



Gradle

Eko Kurniawan Khannedy



License

- Dokumen ini boleh Anda gunakan atau ubah untuk keperluan non komersial
- Tapi Anda wajib mencantumkan sumber dan pemilik dokumen ini
- Untuk keperluan komersial, silahkan hubungi pemilik dokumen ini

Eko Kurniawan Khannedy

- Technical architect at one of the biggest ecommerce company in Indonesia
- 10+ years experiences
- youtube.com/c/ProgrammerZamanNow



Pengenalan Gradle



Pengenalan Build Automation

- Build Automation adalah proses meng-otomatisasi tahapan pembuatan software dan hal-hal yang berhubungan dengannya, seperti: kompilasi source code menjadi binary code, mem-package binary code menjadi distribution file, membuat dokumentasi, menjalankan automated test sampai manajemen dependency.

Kompilasi Source Code



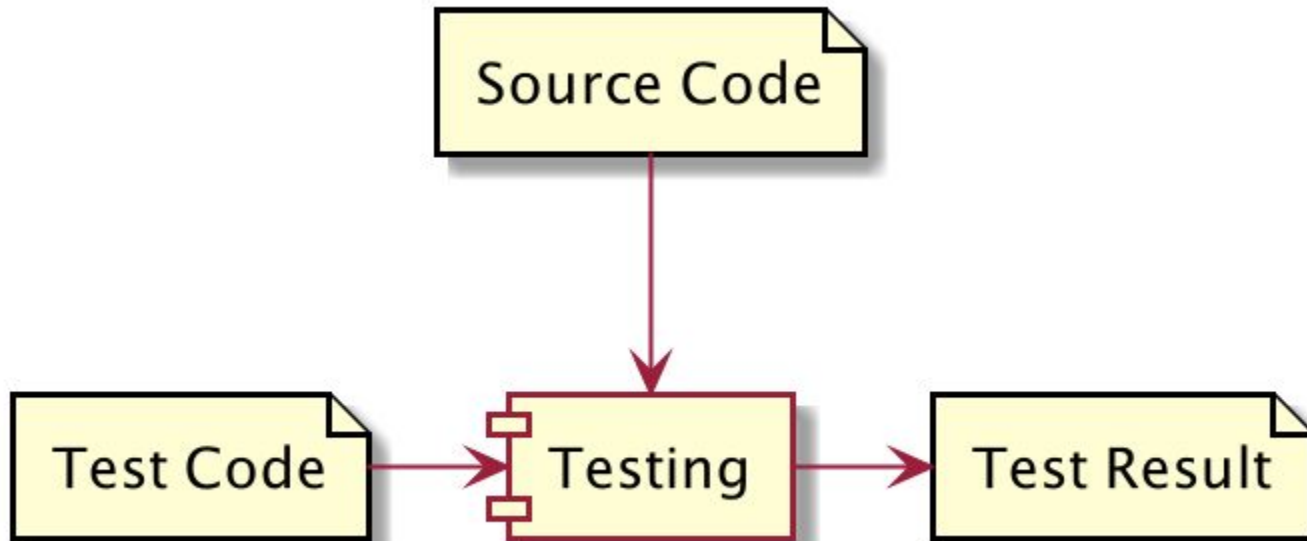
Mem-Package Binary File



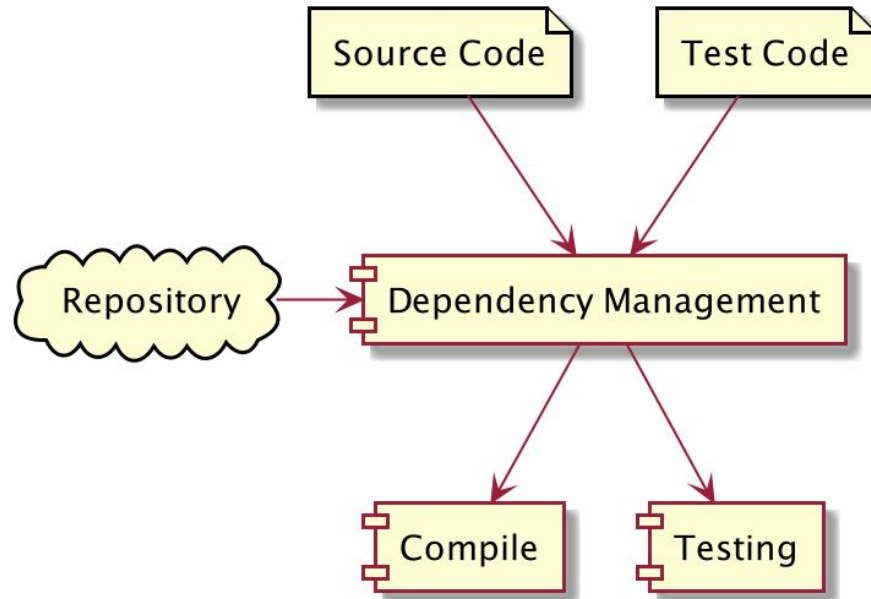
Membuat Dokumentasi



Menjalankan Automated Test



Management Dependency





Contoh Build Automation Tool

- Gradle
- Apache Maven
- Apache Ivy



Pengenalan Gradle

- Gradle adalah salah satu build automation yang free dan open source
- Gradle menggunakan bahasa pemrograman Groovy atau Kotlin untuk mendefinisikan build script nya
- Gradle saat ini sangat populer di kalangan Developer Android
- Gradle menggunakan JVM sebagai fondasi dasar
- <https://gradle.org/>



Teknologi yang Didukung

Gradle mendukung build automation untuk banyak teknologi, seperti :

- JVM (Java, Groovy, Kotlin, Scala)
- Android
- C++
- Swift

Menginstall Gradle



Download Gradle

- <https://gradle.org/releases/>



Setting Path

- Windows : <https://medium.com/programmer-zaman-now/setting-java-path-di-windows-4da2c65d8298>
- Mac & Linux

```
# Add to .bashrc or .zshrc
```

```
export GRADLE_HOME="/Users/khannedy/Tools/gradle-6.5"  
export PATH="$GRADLE_HOME/bin:$PATH"
```




Mengecek Gradle

```
→ ~ gradle --version
```

```
-----  
Gradle 6.5  
-----
```

```
Build time: 2020-06-02 20:46:21 UTC
```

```
Revision: a27f41e4ae5e8a41ab9b19f8dd6d86d7b384dad4
```

```
Kotlin: 1.3.72
```

```
Groovy: 2.5.11
```

```
Ant: Apache Ant(TM) version 1.10.7 compiled on September 1 2019
```

```
JVM: 14 (Oracle Corporation 14+36-1461)
```

```
OS: Mac OS X 10.15.3 x86_64
```

Membuat Project



Membuat Project dengan Gradle

- Gradle mendukung banyak jenis tipe project, seperti Java, Kotlin, Scala, Groovy dan lain-lain
- Saat ini Gradle banyak digunakan oleh programmer Kotlin, terutama developer Android

Membuat Kotlin Project

- Buat folder project, lalu masuk ke folder tersebut lewat terminal, dan ketik :
 - gradle init

```
drwxr-xr-x 15 khannedy staff 416B May 20
→ BELAJAR mkdir belajar-gradle
→ BELAJAR cd belajar-gradle
→ belajar-gradle gradle init

Select type of project to generate:
 1: basic
 2: application
 3: library
 4: Gradle plugin
Enter selection (default: basic) [1..4] 2

Select implementation language:
 1: C++
 2: Groovy
 3: Java
 4: Kotlin
 5: Swift
Enter selection (default: Java) [1..5] 4

Select build script DSL:
 1: Groovy
 2: Kotlin
Enter selection (default: Kotlin) [1..2] 2

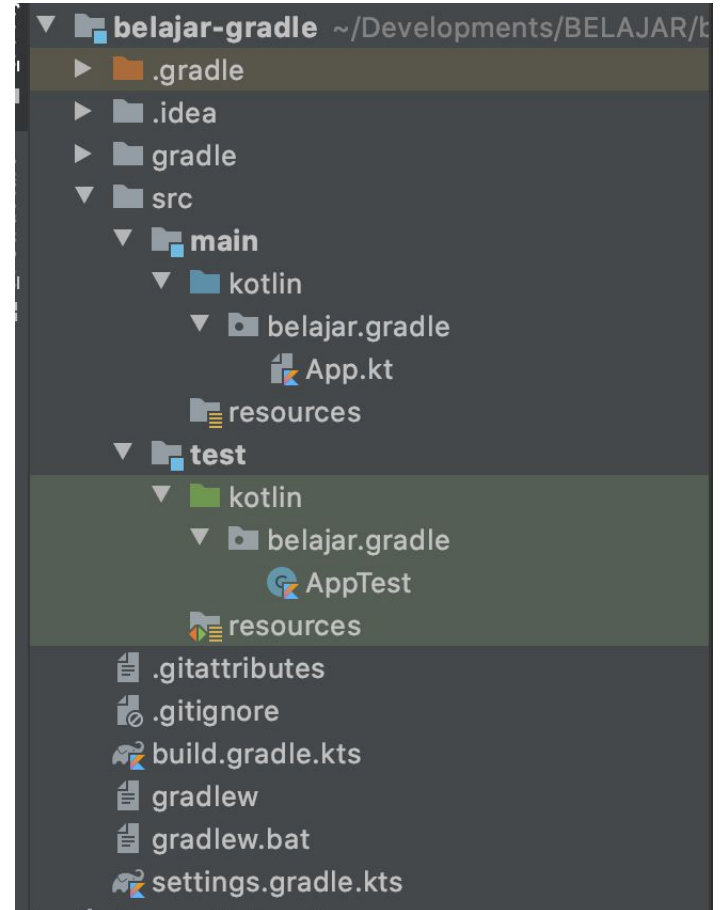
Project name (default: belajar-gradle):
Source package (default: belajar.gradle):

BUILD SUCCESSFUL in 28s
2 actionable tasks: 2 executed
→ belajar-gradle
```

Struktur Project



Struktur Project Gradle



Gradle Task



Task

- Perintah-perintah di gradle itu berbasis tasks
- Kita bisa menjalankan task di gradle dengan menggunakan perintah : gradle namatask
- Jika kita ingin melihat daftar task yang tersedia di gradle, kita bisa menggunakan perintah : gradle tasks --all



Membuat Tasks

- Gradle menggunakan kotlin sebagai DSL
- Oleh karena itu, kita juga bisa membuat task menggunakan bahasa kotlin
- Kita cukup menambahkan task yang kita inginkan di file build.gradle.kts



Kode : Membuat Kotlin

```
5 }  
6  
7 tasks.register("helloworld") {  
8     doFirst {  
9         println("Hello")  
10    }  
11    doLast {  
12        println("World")  
13    }  
14 }
```

Gradle Properties

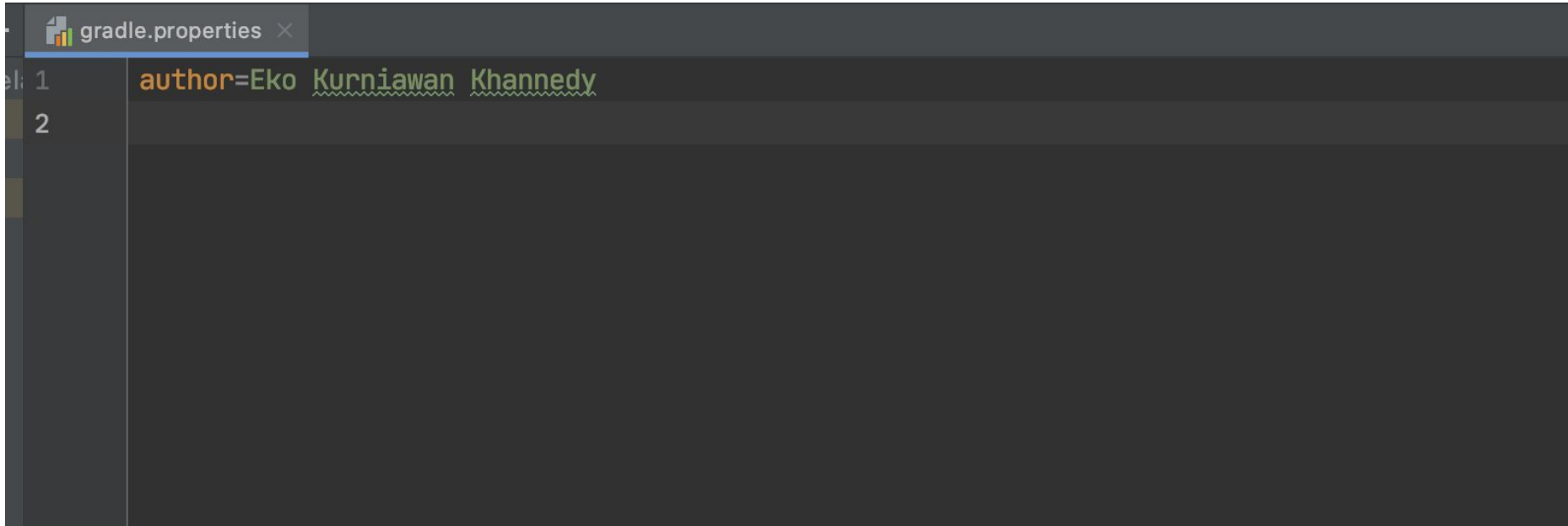


Gradle Properties

- Gradle mendukung konfigurasi menggunakan file `gradle.properties`
- Secara otomatis kita bisa mengambil data konfigurasi dari `gradle.properties`



Kode : Gradle Properties



```
gradle.properties x
1 author=Eko Kurniawan Khannedy
2
```



Kode : Menggunakan Gradle Properties

```
39
40 tasks.register("author") {
41     val author: String by project
42     doFirst {
43         println(author)
44     }
45 }
46
47 tasks.register("helloworld") {
48     doFirst {
49         println("Hello")
50     }
}
```

Build Project



Build Project

- Saat kita membuat project biasanya akan ada 2 jenis kode yang kita buat, kode program nya, dan kode testing nya
- Gradle mendukung kedua hal ini



Menjalankan Kompilasi Program

gradle classes



Menjalankan Unit Test

gradle test

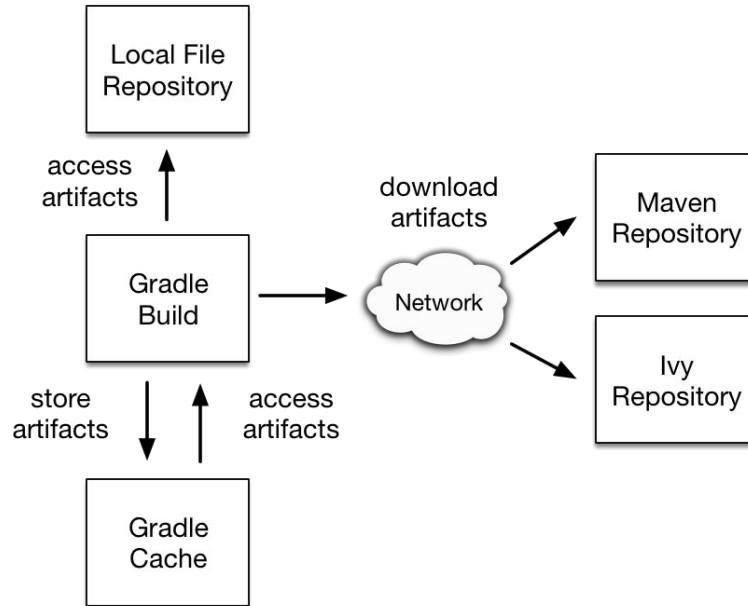
Dependency Management



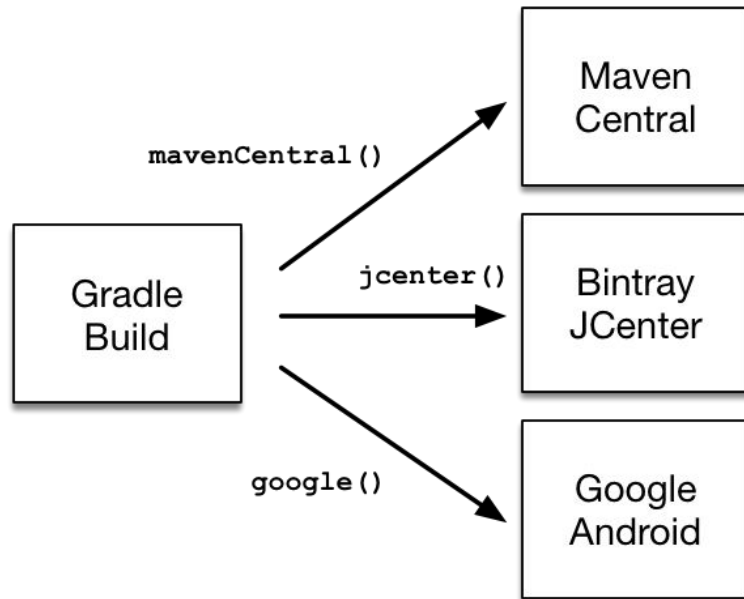
Dependency Management

- Proyek aplikasi jarang sekali berdiri sendiri, biasanya membutuhkan dukungan dari pihak lain, seperti tool atau library
- Tanpa build tool seperti Gradle, untuk menambahkan library dari luar, kita harus melakukannya secara manual
- Gradle mendukung dependency management, dimana kita tidak perlu me-manage secara manual proses menambahkan dependency (tool atau library) ke dalam proyek aplikasi kita

Dependency Management



Repository





Kode : Menambah Repository

```
14
15 repositories {
16     // Use jcenter for resolving dependencies.
17     // You can declare any Maven/Ivy/file repository here.
18     jcenter()
19     mavenCentral()
20     google()
21     maven {
22         url = uri("https://dl.bintray.com/blibliidotcom/maven")
23     }
24 }
25
```



Dependency Configuration

Secara garis besar ada 2 jenis configuration untuk dependency di Gradle (sebenarnya masih banyak, cuma yang biasa digunakan ada 2)

- `implementation`, adalah dependency yang digunakan oleh kode program kita
- `testImplementation`, adalah dependency yang digunakan oleh kode unit test kita



Kode : Menambah Dependency

```
dependencies {  
    // Align versions of all Kotlin components  
    implementation(platform("org.jetbrains.kotlin:kotlin-bom"))  
    // Use the Kotlin JDK 8 standard library.  
    implementation("org.jetbrains.kotlin:kotlin-stdlib-jdk8")  
    // Use the Kotlin test library.  
    testImplementation("org.jetbrains.kotlin:kotlin-test")  
    // Use the Kotlin JUnit integration.  
    testImplementation("org.jetbrains.kotlin:kotlin-test-junit")  
    // Add coroutine  
    implementation("org.jetbrains.kotlinx:kotlinx-coroutines-core:1.3.7")  
}
```



Mencari Dependency

- <https://search.maven.org/>
- <https://mvnrepository.com/>

Membuat Distribution File



Membuat Distribution File

- Hal terakhir yang akan kita lakukan setelah aplikasi kita selesai adalah membuat distribution file
- Distribution file ini yang nanti akan kita distribusikan untuk di install di komputer client atau server misalnya



Membuat Distribution File

`gradle distTar`

`gradle distZip`

- Hasil distribution file bisa kita lihat di direktori build/distributions

Multi Module Project



Multi Module Project

- Saat aplikasi kita sudah sangat besar, kadang ada baiknya kita buat aplikasi dalam bentuk modular
- Misal kita pisahkan module model, controller, view, service, repository, dan lain-lain
- Untungnya, Gradle mendukung pembuatan project multi module



Membuat Module Baru

- Untuk membuat module baru, di dalam project yang sudah ada, kita hanya tinggal membuat folder baru, lalu buat gradle project baru menggunakan perintah gradle init



Menambahkan Modul ke Project Utama

```
settings.gradle.kts (belajar-gradle) ×  
1 rootProject.name = "belajar-gradle"  
2  
3 include("belajar-gradle-library")
```



Menambahkan Modul ke Dependency

```
31 // Use the Kotlin test library.
32 testImplementation("org.jetbrains.kotlin:kotlin-test")
33 // Use the Kotlin JUnit integration.
34 testImplementation("org.jetbrains.kotlin:kotlin-test-junit")
35 // Add coroutine
36 implementation("org.jetbrains.kotlinx:kotlinx-coroutines-core:1.3.7")
37
38 // add subprojects
39 implementation(project(":belajar-gradle-library"))
40 }
41
42 application {
```



Menjalankan Task untuk Modul

Untuk menjalankan task spesifik untuk module tertentu, kita bisa menambahkan nama module diikuti dengan :namatasks, misal :

```
gradle namamodule:nama-task
```

Materi Selanjutnya



Materi Selanjutnya

Yang bisa Anda lakukan :

- Mulai Membuat Aplikasi Java / Kotlin

Materi Selanjutnya :

- Kotlin Unit Test



Eko Kurniawan Khannedy

- Telegram : @khannedy
- Facebook : fb.com/ProgrammerZamanNow
- Instagram : instagram.com/programmerzamannow
- Youtube : youtube.com/c/ProgrammerZamanNow
- Telegram Channel : <https://t.me/ProgrammerZamanNow>
- Email : echo.khannedy@gmail.com