**Rangkuman Unit Testing**

Kita sudah berada di penghujung materi Unit Testing. Sampai sejauh ini, Anda diharapkan telah paham terkait tes unit menggunakan library unittest pada Python. Sekarang, mari kita rangkum secara saksama.

**Pengantar Unit Testing**

Saat membangun aplikasi atau program yang lebih kompleks, aplikasi tersebut akan memunculkan dependensi, yaitu satu atau lebih fungsi digunakan oleh fungsi lain. Bahkan, ketika kita mulai membangun aplikasi dengan rekan kita, kita membuat fungsi yang digunakan oleh rekan ataupun sebaliknya. Dependensi tersebut tentu perlu dipastikan bahwa fungsionalitasnya dapat berjalan dengan baik dan tidak terganggu dengan adanya perubahan atau fungsi baru yang dibuat.

Di sinilah kita butuh pengujian (tes) untuk fungsi-fungsi tersebut. Pengujian sebenarnya dapat dibedakan menjadi dua tipe, yaitu **manual** dan **otomatis**.

1. Manual testing adalah proses pengujian yang dilakukan oleh seseorang yang ditunjuk sebagai tester (penguji).
2. Testing otomatis merupakan hal yang sebaliknya. Ini adalah pengujian yang dilakukan secara otomatis terhadap kode-kode yang kita bangun berdasarkan rencana pengujian (test plan).

Tidak hanya sekadar manual dan otomatis, ada juga **unit testing** dan **integration testing**.

1. Integration testing adalah pengujian yang bertujuan untuk menguji suatu sistem sebagai satu kesatuan.
2. Unit testing adalah pengujian yang lebih spesifik dan fokus terhadap bagian-bagian kecil.

Dalam Python, untuk melakukan unit testing dapat menggunakan library **unittest**. Layaknya sebuah framework pengujian, unittest mendukung beberapa hal esensial berikut.

* Pengujian secara otomatis.
* Kode awal proses (setup) dan akhir proses (shutdown) yang dapat digunakan ulang.
* Penyatuan sejumlah pengujian dalam sebuah koleksi.
* Terpisahnya framework pengujian dari framework pelaporan (*reporting*).

Library unittest mendukung sejumlah konsep penting yang berorientasi objek, antara lain berikut.

* Test fixture merepresentasikan **persiapan**yang dibutuhkan untuk melakukan satu pengujian atau lebih serta proses pembersihannya (cleanup).
* Test case adalah sebuah unit dari pengujian ketika ia mengecek sejumlah respons dari sebagian kelompok masukan. unittest menyediakan basis class dan TestCase yang akan digunakan untuk membuat kasus pengujian baru.
* Test suite adalah sebuah koleksi dari kasus-kasus pengujian, koleksi dari test suite itu sendiri, atau gabungan keduanya.
* Test runner adalah komponen yang akan mengatur (*orchestrates*) eksekusi dari pengujian-pengujian dan menyediakan keluaran untuk pengguna.

**Penerapan Unit Test dengan Library unittest**

Unit melakukan unit testing, Anda bisa menggunakan library bawaan dari Python, yaitu **unittest**.

1. import unittest

Berikut adalah salah satu penerapan unit testing dengan menggunakan library unittest.

1. import unittest
3. def koneksi\_ke\_db():
4. print("[terhubung ke db]")
6. def putus\_koneksi\_db(db):
7. print("[tidak terhubung ke db {}]".format(db))
9. class User:
10. username = ""
11. aktif = False
13. def \_\_init\_\_(self, db, username):  # using db sample
14. self.username = username
16. def set\_aktif(self):