");
18     }
19 }
```

---

# File Output



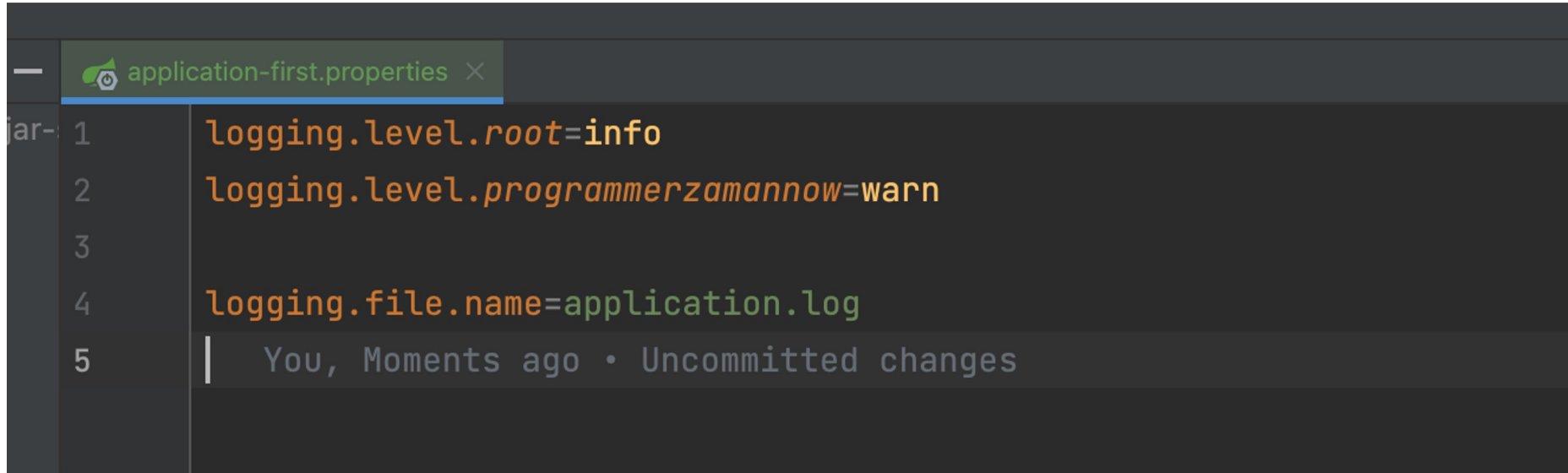
# File Output

- Secara default, Spring Boot melakukan konfigurasi logging output hanya ke Console
- Namun Spring Boot memiliki fitur untuk mengaktifkan logging output ke File, dengan cara menggunakan properties :
- `logging.file.name` untuk menentukan nama file output logging, misal `application.log` atau `/tmp/application.log`
- `logging.file.path` untuk menentukan nama folder output logging, dan secara otomatis akan menggunakan nama file `spring.log`, misal jika menggunakan `/tmp/`, maka secara otomatis file log nya adalah `/tmp/spring.log`





## Kode : Application Properties



The screenshot shows an IDE window titled "application-first.properties" with a close button. The code is as follows:

```
1 logging.level.root=info
2 logging.level.programmerzamannow=warn
3
4 logging.file.name=application.log
5 | You, Moments ago • Uncommitted changes
```

---

# File Rolling



# File Rolling

- Spring Boot juga memiliki fitur untuk mempermudah ketika kita ingin melakukan rolling file logging menggunakan Logback
- Ada banyak pengaturan di application properties yang bisa kita gunakan



# File Rolling Properties

Name	Description
<code>logging.logback.rollingpolicy.file-name-pattern</code>	The filename pattern used to create log archives.
<code>logging.logback.rollingpolicy.clean-history-on-start</code>	If log archive cleanup should occur when the application starts.
<code>logging.logback.rollingpolicy.max-file-size</code>	The maximum size of log file before it is archived.
<code>logging.logback.rollingpolicy.total-size-cap</code>	The maximum amount of size log archives can take before being deleted.
<code>logging.logback.rollingpolicy.max-history</code>	The maximum number of archive log files to keep (defaults to 7).



## Code : Application Properties

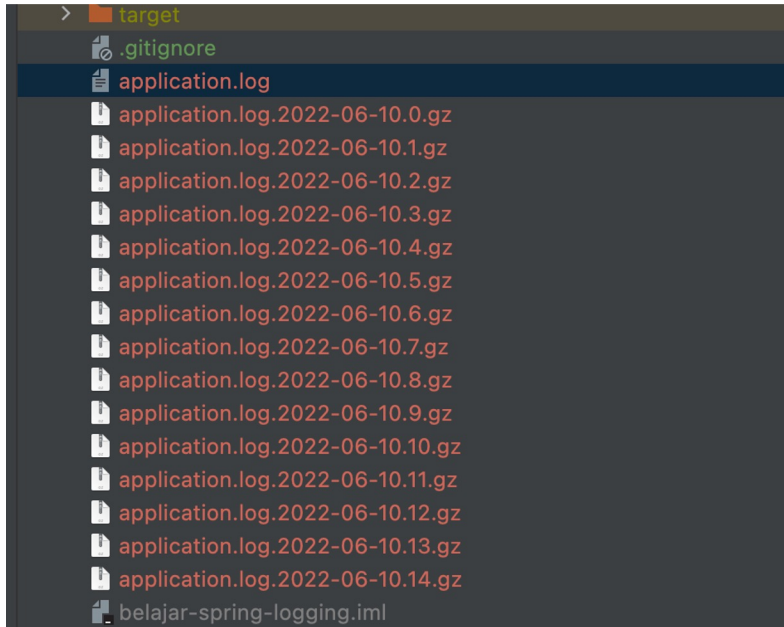
```
application-first.properties x
1 logging.level.root=info
2 logging.level.programmerzamannow=warn
3
4 logging.file.name=application.log
5
6 logging.logback.rollingpolicy.max-file-size=10KB
7 logging.logback.rollingpolicy.max-history=10
8 logging.logback.rollingpolicy.total-size-cap=1GB
9
```

## Kode : Test Long Logging

```
8  @Slf4j
9  @SpringBootTest
10 @TestPropertySource("classpath:/application-first.properties")
11 public class LoggingTest {
12
13     @Test
14     void testLongLogging() {
15         for (int i = 0; i < 100_000; i++) {
16             log.warn("Hello World {}", i);
17         }
18     }
19 }
```



# Hasil File Rolling



—

# Log Group





# Log Group

- Pada beberapa kasus, kadang kita ingin membuat konfigurasi logging level yang sama untuk beberapa package
- Spring Boot memiliki fitur bernama Log Group, dimana kita bisa membuat nama log group untuk beberapa package
- Keuntungannya, kita bisa menggunakan nama log group tersebut ketika melakukan pengaturan logging level
- Kita bisa menggunakan application properties :  
`logging.group.nama=package1,package2,danseterusnya`



# Log Group Spring Boot

Name	Loggers
web	<code>org.springframework.core.codec</code> , <code>org.springframework.http</code> , <code>org.springframework.web</code> , <code>org.springframework.boot.actuate.endpoint.web</code> , <code>org.springframework.boot.web.servlet.ServletContextInitializerBeans</code>
sql	<code>org.springframework.jdbc.core</code> , <code>org.hibernate.SQL</code> , <code>org.jooq.tools.LoggerListener</code>

## Kode : Application Properties

```
application-first.properties x
1 logging.group.pzn=programmerzamannow,com.programmerzamannow,com.khannedy
2
3 logging.level.root=info
4 logging.level.pzn=warn
5
6 logging.file.name=application.log
7
8 logging.logback.rollingpolicy.max-file-size=10KB
9 logging.logback.rollingpolicy.max-history=10
10 logging.logback.rollingpolicy.total-size-cap=1GB
11
```

---

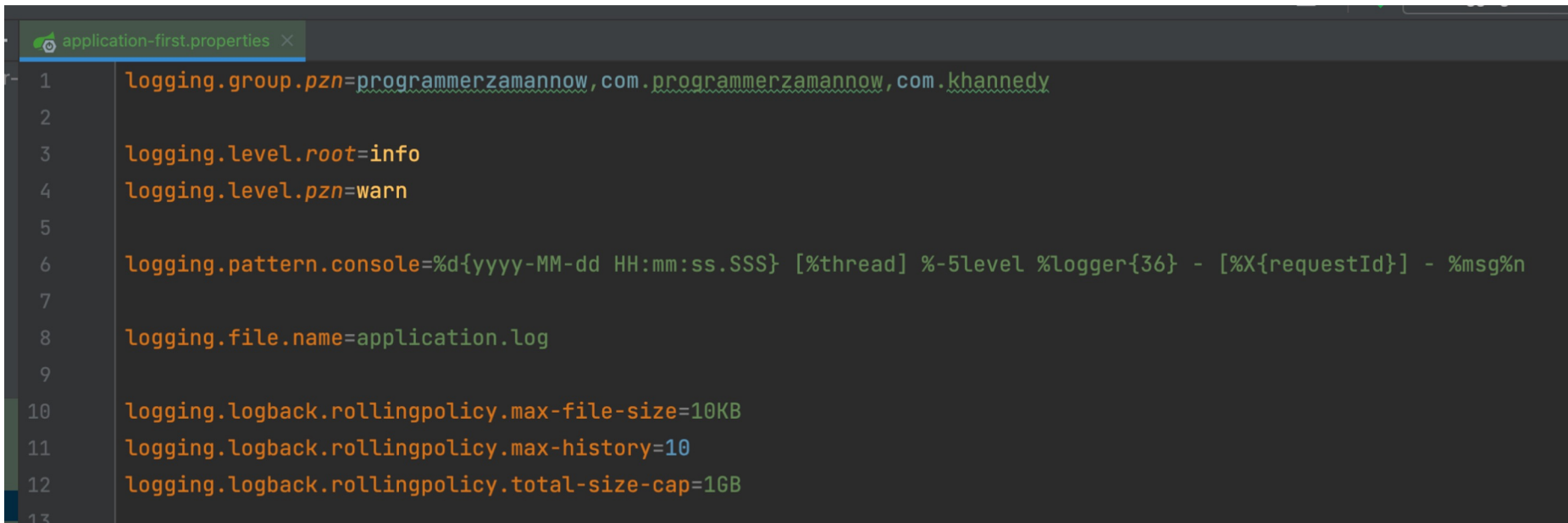
# Log Pattern



# Log Pattern

- Kita juga bisa mengubah default log pattern bawaan Spring Boot jika kita mau
- Kita bisa menggunakan application properties
- `logging.pattern.console` untuk mengubah log pattern console
- `logging.pattern.file` untuk mengubah log pattern file

## Kode : Application Properties

A screenshot of a code editor with a dark theme. The editor has a tab at the top labeled 'application-first.properties' with a close button. The code is written in a monospaced font with syntax highlighting. Line numbers 1 through 13 are visible on the left side of the editor. The code defines logging properties for a group, level, pattern, file, and rolling policy.

```
1 logging.group.pzn=programmerzamannow,com.programmerzamannow,com.khannedy
2
3 logging.level.root=info
4 logging.level.pzn=warn
5
6 logging.pattern.console=%d{yyyy-MM-dd HH:mm:ss.SSS} [%thread] %-5level %logger{36} - [%X{requestId}] - %msg%n
7
8 logging.file.name=application.log
9
10 logging.logback.rollingpolicy.max-file-size=10KB
11 logging.logback.rollingpolicy.max-history=10
12 logging.logback.rollingpolicy.total-size-cap=16B
13
```

---

# Custom Log Configuration



# Custom Log Configuration File

- Jika kita tidak ingin menggunakan fitur bawaan Spring Boot, dan lebih suka membuat konfigurasi logback sendiri, kita juga bisa menambahkan file logback.xml di resources
- Secara otomatis jika Spring Boot mendeteksi terdapat file logback.xml, maka Spring Boot akan membaca konfigurasi Logging dari file tersebut
- Untuk konfigurasi manual logback.xml sudah kita bahas di kelas Java Logging