KODE ANALISIS WEB GOAT DAN BUG FIXING WEBGOAT DENGAN MENGGUNAKAN FIND BUGS

Penyusun :

Irfan Afif - 23514027



SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2015

\*Dokumen ini juga diletakkan di README.MD

# Persiapan

Tugas ini dikerjakan dengan menggunakan Eclipse Luna dan JDK 1.8. Tools yang digunakan adalah tortoise git. Pada tugas ini dibuat folder screenshot yang menjadi rujukan tulisan ini.

Langkah awal yang dilakukan :

* Install Maven Plugin di eclipse (m2e)
* Install FindBugs Plugin di eclipse
* Clone Git
* Import Project to Eclipse
* Atur setting FindBugs menjadi menampilkan malicious code dan security (screenshot : “setting findbug eclipse.JPG”)

# Pencarian Bug

Setelah persiapan selesai, jalankan findbugs. Hasil dari findbugs syang dijalankan (screenshot : “bug found.JPG”) adalah :

* Scary – Normal Confidence :
  + Empty Database Password (DatabaseUtilities.java : line 134)
  + Empty Database Password (UserDatabase.java : line 51)
* Troubling – High Confidence
  + Non Constant String passed to execute or addBatch method on Sql statement (AbstractLesson.java : line 731)
* Of Concern – Normal Confidence
  + Field isn’t final and can’t be protected from malicious code (Course.java : line 70)
  + Field should be package protected (HammerHead.java : line 78)
* Of Concern – Low confidence
  + Field isn’t final but should be (Screen.java : line 53)
  + Nonconstant string passed to execute or addBatct method on an SQL statement (ViewDatabase.java line : 82)
  + Nonconstant string passed to execute or addBatct method on an SQL statement (CreateDB.java : line 998)
  + Nonconstant string passed to execute or addBatct method on an SQL statement (UserDatabase.java : line 90)

# Testing and Fixing

## Empty Database Password (DatabaseUtilities.java : line 134)

Bug terjadi karena ketika melakukan pemanggilan koneksi database, parameter yang digunakan menggunakan string kosong. Testing Junit untuk bug ini sulit dilakukan. Perlu dilakukan modifikasi untuk dapat dilakukan JUnit testing. JUnit testing sulit dilakukan untuk kasus ini karena berhubungan dengan state servlet dan koneksi database. Untuk bug ini, dilakukan perbaikan dengan menambahkan parameter password. Fixing dilakukan dengan melakukan pengecekan null atau empty terhadap parameter password tersebut. Setelah itu testing dengan JUnit pada kelas DatabaseUtilitesTest. Hasil perubahan dilakukan commit dengan judul commit diberi kode 1.

## Empty Database Password (UserDatabase.java : line 51)

Bug ini terjadi karena melakukan pemanggilan method koneksi database dengan parameter password string kosong. JUnit testing sulit dilakukan untuk kasus ini karena berhubungan dengan koneksi database. Perbaikan dilakukan dengan menggunakan fungsi yang menghasilkan password untuk database. Hasil perbaikan dilakukan commit dengan judul commit diberi kode 2.

## Non Constant String passed to execute or addBatch method on Sql statement (AbstractLesson.java : line 731)

Bug ini terjadi karena adanya penggabungan string pada query sql tanpa menggunakan prepared statement. JUnit testing sulit dilakukan karena berhubungan dengan koneksi database dan banyak membutuhkan kelas dan fungsi-fungsi yang lain. Hasil perbaikan dilakukan commit dengan judul commit diberi kode 3.

## Field isn’t final and can’t be protected from malicious code (Course.java : line 70)

Terjadi keanehan pada bug ini dikarenakan variabel yang dimaksud sudah bertipe final. Kesalahan sepertinya berada di pendeksian findbugs. Setelah ditelusuri lebih lanjut, bug ini terpengaruh oleh kode HammerHead.java line 89. Sifat bug sulit diperbaiki karena dibutuhkan oleh logic program. Dikarenakan level bug adalah of Concern, maka tidak ada aksi yang dilakukan untuk bug ini.

## Field should be package protected (HammerHead.java : line 78)

Perbaikan dilakukan dengan merubah variable menjadi private. Hasil perbaikan dilakukan commit dengan judul commit diberi kode 5.

## Field isn’t final but should be (Screen.java : line 53)

Perbaikan dilakukan dengan merubah variable menjadi final. Hasil perbaikan dilakukan commit dengan judul commit diberi kode 5.

## Nonconstant string passed to execute or addBatct method on an SQL statement (ViewDatabase.java line : 82)

SQL query didapat dari parameter websession sehingga dibagian ini tidak dapat dilakukan perbaikan bug.

## Nonconstant string passed to execute or addBatct method on an SQL statement (CreateDB.java : line 998)

Dikarenakan SQL query semua berasal dari kode, tidak ada masukan user, maka tidak diperlukan perbaikan.

## Nonconstant string passed to execute or addBatct method on an SQL statement (UserDatabase.java : line 90)

Bug terjadi karena ada SQL Query yang memasukkan parameter dengan menggunakan String replace. Perbaikan dilakukan dengan menggunakan PreparedStatement

# Penutup

Analisis kode dengan menggunakan findbugs mudah untuk dilakukan. Walaupun begitu, kode analisis dengan menggunakan findbugs belum cukup. Bug yang ditemukan pun masih terdapat kemungkinan salah identifikasi. Terjadi satu bug false positif yaitu kasus nomor delapan. Terdapat juga bug yang tidak dapat diperbaiki karena bertabrakan dengan logika program, terutama pada kasus no 4 dan 7. Walupun begitu, sangat baik jika analisis findbugs dijadikan standar untuk pembuatan program.

Dalam tugas ini, kesulitan terbesar adalah dalam hal Testing. Hampir sebagian besar bug tidak dapat dilakukan Unit Testing karena kodenya yang saling berkaitan antara satu dengan yang lain, terutama antara WebServlet ataupun session dan dengan koneksi database. Hal ini dikarenakan kode pada bug yang ditemukan tersebut tidak di desain untuk dilakukan unit testing. Salah satu saran yang dapat diberikan dari tugas ini adalah desain kode harus mengakomodir dilakukannya unit testing dan desain tersebut diterapkan dari awal pembuatan program.