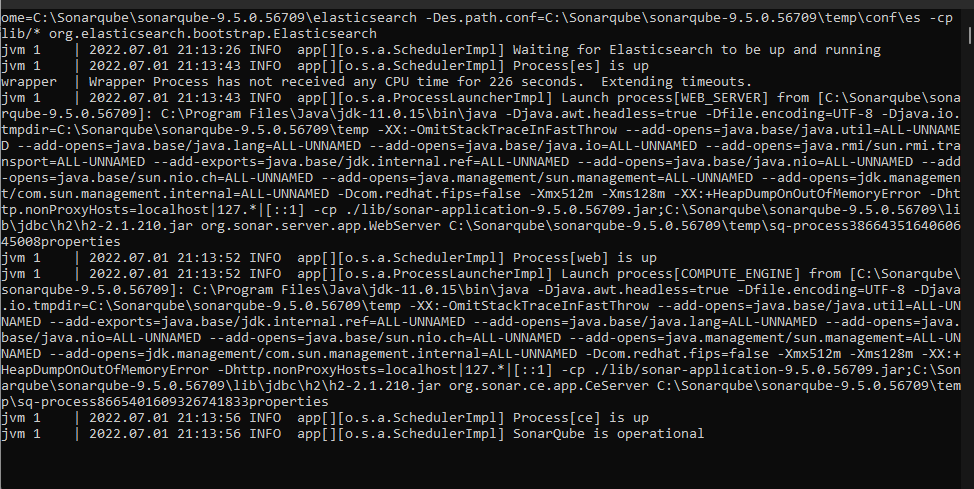
# Integrasi Jacoco dan SonarQube

Nama : Mikail Gibran Fernanda Lubis

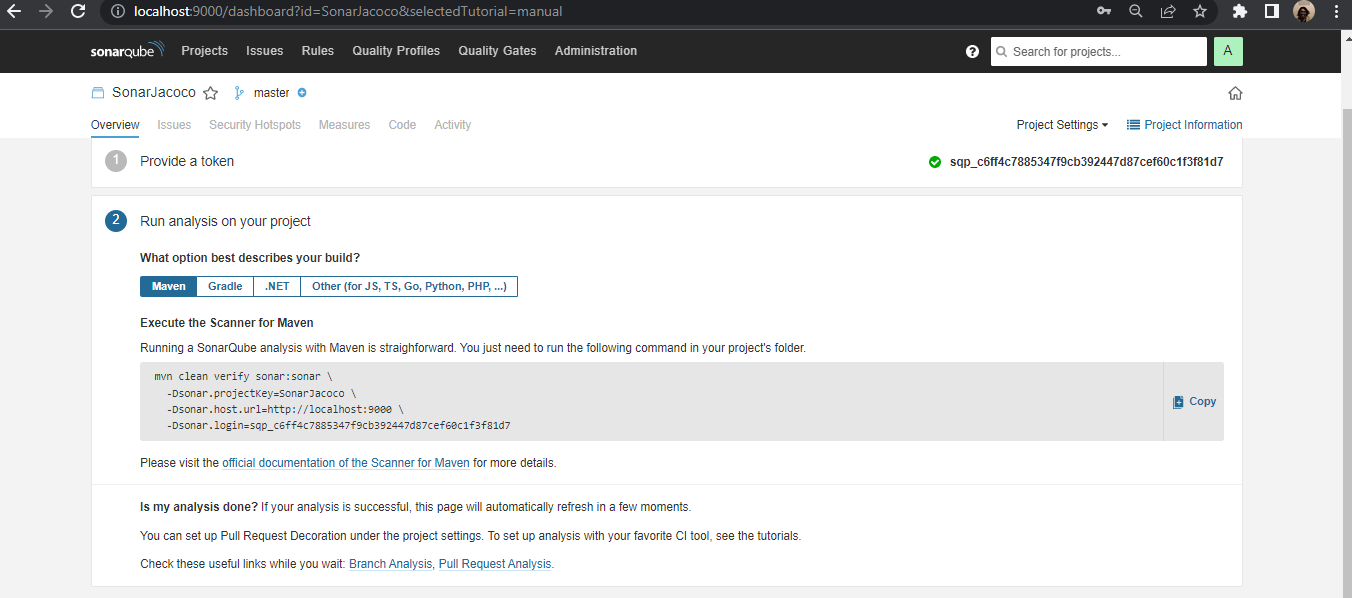
Kode Peserta: JVSB001ONL020

# Konfigurasi SonarQube

* Download SonarQube di website <https://www.sonarqube.org/downloads/>
* Membuka direktori sonarqube di cmd
* Melakukan run sonarqube dengan memasukkan perintah StartSonar.bat
* Menunggu hingga command yang ada di cmd menunjukkan bahwa sonarqube is operational seperti gambar dibawah ini:



* Setelah Sonarqube beroperasi, sekarang masuk ke localhost:9000 dimana localhost tersebut adalah sebuah dashboard dari sonarqube.
* Membuat project manually di sonarqube dan memilih versi maven, kemudian sonarqube akan memberikan kode untuk mengintegrasikan file project dengan dashboard sonarqube seperti dibawah ini

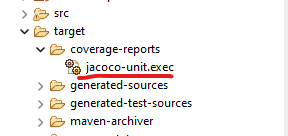


# Menjalankan SonarQube

* Ketika sonarqube sudah jalan, pilih salah satu project untuk dilakukan pengetesan, dalam kasus ini, saya menggunakan project BelajarBasicAuth.
* Di dalam project BelajarBasicAuth, ditambahkan dependency yang diperlukan untuk melakukan pengetesan sonarqube. Dependency tersebut diantaranya ialah seperti kode dibawah ini

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>  <parent>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-parent</artifactId>  <version>2.4.2</version>  <relativePath /> <!-- lookup parent from repository -->  </parent>  <groupId>com.belajar.spring</groupId>  <artifactId>BelajarAPILatihan</artifactId>  <version>0.0.1-SNAPSHOT</version>  <name>BelajarAPILatihan</name>  <description>Demo project for Spring Boot</description>  <properties>  <java.version>17</java.version>  <problem-spring-web.version>0.25.0</problem-spring-web.version>  <jacoco.version>0.8.3</jacoco.version>  <sonar.java.coveragePlugin>jacoco</sonar.java.coveragePlugin>  <sonar.dynamicAnalysis>reuseReports</sonar.dynamicAnalysis>  <sonar.jacoco.reportPath>${project.basedir}/../target/jacoco.exec</sonar.jacoco.reportPath>  <sonar.language>java</sonar.language>  </properties>  <dependencies>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-data-jpa</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-thymeleaf</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <scope>runtime</scope>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-test</artifactId>  <scope>test</scope>  <exclusions>  <exclusion>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  </exclusion>  </exclusions>  </dependency>  <dependency>  <groupId>org.junit.jupiter</groupId>  <artifactId>junit-jupiter-engine</artifactId>  <scope>test</scope>  </dependency>  <dependency>  <groupId>org.junit.jupiter</groupId>  <artifactId>junit-jupiter-params</artifactId>  <scope>test</scope>  </dependency>  <dependency>  <groupId>org.junit.platform</groupId>  <artifactId>junit-platform-launcher</artifactId>  <version>1.3.2</version>  <scope>test</scope>  </dependency>  <dependency>  <groupId>org.junit.vintage</groupId>  <artifactId>junit-vintage-engine</artifactId>  <version>5.3.2</version>  <scope>test</scope>  </dependency>  <dependency>  <groupId>org.mockito</groupId>  <artifactId>mockito-junit-jupiter</artifactId>  <scope>test</scope>  </dependency>  <dependency>  <groupId>org.apache.commons</groupId>  <artifactId>commons-lang3</artifactId>  <version>3.9</version>  </dependency>  <dependency>  <groupId>org.zalando</groupId>  <artifactId>problem-spring-web-starter</artifactId>  <version>${problem-spring-web.version}</version>  <type>pom</type>  </dependency>  <dependency>  <groupId>org.projectlombok</groupId>  <artifactId>lombok</artifactId>  <optional>true</optional>  </dependency>  <dependency>  <groupId>io.github.jpenren</groupId>  <artifactId>thymeleaf-spring-data-dialect</artifactId>  <version>3.4.0</version>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-security</artifactId>  </dependency>  </dependencies>  <build>  <plugins>  <plugin>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-maven-plugin</artifactId>  </plugin>  <plugin>  <groupId>org.jacoco</groupId>  <artifactId>jacoco-maven-plugin</artifactId>  <version>${jacoco.version}</version>  <configuration>  <destFile>${basedir}/target/coverage-reports/jacoco-unit.exec</destFile>  <dataFile>${basedir}/target/coverage-reports/jacoco-unit.exec</dataFile>  <output>file</output>  <append>true</append>  <excludes>  <exclude>\*MethodAccess</exclude>  </excludes>  </configuration>  <executions>  <execution>  <id>jacoco-initialize</id>  <goals>  <goal>prepare-agent</goal>  </goals>  <phase>test-compile</phase>  </execution>  <execution>  <id>jacoco-site</id>  <phase>verify</phase>  <goals>  <goal>report</goal>  </goals>  </execution>  </executions>  </plugin>  </plugins>  </build>  </project> |

* Setelah semua dependency terpasang, dilakukan maven clean install pada project sehingga muncul file di dalam src/target/coverage-reports/jacoco-unit.exec seperti gambar dibawah ini



* Kemudian jalankan dengan memasukkan kode integrase yang didapatkan dari localhost sonarqube di tahap sebelumnya. Jika sukses, akan muncul koreksi dan data analytics dari code yang ada di project.

# Failure

Kegagalan dalam menjalankan Sonarqube juga bisa terjadi jika terdapat beberapa faktor seperti versi java yang tidak support, dependency yang tidak sesuai, hingga versi IDE yang tidak support untuk melakukan integrase dengan sonarqube

