JAVA UNIT TEST

## **Pendahuluan**

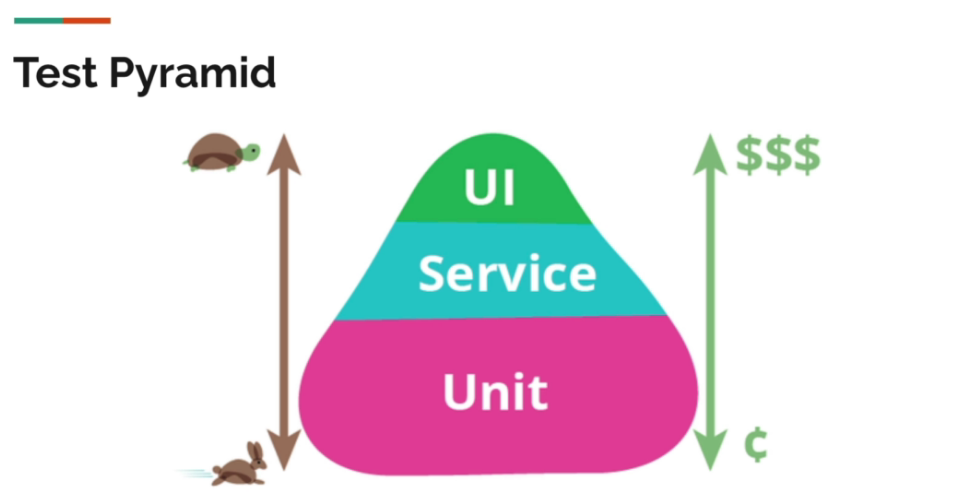
## ## Sebelum Belajar Materi ini ##

## sebaiknya mempelajari Java Dasar, Java Object Oriented Programming, Java Generic, Java Collection, Java Lambda, Apache Maven

## ## Agenda ##

Pengenalan Software Testing, Pengenalan JUnit, Membuat Test, Menggunakan Assertions, Menggunakan Assumptions, Dependency Injection di Test, Mocking Dan lain-lain

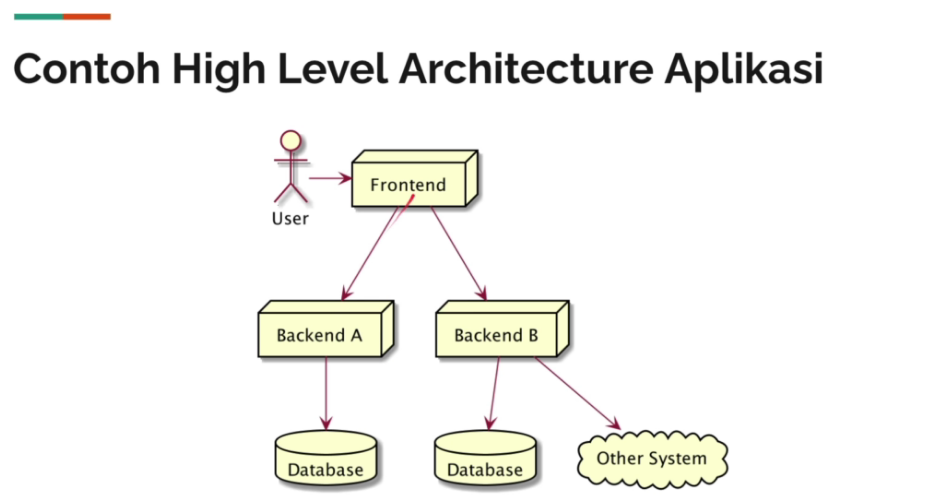
## **Pengenalan Software Testing**

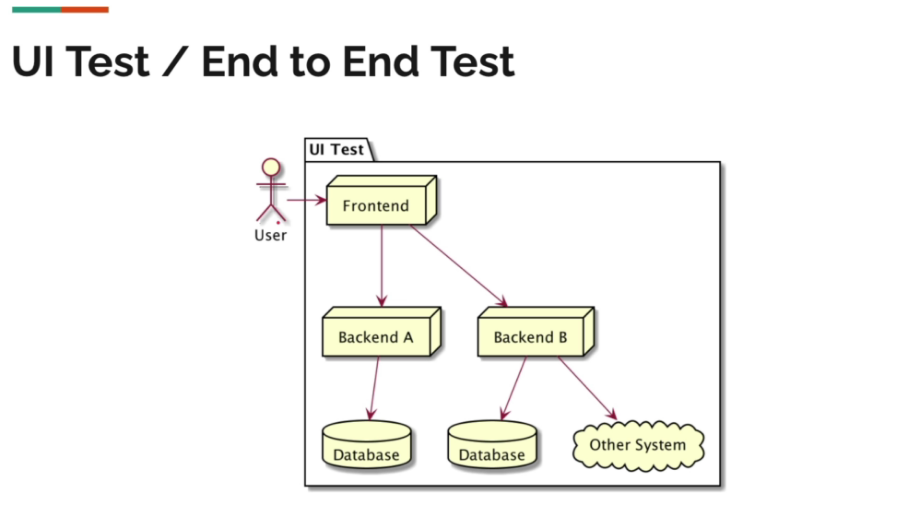


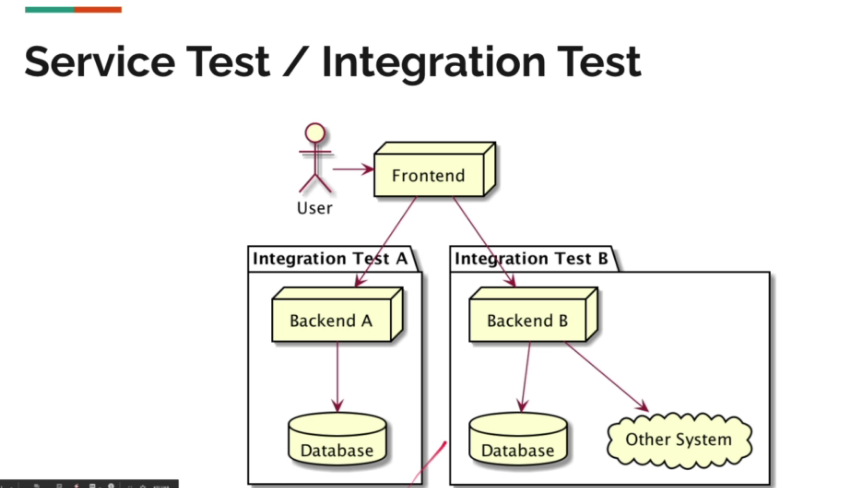
Unit test : testing yang harusnya dibuta sebanyak-banyaknya , yang cepat dan murah

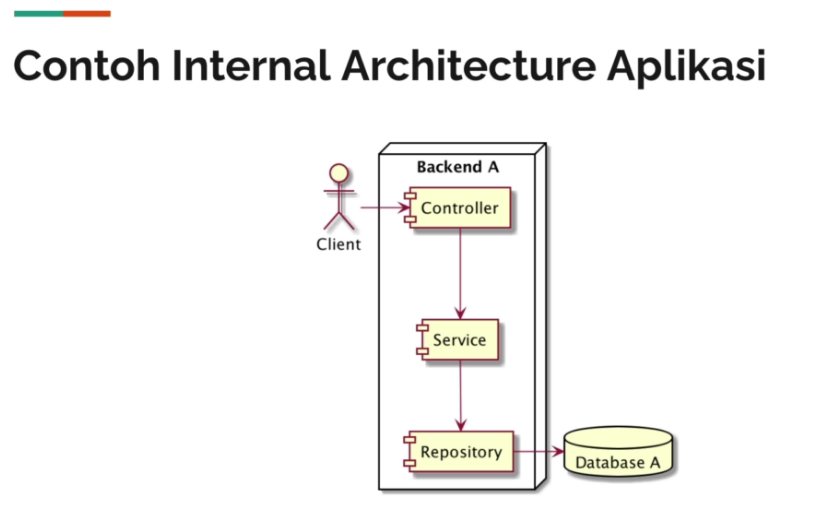
Service : testing pertengahan tidak murah banget, tidak lambat banget juga

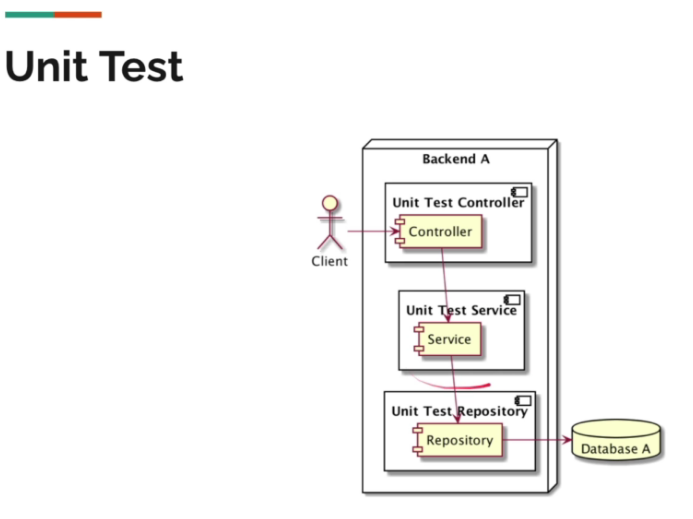
UI : paling lambat pengerjaan, paling mahal











## ## Unit Test ##

## Unit test akan focus menguji bagian kode program terkecil, biasanya menguji sebuah method

* Unit test biasanya dibuat dan cepat, oleh karena itu biasanya kadang kode unit test lebih bnyak dari kode program aslinya, karena semua scenario pengujian akan dicoba di unit test
* Unit test bisa digunakan sebagai cara untuk meningkatkan kualitas kode program kita

## **Pengenalan Junit**

## Junit ##

* Junit adalah test framework yang paling popular di Java
* Saat ini versi terbaru Junit adalah versi 5
* Junit 5 membutuhkan Java minimal versi 8
* https://junit.org/

## Creatre Maven ##

* mvn archetype:generate
* maven-archetype-quickstart
* Pilih *org.maven.archetypes-maven-archetypes-quickstart*
* Pilih version
* Groupid, biasanya nama atau url organisasi atau perusahaan
* artifacId, nama project bisanya hurus kecil semua dan dipisah dengan tanda “-”
* version yaitu terkait versi project
* package yaitu nama package project , biasanya dipihasahkan dengan titik penamaannya

## **Membuat Test**

* Untuk membuat test di Junit itu sederhana, kita cukup membuat class, lalu menambahkan method-method test nya
* Method akan sianggap sebuah test jika ditambahkan annotation @Test
* Kode test disimpan dibagian test folder di maven, bukan di main folder
* Biasanya saat membuat class untuk test, rata- rata orang biasa membuat nama class nya sama dengan nama class yang akan di test, tapi diakhiri dengan kata Test, missal jika nama class nya adalah Calculator, maka nama class test nya adalah Calculator Test

## Menggagalkan Test ##

* Kadang dalam membuat unit test, kita tidak hanya ingin mengetest kasus sukses atau gagal
* Ada kalanya kita ingin mengetes sebuah exception misalnya
* Assertion juga bisa digunakan untuk mengecek apakah sebuah exception terjadi