

Working with Media and PDF

Gold - Chapter 5 - Topic 5

Selamat datang di **Chapter 5 Topic 5**
online course **Back End Java** dari
Binar Academy!

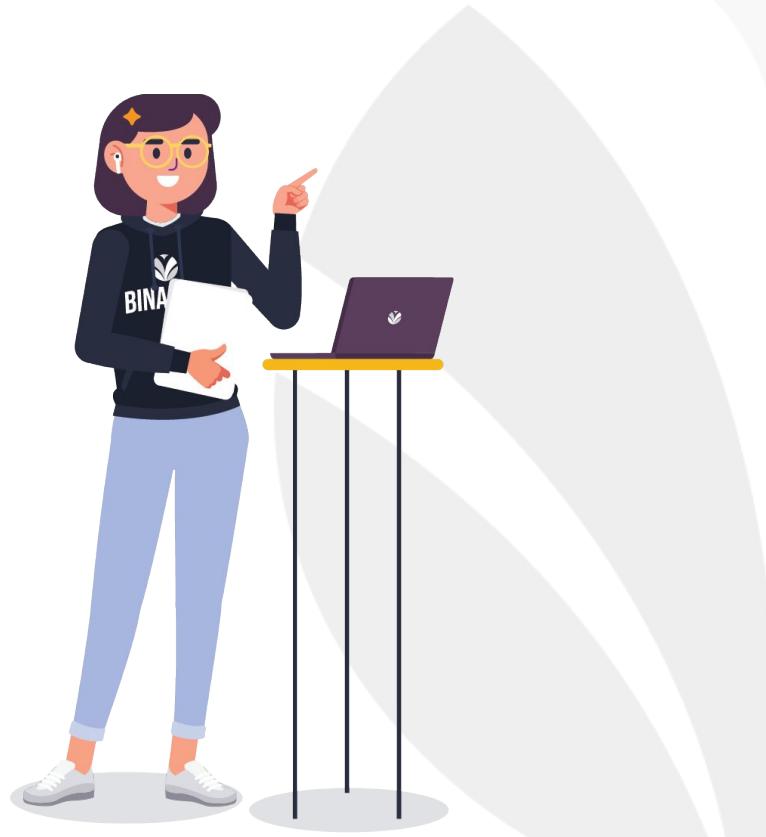


Kita ketemu lagi, Binarian! ☺

Masih inget nggak kemarin belajar apa? yang jawab belajar tentang API Documentation with Swagger kalian bener!

Lanjut pada topic kelima ini, kita bakal mengelaborasi tentang **Working with Media and PDF**. Mulai dari konsep dasar Spring WebFlux, cara upload dan download sebuah file, sampai cara bikin file PDF.

Let's go!



Dari sesi ini, kita bakal bahas hal-hal berikut:

- Konsep Spring WebFlux
- Cara download dan upload file berupa foto dan video
- Menggunakan Jasper Report untuk membuat file PDF



Pergi ke Bali buat lihat teluk.

Yuk kita awali dengan Konsep Spring WebFlux.

Kira-kira Spring WebFlux itu apa, ya?



Spring Webflux adalah sebuah modul~

Dengan kata lain, Spring WebFlux merupakan sebuah **modul dari Spring** yang udah meng-cover pemrograman reactive (reactive programming) pada web.



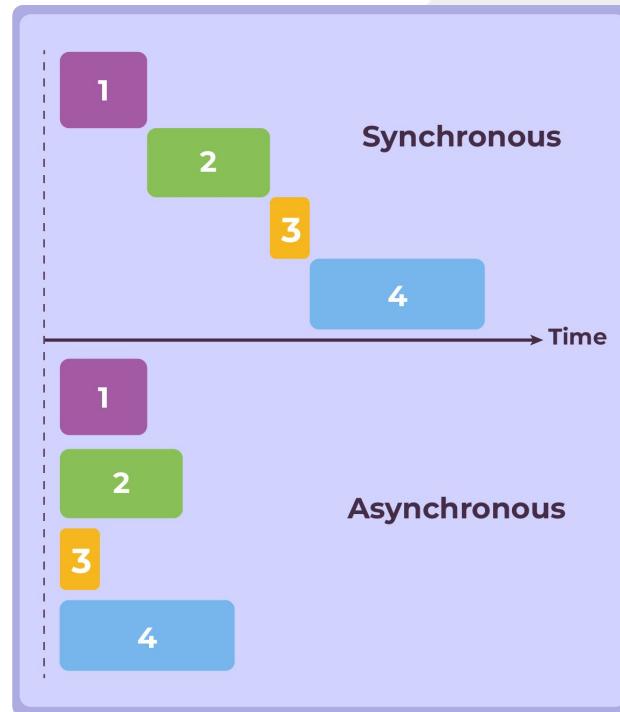
Apa itu reactive programming?

Reactive programming merupakan paradigma pemrograman yang memungkinkan pemrograman meng-handle real time logic.

Misalnya kayak meng-update secara real time.



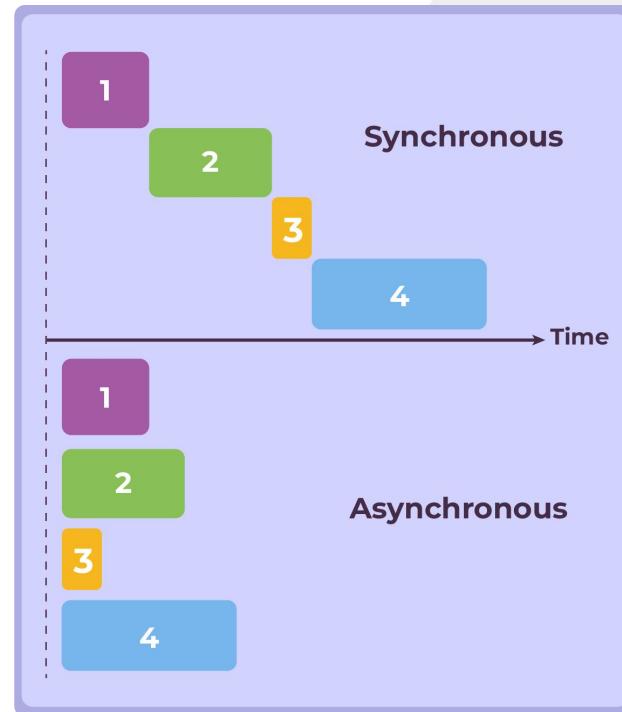
Cukup berbeda dengan paradigma pemrograman Java yang sebelumnya dilakukan (berupa imperative atau code-code yang dieksekusi secara berurutan), Spring WebFlux memungkinkan untuk memproses sebuah request secara **Non Blocking**.



Gimana konsep Non Blocking?

Proses Non Blocking ini berarti **sebuah proses bakal dijalankan tanpa harus menunggu proses lain selesai** atau istilah lainnya yaitu asynchronous.

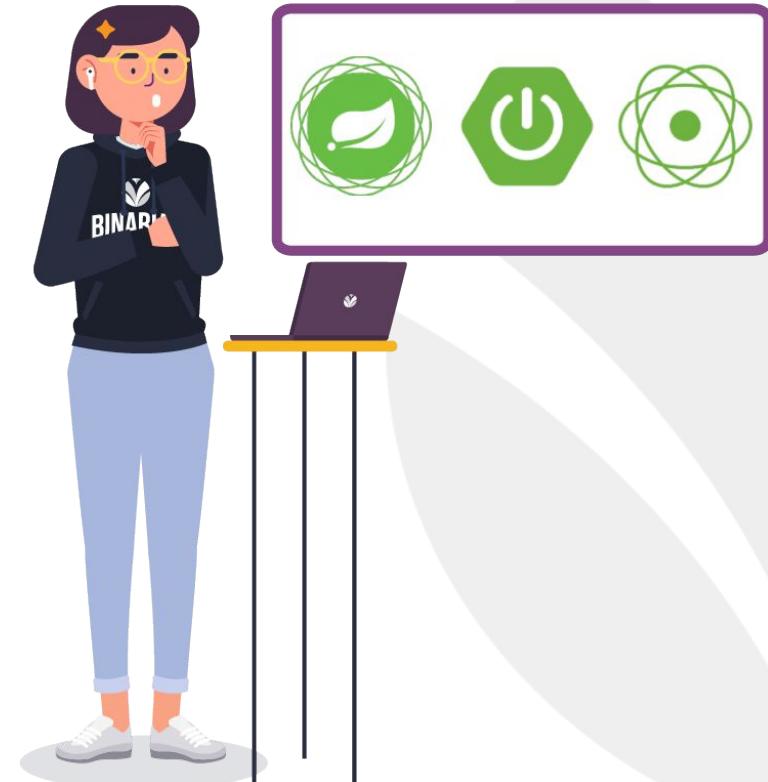
Masalahnya, hal ini nggak bisa di-handle oleh Spring Web biasa. Jadinya, kita butuh banget nih sama Spring WebFlux.



Oh iya kalau kamu bertanya-tanya, apa cuma Spring WebFlux aja yang punya jurus ini?

Jawabannya, no. Modul Spring dengan pemrograman reactive nggak hanya ada di modul Spring WebFlux aja, tapi juga ada di pemrosesan data ke relational database, yaitu Spring Data R2DBC.

Keren, kan?



Reactive programming bikin pemrosesan data jadi lebih cepat~

Manfaat dari penggunaan reactive programming ini adalah pemrosesan yang lebih cepat dan bisa melakukan pemrosesan secara real time.

Contoh penggunaan dari Spring WebFlux ini adalah Video Streaming atau Real time Data Streaming.



Sampai disini, kita tahu bahwa ada dua key point penting, yaitu reactive programming dan Spring WebFlux.

Berhubung rasa ingin tahu kita kuat banget, kita perlu belajar satu-satu dulu, nih. Supaya belajar kita nyaman.

Spring WebFlux yang kita pelajari belum akan dibahas secara mendalam karena kita harus belajar tentang reactive programming dulu.



Dari situ, pada topik kali ini kita hanya berfokus pada **implementasi dari media handling** aja. Yaitu, video handling pakai Spring WebFlux

Referensi tentang video handling pakai Spring WebFlux bisa kamu pelajari [di sini](#), yaa~



Kamu pasti sering download atau upload sebuah file, kan? Entah itu tentang tugas ataupun kerjaan yang lainnya.

Mirip-mirip tipis, bedanya kita bakal bahas cara **Download and Upload File** tapi kali ini pakai REST API.

Kayak gimana sih caranya?



Langsung aja! Begini cara Download dan Upload sebuah File pakai REST API~

Supaya bisa melakukan upload file, maka sebuah method yang jadi endpoint sebaiknya punya parameter MultipartFile dengan annotation @RequestParam.

Sedangkan untuk download file, kita bisa pakai ResponseEntity dengan body kayak byte[], InputStream atau Resource.



Nggak berhenti sampai situ aja, logic yang harus diatur adalah gimana caranya supaya bisa mengelola lokasi file.

Biar lebih lengkap, kamu bisa akses referensi berikut ya, bestie~

[Java Spring Boot API to Upload/Download File](#)



Next, ada pembahasan tentang **Jasper Report**.

Eh ini bukan Jasper si hantu di film animasi itu,
yaa~

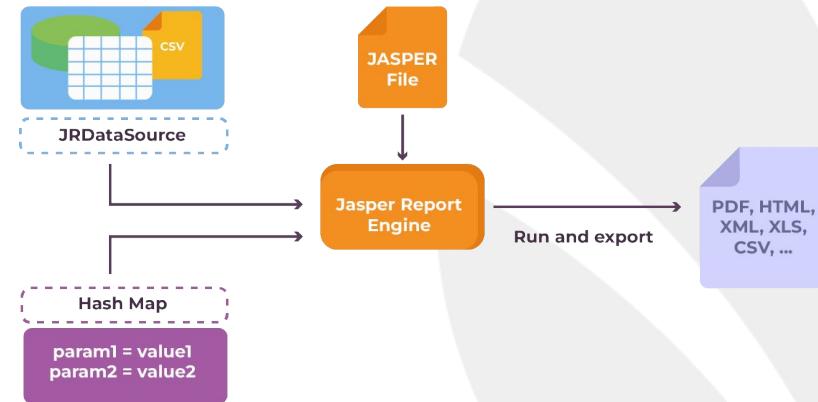
Iya, soalnya itu Casper, bukan Jasper □



Ini nih arti dari Jasper Report yang sebenarnya~

Jasper Report adalah sebuah **tools untuk membantu dalam pembuatan suatu report**.

Bisa bantu apa aja? Jasper bisa menghasilkan file kayak PDF, HTML, Microsoft Excel, RTF, ODT, comma-separated values (CSV) dan XML.

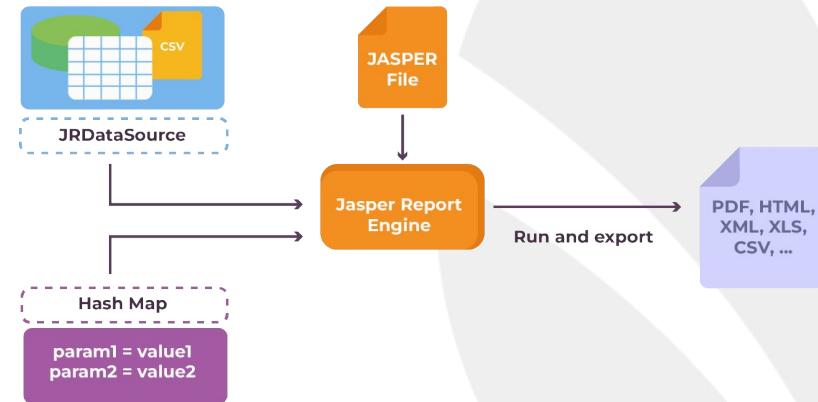


Sebuah file Jasper punya ekstensi yang beda, lho!

Proses Jasper bekerja tuh gini. Jasper membaca file XML yang terdiri dari attribute template report yang udah dibikin.

File XML ini beda dengan yang lainnya karena udah dikhususkan untuk Jasper Report. Sehingga si file tersebut punya ekstensi **.jrxml**.

Nantinya, file XML bakal di decompile supaya template tersebut siap buat di populate dengan data. File yang udah di-compile punya ekstensi **.jasper**.



Kita coba bikin report berupa file PDF pakai JaspersoftStudio, yuk!

Untuk memudahkan kita dalam mengolah jrxml dengan GUI, kita bisa pakai **iReport dengan Java 7** dan **Jaspersoft Studio dengan Java 8 ke atas**.

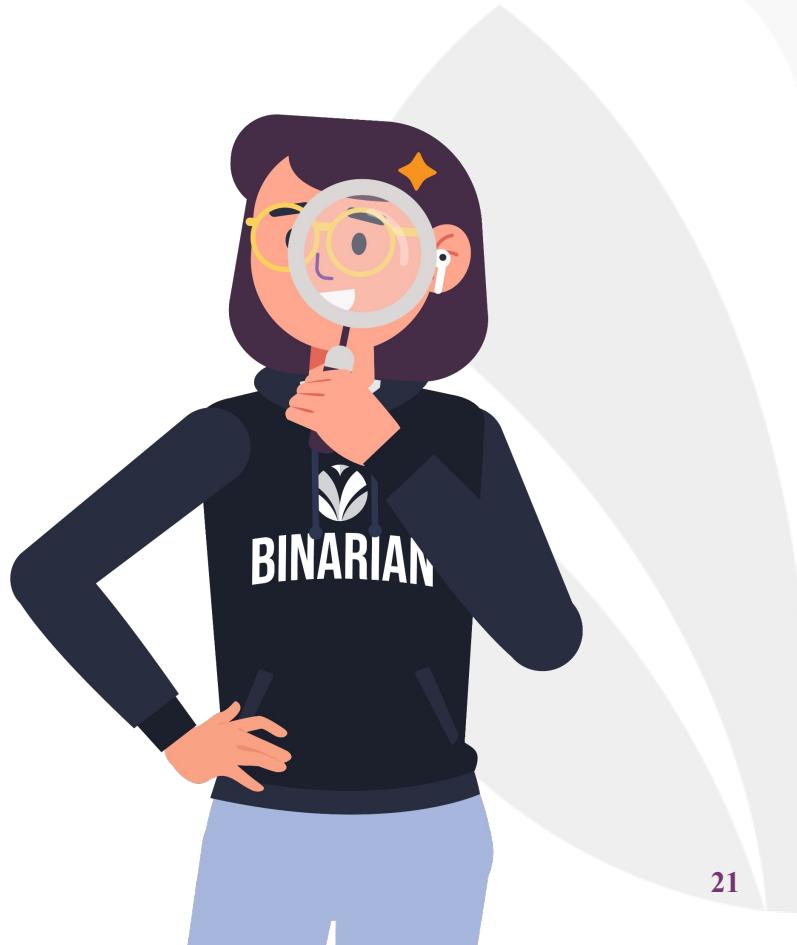
Hal ini berpengaruh pada proses kompilasi jrxml nanti.



Nah, pada course ini kita bakal bikin report berupa file PDF pakai JaspersoftStudio sebagai editor jrxml-nya.

Jasperstudio merupakan editor yang bersifat **WYSIWYG alias (What You See Is What You Get)**.

WYSIWYG merupakan suatu editor yang digunakan oleh suatu aplikasi untuk menggambarkan, bahwa konten yang sedang diedit akan terlihat sama persis dengan hasil akhirnya.



Berikut adalah link untuk download JaspersoftStudio, bestie~
[Download Jaspersoft](#)

Kita pakai versi community, ya!



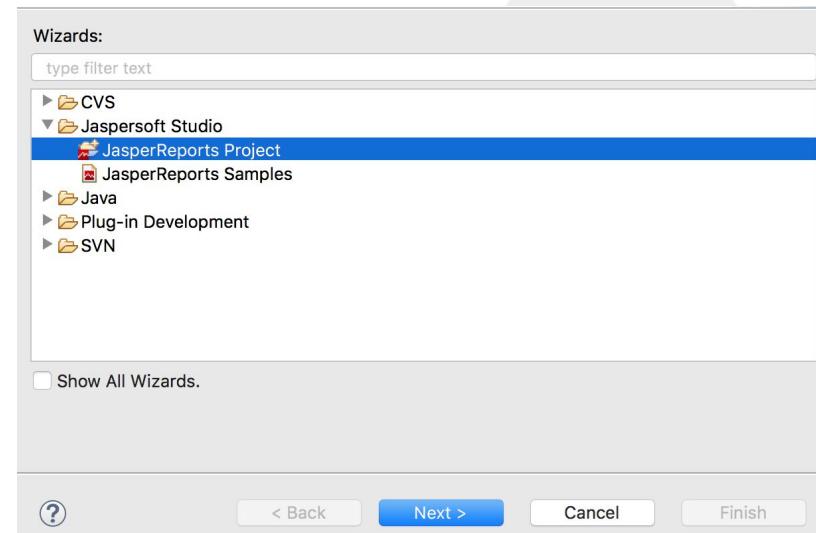
Pas udah install JaspersoftStudio, kita bakal bikin project terlebih dahulu, gengs!

Lanjut, nih! siap-siap ya karena kita akan mulai petualangan eksplorasi JaspersoftStudio. Berikut cara mulai project-nya:

1. Pilih File → New → Project

Pilih Jenis file yang mau dibikin pakai Jaspersoft Studio.

File tersebut adalah **JasperReports Project**.



2. Nantinya bakal keluar tampilan kayak gambar berikut setelah selesai bikin project baru, sob.

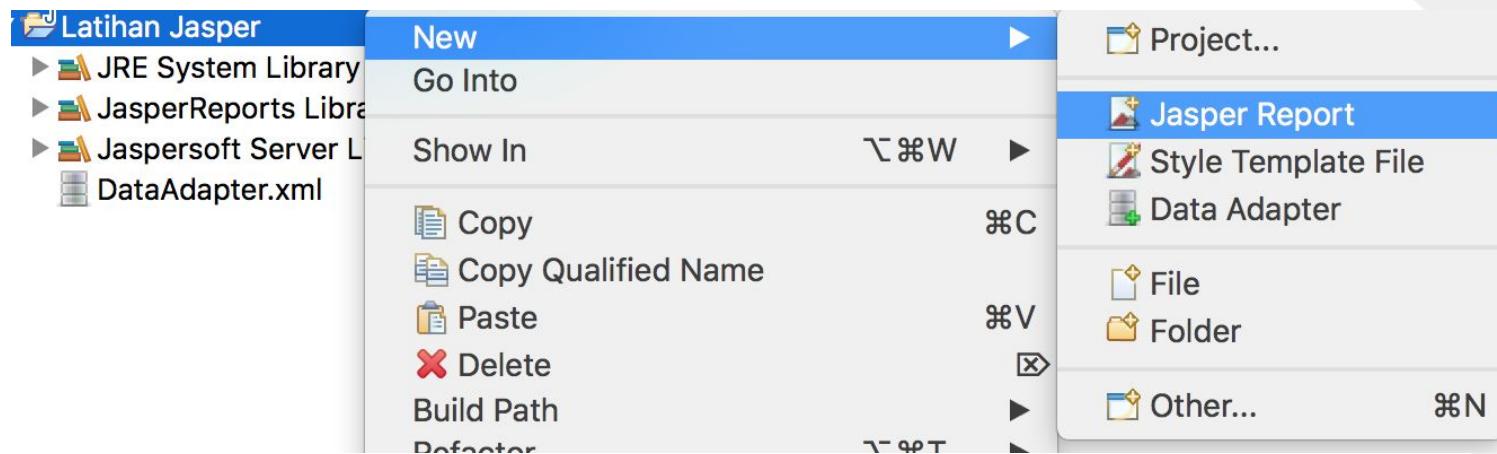
▼ Latihan Jasper

- ▶  JRE System Library [JavaSE-1.8]
- ▶  JasperReports Library
- ▶  Jaspersoft Server Library

3. Langkah selanjutnya merupakan langkah utama yang paling butuh ketelitian, yaitu pembuatan desain report.

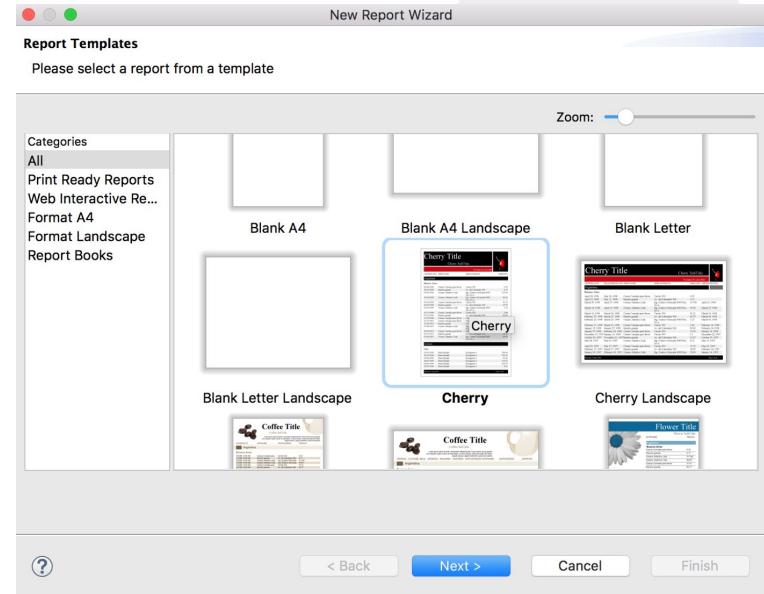
Perhatikan gambar berikut ya, sob.

project → New → Jasper Report



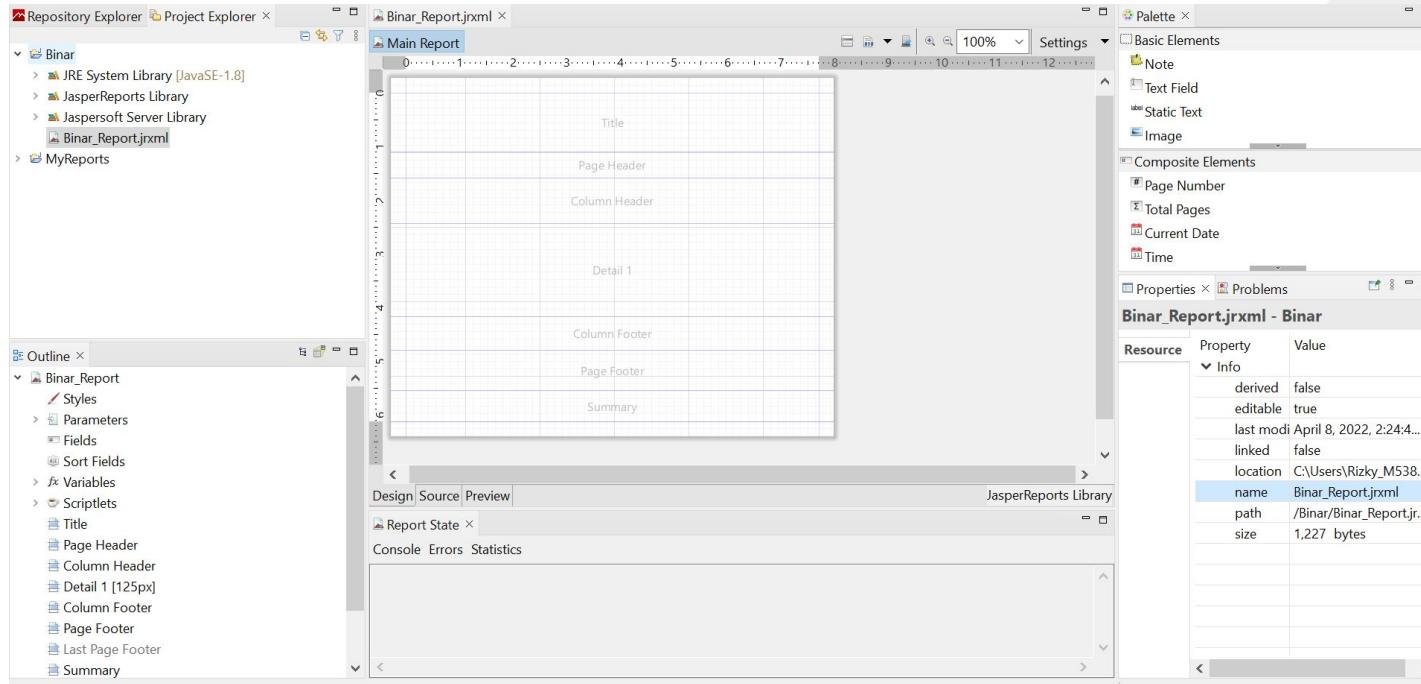
Ada catetan~ supaya memudahkan kamu ketika membuat design report, kita bisa pakai template yang udah ada.

Tapi, kalau kita mau bikin dari nol, kita bisa pilih yang blank, ya.



Working with Media and PDF

Berikut adalah contoh tampilan dari blank report!



The screenshot shows the JasperReports Studio interface with the following components:

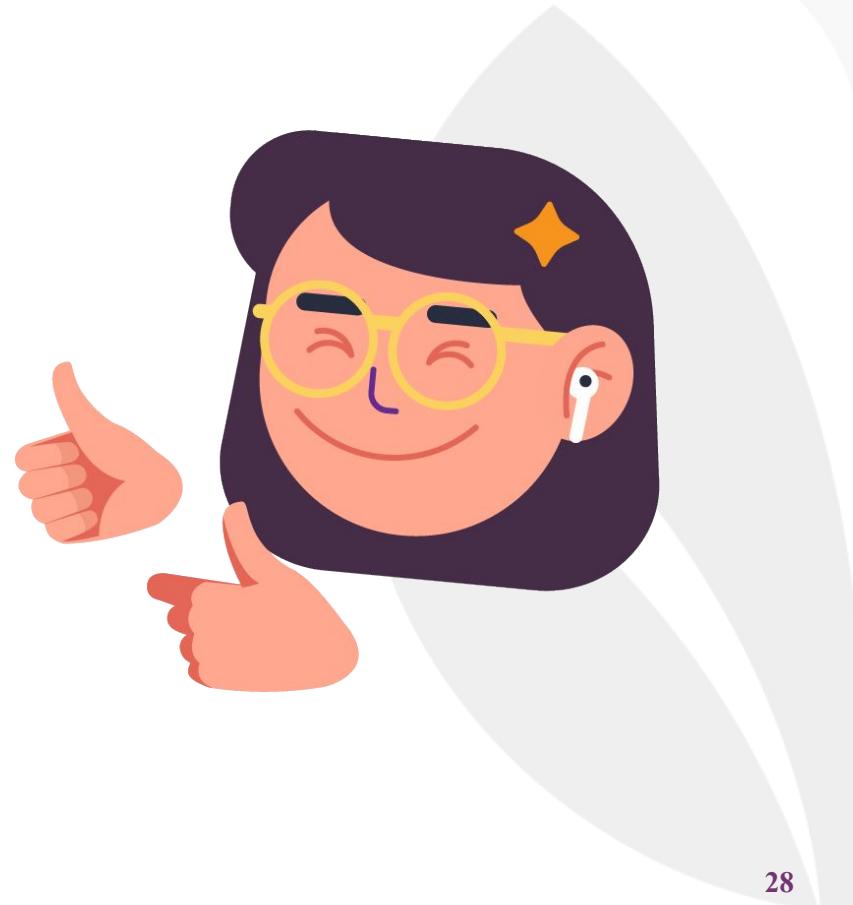
- Repository Explorer**: Shows a project structure with a folder named "Binar" containing "JRE System Library [JavaSE-1.8]", "JasperReports Library", "Jaspersoft Server Library", and the report file "Binar_Report.jrxml".
- Project Explorer**: Shows a "MyReports" folder.
- Design View**: Displays a blank report template with sections: Title, Page Header, Column Header, Detail 1, Column Footer, Page Footer, and Summary.
- Outline**: Shows the report structure with items like Binar_Report, Styles, Parameters, Fields, Sort Fields, Variables, Scriptlets, Title, Page Header, Column Header, Detail 1 [125px], Column Footer, Page Footer, Last Page Footer, and Summary.
- Palette**: Contains categories for Basic Elements (Note, Text Field, Static Text, Image) and Composite Elements (Page Number, Total Pages, Current Date, Time).
- Properties**: Shows properties for the report file "Binar_Report.jrxml" under the "Info" section, including name (Binar_Report.jrxml), path (/Binar/Binar_Report.jr...), and size (1,227 bytes).
- Report State**: Shows Console, Errors, and Statistics tabs.

Selesai deh~

Itu tadi adalah tahapan untuk membuat project melalui Jasper Report.

Berikutnya kita belajar tentang bagian-bagian dari tampilan pada editor JaspersoftStudio.

Let's go~



Sekarang, kita coba perhatikan contoh tampilan dari tab design, ya~

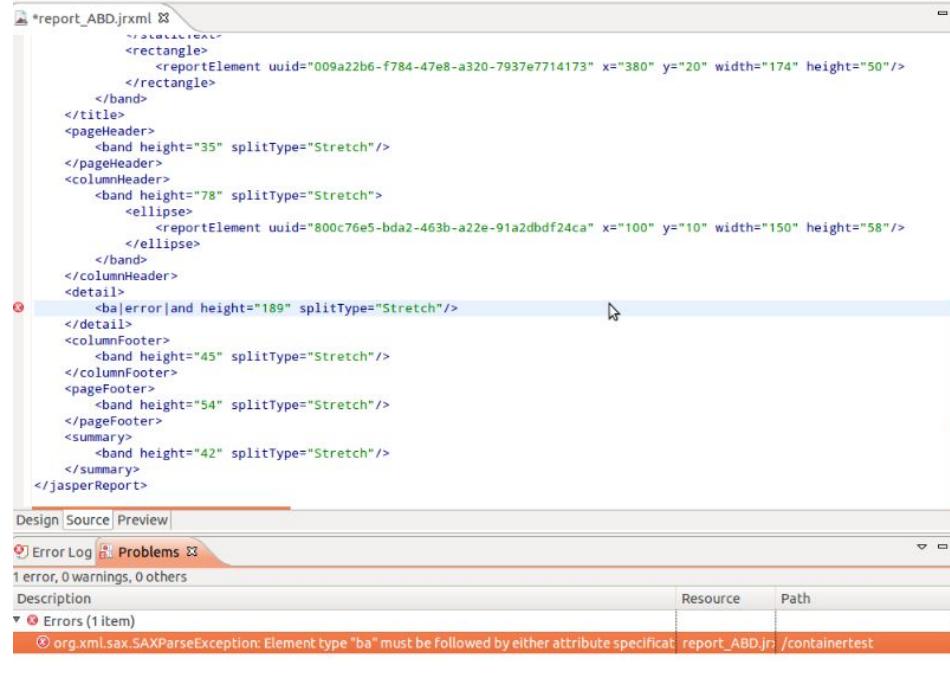
Gambar berikut disebut juga dengan **report state**.

Ada tiga tab tampilan di dalam report state nih, yaitu **Design**, **Source**, dan **Review**.



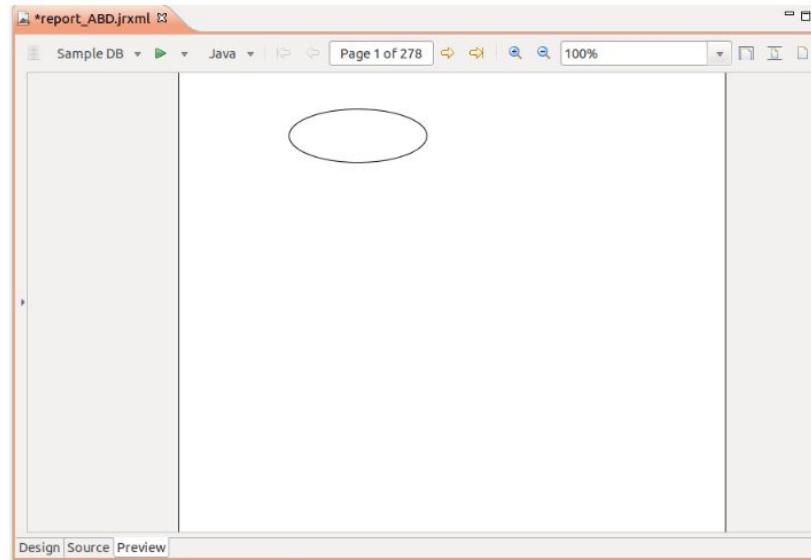
Gambar berikut adalah contoh dari **tab Source**.

Tab source bakal menampilkan source code dari file jrxml.

A screenshot of the JasperReports Studio software interface. The main window shows the XML code for a report named 'report_ABD.jrxml'. The code includes various report structures like title, pageHeader, columnHeader, detail, and summary bands. A specific line of code, '<ba|error> and height="189" splitType="Stretch"/>', is highlighted with a red rectangle. Below the code editor, there are tabs for 'Design', 'Source' (which is currently selected), and 'Preview'. At the bottom, there's an 'Error Log' tab showing one error: 'org.xml.sax.SAXParseException: Element type "ba" must be followed by either attribute specification or nothing'.

```
<rectangle>
    <reportElement uuid="009a22b6-f784-47e8-a320-7937e7714173" x="380" y="20" width="174" height="50"/>
</rectangle>
</band>
</title>
<pageHeader>
    <band height="35" splitType="Stretch"/>
</pageHeader>
<columnHeader>
    <band height="78" splitType="Stretch">
        <ellipse>
            <reportElement uuid="800c76e5-bda2-463b-a22e-91a2dbdf24ca" x="100" y="10" width="150" height="58"/>
        </ellipse>
    </band>
</columnHeader>
<detail>
    <ba|error> and height="189" splitType="Stretch"/>
</detail>
<columnFooter>
    <band height="45" splitType="Stretch"/>
</columnFooter>
<pageFooter>
    <band height="54" splitType="Stretch"/>
</pageFooter>
<summary>
    <band height="42" splitType="Stretch"/>
</summary>
</jasperReport>
```

Description	Resource	Path
Errors (1 item)		
org.xml.sax.SAXParseException: Element type "ba" must be followed by either attribute specification or nothing		

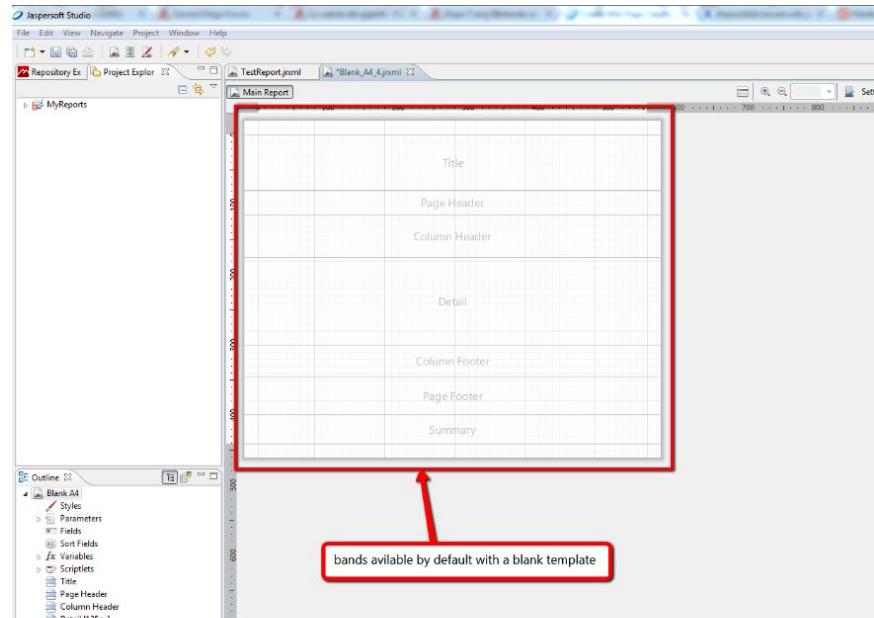


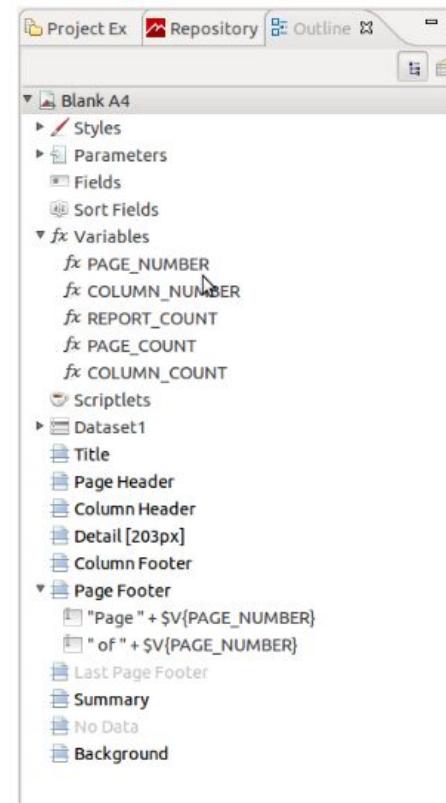
Kalau ini adalah contoh tampilan dari **tab preview**.

Kita bisa mengatur tampilan kalau dijadikan suatu report yang udah jadi kayak PDF dan Excel.

Selanjutnya adalah **Band**.

Band merupakan element-element di mana kita bisa menaruh element lainnya kayak table, text field, static text atau grafik. Berikut band yang disediakan secara default. Untuk memuat data berupa table, sebaiknya diletakkan di detail band, yaaa.



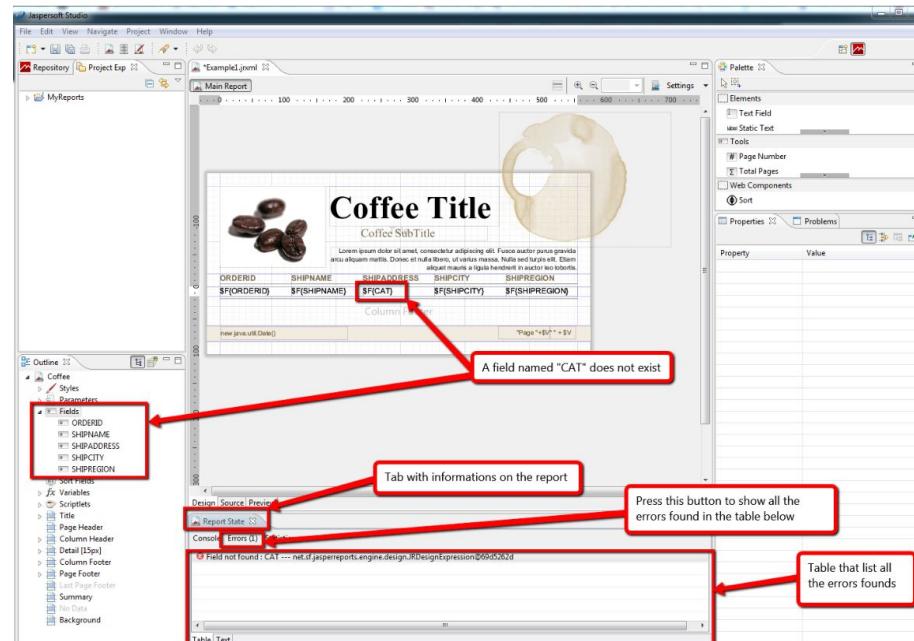


Berikut adalah layout dari **panel outline**.

Outline berisi band, style, parameter, field, sort, variable, dataset dari satu file jasper report. Dari tampilan ini kita bisa memilih salah satu element dari outline tersebut untuk diedit.

Next, ada **Field**.

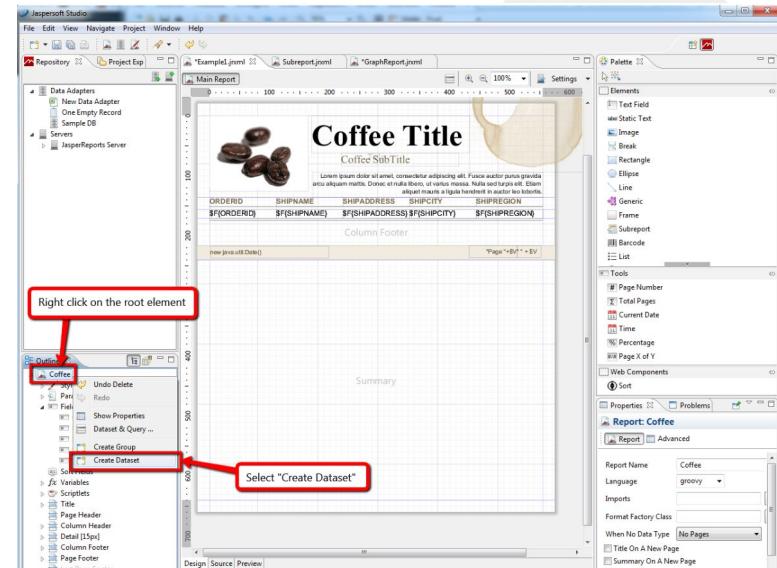
Field merupakan data yang bakal di-populate ke dalam Jasper. Element field ini bisa disesuaikan dengan datasource.



Oke! Karena kita bakal memuat data yang berasal dari aplikasi Java, kita pakai Java Bean, nih.

Untuk mendeteksi field secara langsung, kita bisa pakai **Dataset** kayak gambar di samping.

Perhatikan baik-baik, yaaa~



Berikut ini adalah cara menghubungkan class Java Bean pakai Jaspersoft Studio!

Kamu bisa coba pelajari dari referensi berikut, ya.

[Working with JavaBeans Data Adaptor](#)

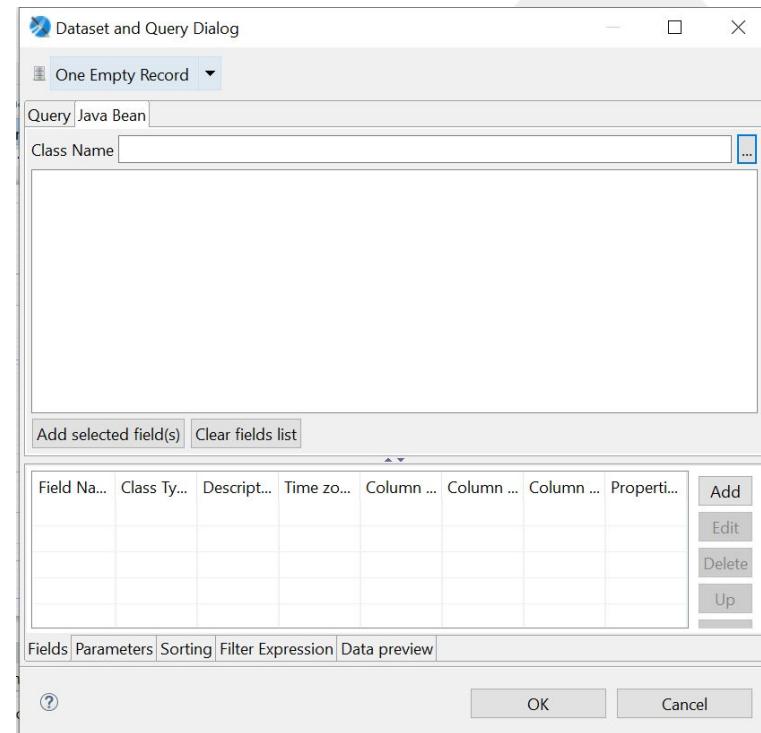
Kalau mau bikin field secara manual, kita bisa bikin field secara langsung dengan melakukan create field di element fields secara langsung, gengs.



Berikut adalah **window dari dataset and Query**.

Kita juga bisa menambahkan field secara manual dan otomatis, lho.

Canggih banget, kan?

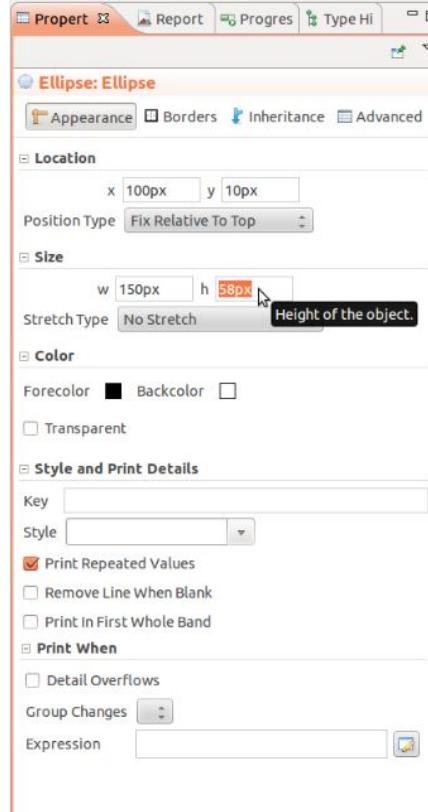


Kalau tampilan di samping adalah panel dari **properties**.

Di panel properties, akan ditampilkan properties dari element report dan report itu sendiri.

Dengan Properties kita bisa mengatur size, appearance, position, style dari sebuah element.

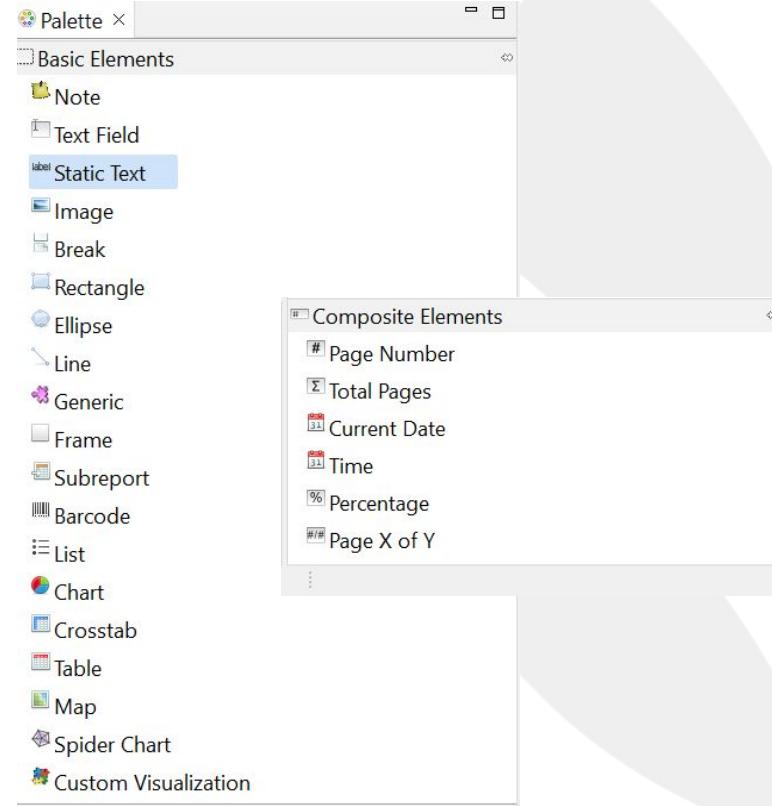
Panel ini bisa diakses dengan cara melakukan klik kanan pada element di outline terus tinggal pilih deh show properties~



Panel palette berisi element-element yang bisa ditambahkan dalam report.

Untuk bisa menambahkannya, kita cuma perlu men-drag element ke dalam report.

Referensi untuk melihat fungsi dari seluruh element yang ada di Jasper, bisa kamu cek [di sini](#), ya!

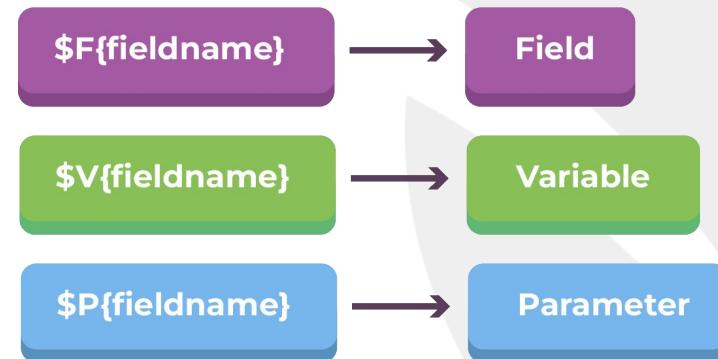


Dari keseluruhan element, element yang bakal sering digunakan adalah text~

Text ini dibagi menjadi dua, yaitu:

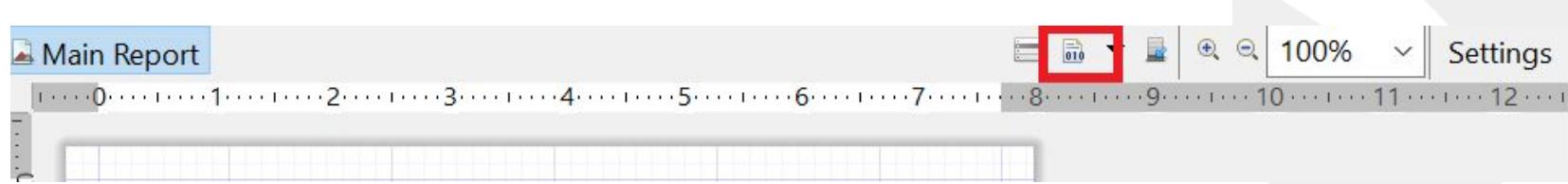
- **Static text** adalah text yang tulisannya nggak bisa diubah.
- **Text Field** merupakan element text yang bisa bersifat dinamis. Value-nya bisa berasal dari field, parameter dan variable.

Di samping adalah syntax textfield untuk mengambil value yang berasal dari field, parameter dan variable.



Untuk meng-compile file jrxml, kita bisa pakai button berikut pada report state.

Setelah klik icon ini, maka file dengan ekstensi .jasper bakal terbentuk deh. Bimsalabim!



Berikutnya kita bakal menghubungkan aplikasi Spring Boot dengan Jasper~

Untuk bisa memproses Jasper, kita harus menambahkan dependency pada pom.xml.

Coba lirik dependency di samping, yaa~



```
<dependency>
    <groupId>net.sf.jasperreports</groupId>
    <artifactId>jasperreports</artifactId>
</dependency>
```

Buat mengolah file jrxml, ada object dan method dari library Jasper yang harus dipakai nih, gengs~

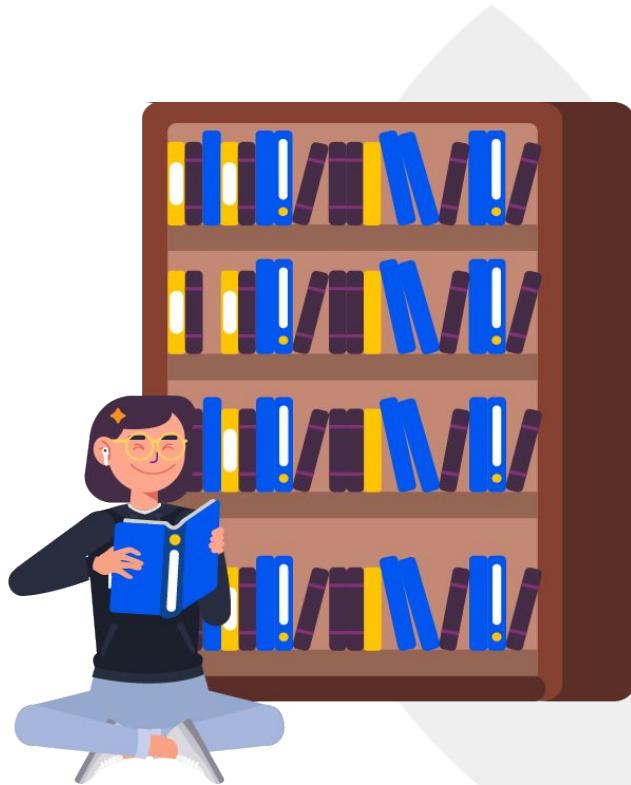
Berikut adalah object dan method-nya:

1. JRBeanCollectionDataSource

Object buat membentuk suatu data yang bakal di-populate ke dalam template Jasper.

2. JasperFillManager.fillReport

Method yang berfungsi buat mem-populate data yang berasal dari data source.



3. JasperCompileManager.compileReport

Kalau file yang disertakan sebagai resource adalah jrxml, maka report bakal decompile terlebih dahulu menggunakan method ini.

Sedangkan kalau file berekstensi Jasper, maka nggak perlu pakai method ini, tapi cuma perlu mendapatkan absolute/relative pathnya.

4. JasperExportManager

Class yang berisi method buat mengenerate file jadi kayak PDF, xlsx, dll.



```
List<User> userList = userRepository.getAllUserDetails(); //data yang akan di-populate

Map<String, Object> dataMap = new HashMap<String, Object>();
dataMap.put("userData", new JRBeanCollectionDataSource(userList));
JasperPrint userReport =
        JasperFillManager.fillReport
(
    JasperCompileManager.compileReport(
        ResourceUtils.getFile("classpath:user-details.jrxml")
            .getAbsolutePath() // path dari file jasper report
        , dataMap // dynamic parameters
        , new JREmptyDataSource()
    );
JasperExportManager.exportReportToPdf(userReport)
```

File tersebut bisa diletakkan sebagai body dari ResponseEntity supaya bisa di-download, gengs.

Berikut adalah referensi untuk mempelajari cara menghubungkan Spring Boot dan Jasper.

Silahkan disimak, bestie~

- [Create and download jasper report PDF via REST service in Spring Boot](#)
- [How to Generate Reports in a Spring Boot App Leveraging Jaspersoft](#)



Materi udah selesai, nih!

Wahh... kita udah sampai di akhir materi pada topik ini. Kalian luar biasa banget udah sampai di titik ini!

Supaya pemahaman kita makin super duper amazing, sebelum move on ke kuis, ada satu latihan buat kamu.



Latihannya gini. **Lakukan unduh dan unggah file format image.**
Kemudian, buat PDF file menggunakan Jasper report.

Isi dan formatnya PDFnya bebas terserah dari kamu. Kamu bisa gunakan database yang pernah kamu buat.

Latihan ini dilakukan di kelas dan jangan lupa untuk mendiskusikan hasil jawabannya bersama teman sekelas dan facilitator, ya. Selamat mencoba~



Sipp deh, kita udah sama-sama belajar tentang cara download dan upload file berupa foto dan video.

Bianarian, kalau menurut kamu antara foto dan video, mana sih yang paling challenging untuk dipelajari?

Sabrina kepo nih sama jawaban kamu □



Nah, selesai sudah pembahasan kita di **Chapter 5** ini. Luar biasa! ☺

Selanjutnya, kita bakal siap-siap untuk move on ke chapter baru yaitu:

✨**Chapter 6** ✨

Sampai jumpa pada chapter selanjutnya~

