Nama Anggota Kelompok:

1. Maulana Iskandar Syah (11180910000051)
2. Muhammad Samiaji (11180910000060)
3. Fauzy Ramadhan Sukasno (11180910000075)

**JUnit**

WHY Permasalahan utama yang sering muncul pada pengujian dengan metode konvensional adalah tidak efisiennya penggunaan waktu. Menuliskan statement-statement debug ke dalam kode adalah metode yang kurang efektif. Metode ini mengharuskan developer untuk mengamati output program dengan cermat setiap kali program dijalankan untuk memastikan program berjalan dengan benar. Cara mempercepat waktu testing dari permasalah diatas adalah dengan meninggalkan Manual Testing dan beralih menggunakan Automated Testing. Salah satu Automated Testing adalah JUnit pada bahasa Java.

WHAT JUnit adalah sebuah testing framework Java yang bersifat open source yang digunakan untuk menulis dan menjalankan software testing secara berulang-ulang. JUnit mengikuti arsitektur xUnit dalam membuat framework unit testing-nya. JUnit adalah Automated Unit Testing yang sangat populer pada bahasa Java.

WHO JUnit dikembangkan oleh Erich Gamma dan Kent Beck.

WHERE JUnit dapat diperoleh dengan mendownload dari situs <http://junit.sourceforge.net/>.

Saat ini Versi terbaru JUnit adalah versi 5. JUnit 5 memerlukan minimal Java versi 8. Berikut adalah Annotation yang ada pada JUnit.

WHEN Unit testing dilakukan saat masa pengembangan (menuliskan kode program) dari sebuah perangkat lunak oleh si pengembang. Seperti namanya, komponen yang diuji pada unit testing adalah unit-unit dari perangkat lunak. Unit disini dapat berupa function, method, procedure, module, maupun object.

HOW Prinsip utama JUnit adalah buat test class untuk setiap class yang dibuat. Dan uji semua method yang ada pada class tersebut. JUnit juga sudah built in di beberapa IDE seperti Blue J dan Eclipse. Apabila menggunakan netbeans, memang harus mendownloadnya jar nya dulu namun setelah itu netbeans bisa me-generate test class sesuai dengan class yang kita buat. Dalam unit testing, terdapat beberapa aturan penamaan yang harus diikuti. Sebuah test class harus memiliki nama yang sama dengan class yang akan di-test dan diberiprefix “test”. JUnit menggunakan anotasi untuk menandai metode sebagai test methods dan mengkonfigurasinya.

|  |  |
| --- | --- |
| **Anotasi** | **Deskripsi** |
| @Test | Mengidentifikasi metode sebagai Test Method |
| @Disabled("reason") | Menonaktifkan Test Method dengan opsi “alasan”. |
| @BeforeEach | Dieksekusi sebelum setiap test. Digunakan untuk mempersiapkan Test Environment misalnya, menginisialisasi bidang di test class, mengonfigurasi lingkungan, dll. |
| @AfterEach | Dieksekusi setelah setiap Test. Digunakan untul mebersihkan Test Environment seperti delete temporary data, restore default. |
| @DisplayName("<Name>") | Digunakan untuk menamai Test yang akan dilakukan dengan memasukan nama yang diinginkan pada <Name>. |
| @RepeatedTest(<Number>) | Mirip dengan @Test tetapi mengulangi pengujian <Number> kali |
| @BeforeAll | Menganotasi metode yang dijalankan satu kali, sebelum memulai semua Test. Digunakan untuk melakukan aktivitas intensif waktu, misalnya, untuk menyambungkan ke database. Metode yang ditandai dengan anotasi ini perlu didefinisikan sebagai statis supaya dapat bekerja dengan JUnit. |
| @AfterAll | Menganotasi metode yang dijalankan satu kali, setelah semua Test selesai. Diigunakan untuk melakukan aktivitas pembersihan, misalnya, untuk memutuskan koneksi dari database. Metode yang dianotasi dengan anotasi ini perlu didefinisikan sebagai statis supaya dapat berjalan di JUnit. |
| @TestFactory | Menganotasi metode yang merupakan Pabrik (Factory) untuk membuat pengujian dinamis. |
| @Nested | Memungkinkan Anda menyarangkan (nest) kelas pengujian dalam untuk memaksakan urutan eksekusi tertentu. |
| @Tag("<TagName>") | Memberi tag Test Method, Testing di JUnit 5 dapat difilter berdasarkan tag. Misalnya, run only tests tagged with "fast". |
| @ExtendWith | Memungkinkan Anda mendaftarkan Extension class yang menambahkan fungsionalitas ke pengujian |

Macam-macam method Assertion:

1. assertEquals : cek apakah 2 object sama.\
2. assertTrue : cek suatu kondisi benar
3. assertFalse : cek suatu kondisi salah
4. assertNull : cek apakah object null
5. assertNotNull : cek apakah object tidak null.

**Contoh JUnit Pada John Travolta**

