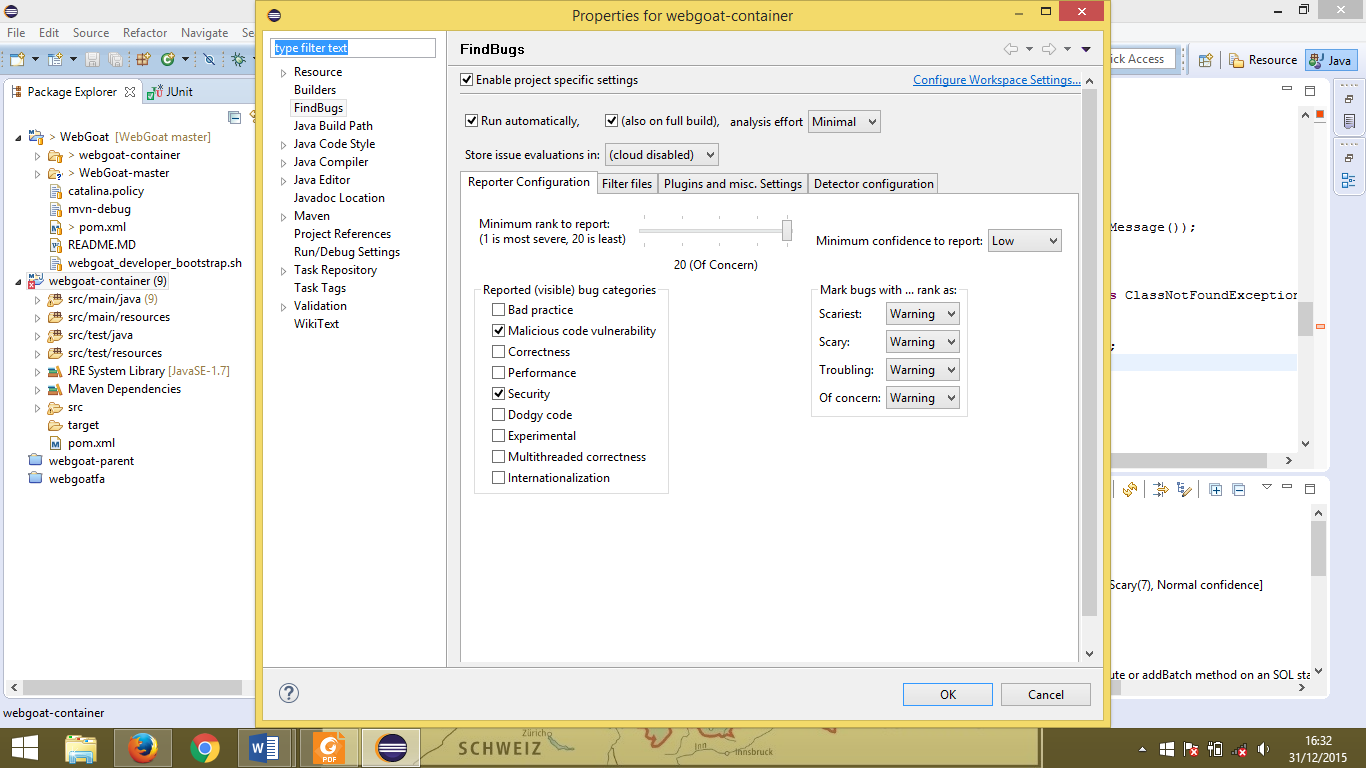
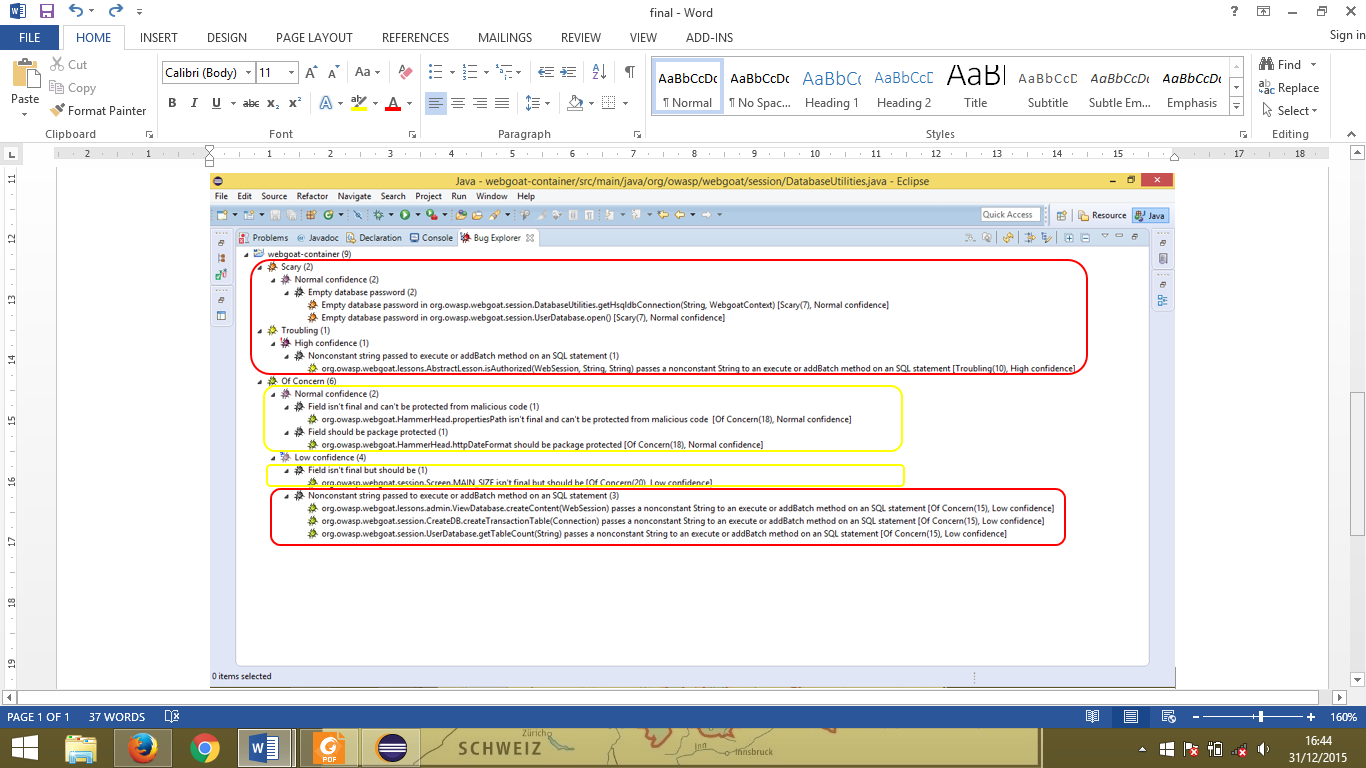
Setting FindBugs untuk mencari bug kategori “Malicious code vulnerability” dan “Security”:

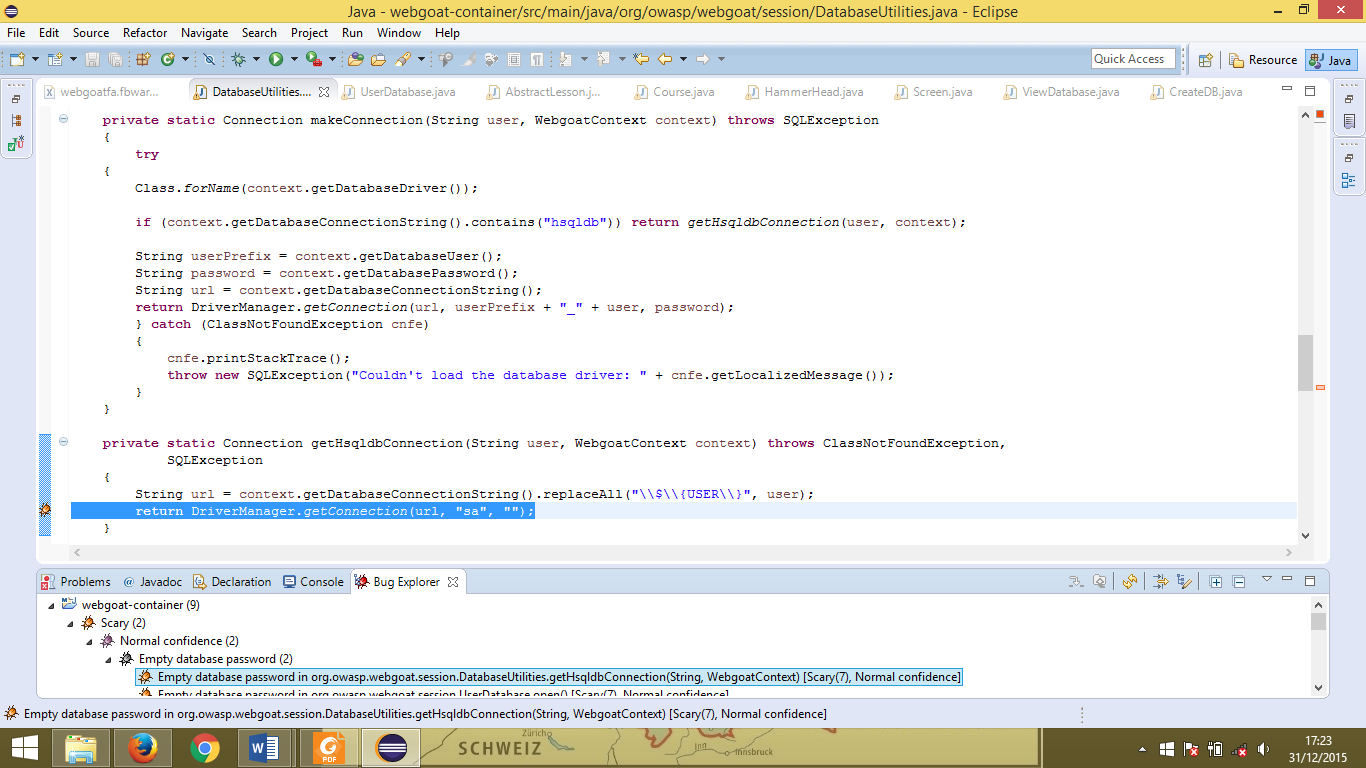


Dari hasil scan dengan FindBugs, ditemukan 9 bugs, yaitu 6 bugs kategori “security” (kotak merah) dan 3 bugs kategori “malicious code vulnerability” (kotak kuning) sebagai berikut:



**#1 Empty database password (DatabaseUtilities.java)**

Kategori: security – scary – normal confidence



**Problem:**

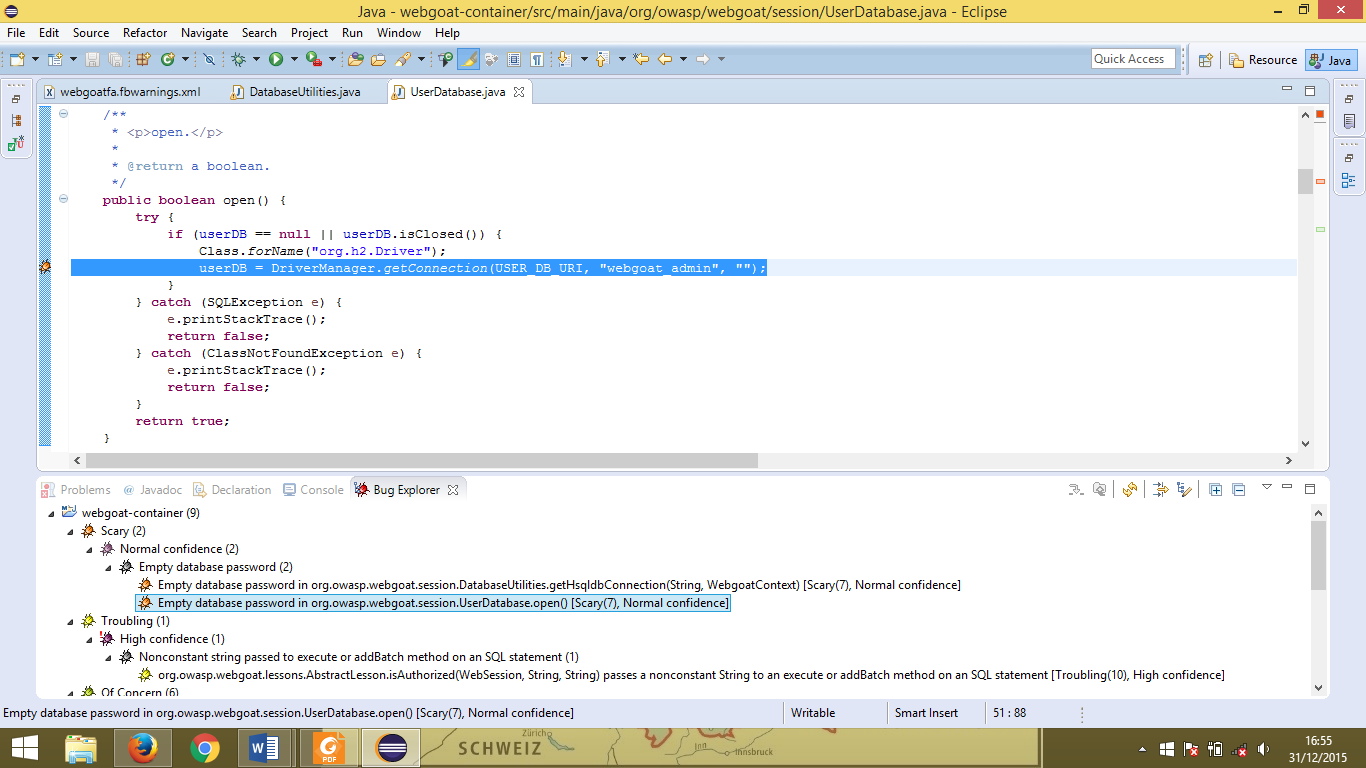
Bug terjadi karena koneksi ke database (dengan username “sa”) tanpa menggunakan password, padahal koneksi dengan username “sa” dan tanpa password tersebut merupakan koneksi default yang dihasilkan oleh hsqldb.

**Perbaikan:**

Tidak akan dilakukan perbaikan karena perlu mengubah konfigurasi untuk koneksi ke database.

**#2 Empty database password (UserDatabase.java)**

Kategori: security – scary – normal confidence



**Problem:**

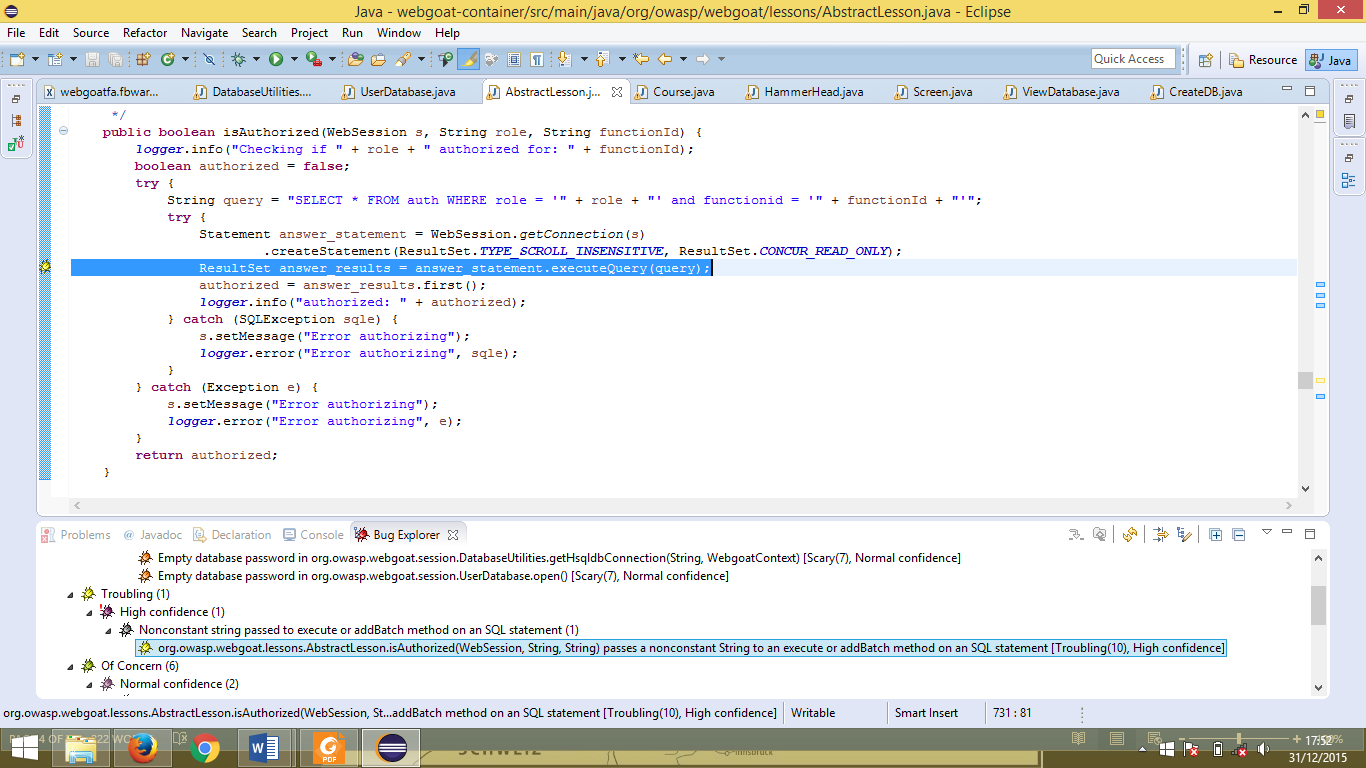
Bug ini sama dengan bug #1, yaitu karena koneksi ke database (dengan username “webgoat\_admin”) tanpa menggunakan password.

**Perbaikan:**

Tidak akan dilakukan perbaikan karena perlu mengubah konfigurasi untuk koneksi ke database.

**#3 Nonconstant string passed to execute or addBatch method on an SQL statement (AbstractLesson.java)**

Kategori: security – troubling – high confidence

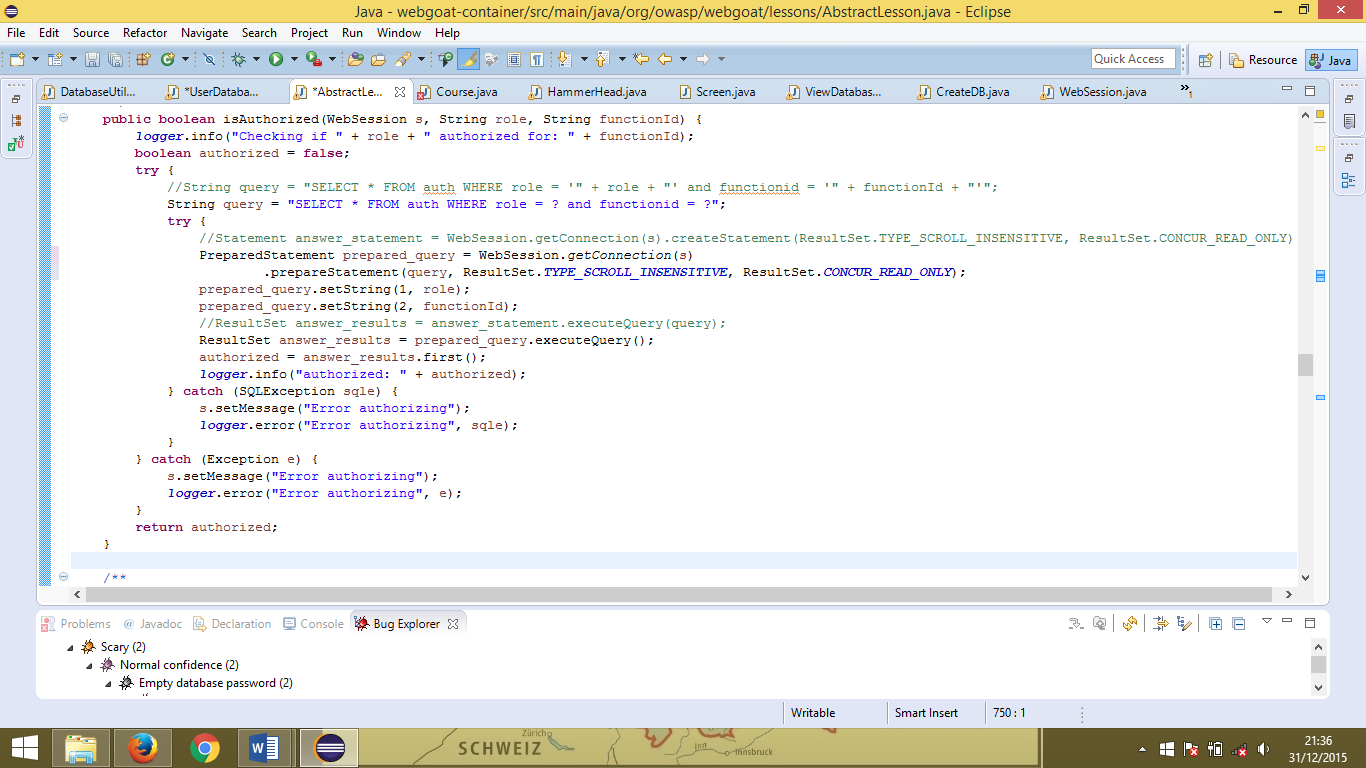


**Problem:**

Bug terjadi karena query menggunakan masukan user (paramater “role” dan “functionId”) secara langsung (tanpa disanitasi) digunakan untuk menjalankan command SQL.

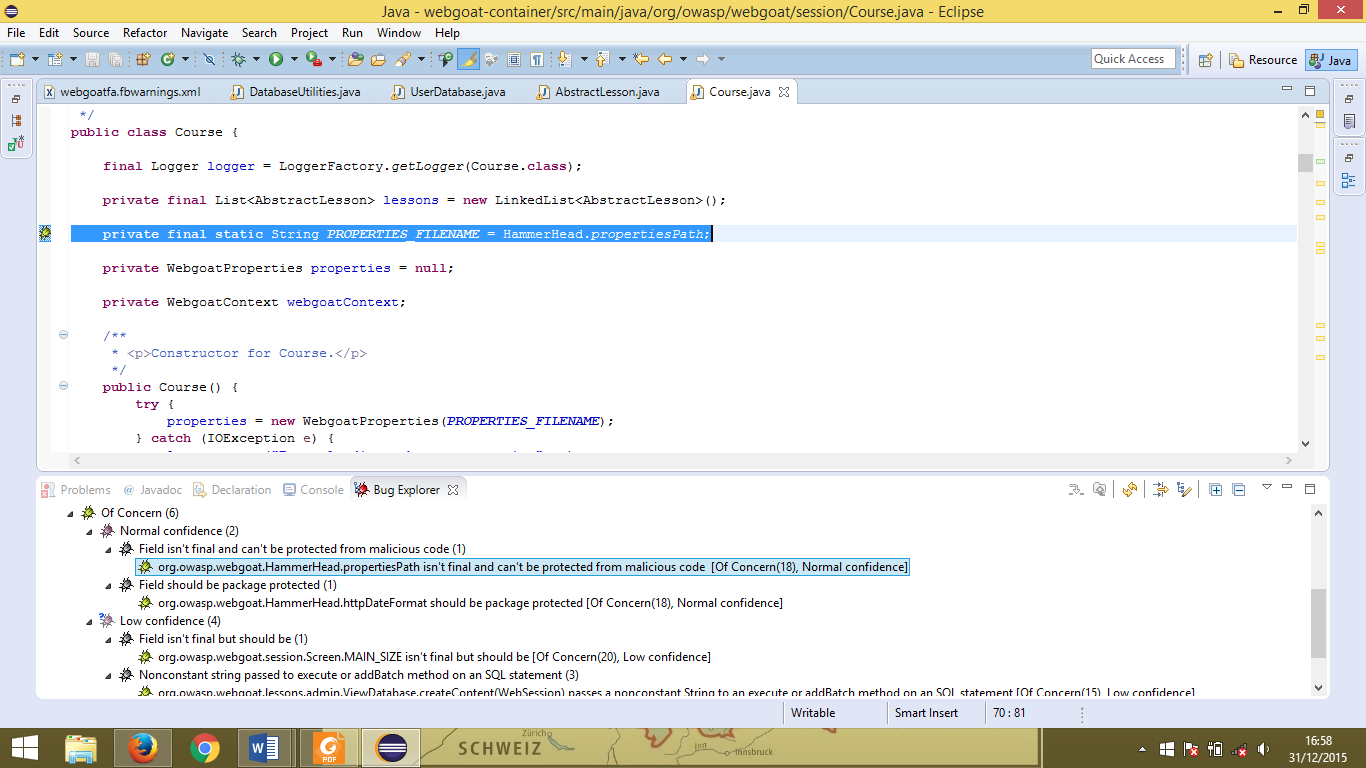
**Perbaikan:**

Menggunakan prepared statement untuk sanitasi parameter “role” dan “functionId”.



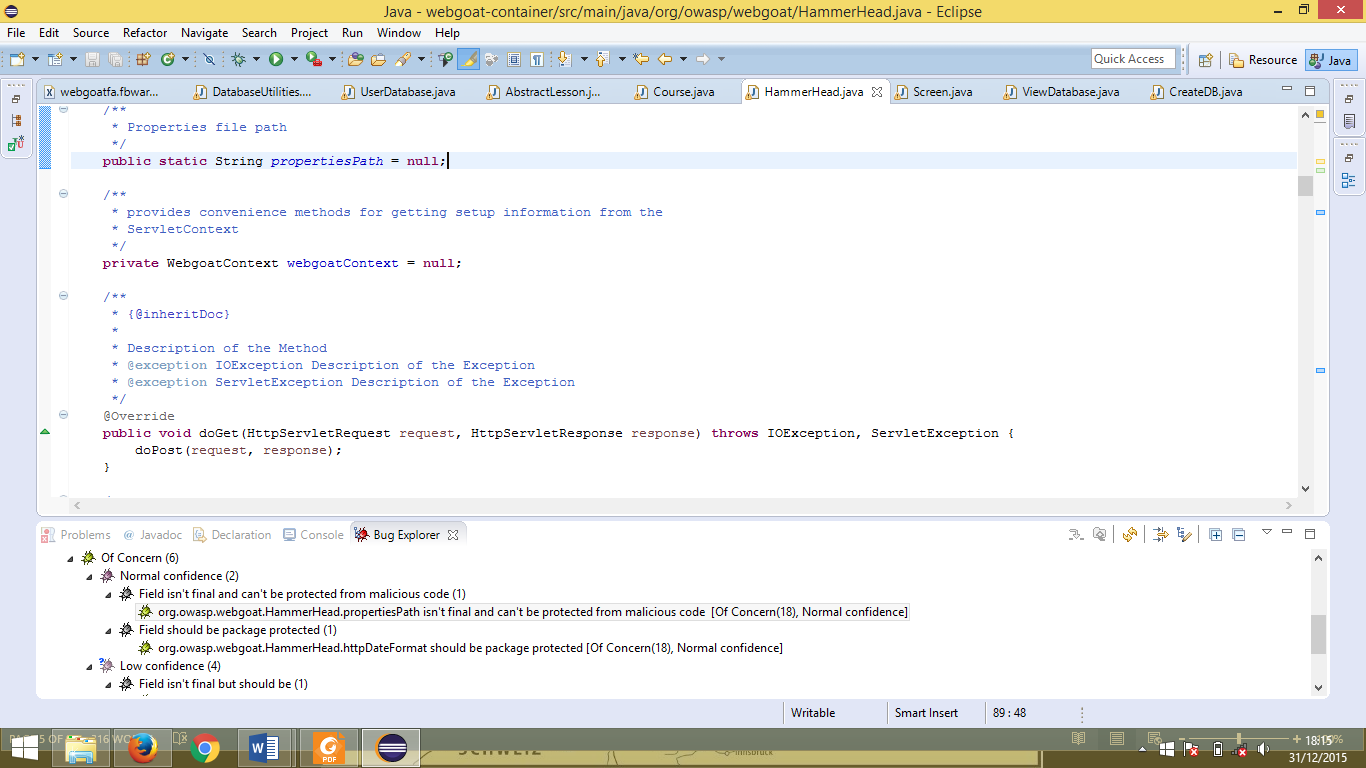
**#4 Field isn’t final and can’t be protected from malicious code (Course.java)**

Kategori: malicious code vulnerability – of concern – normal confidence



**Problem:**

Bug terjadi karena *PROPERTIES\_FILENAME* yang merupakan **final** diberi nilai yang non-final, yaitu HammerHead.propertiesPath (jika ditelusuri ke file HammerHead.java diperoleh bahwa propertiesPath non-final)

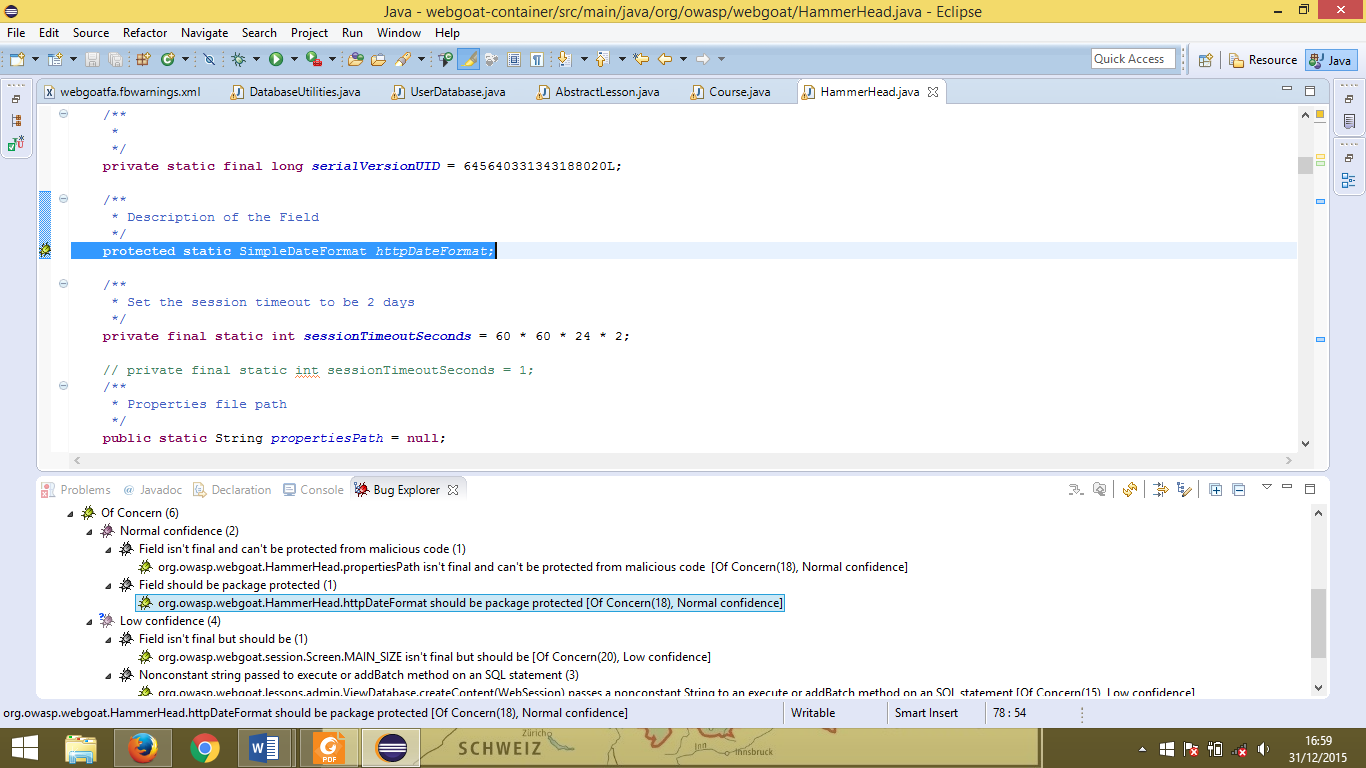


**Perbaikan:**

Tidak dapat dilakukan perbaikan, karena terkait dengan logika program. Seharusnya variabel final diberi nilai final juga.

**#5 Field should be package protected (HammerHead.java)**

Kategori: malicious code vulnerability – of concern – normal confidence

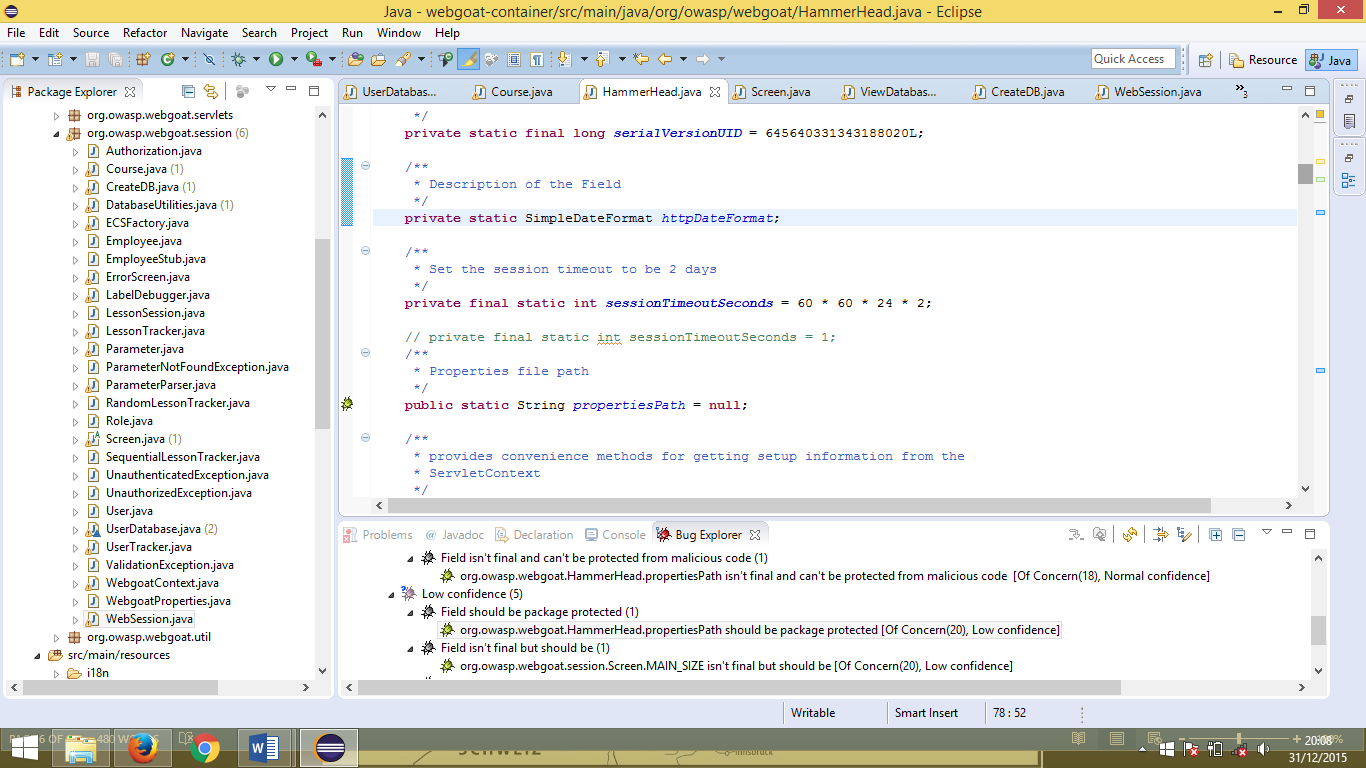


**Problem:**

Variabel httpDateFormat seharusnya diproteksi agar tidak dapat diakses sembarangan dan dimodifikasi.

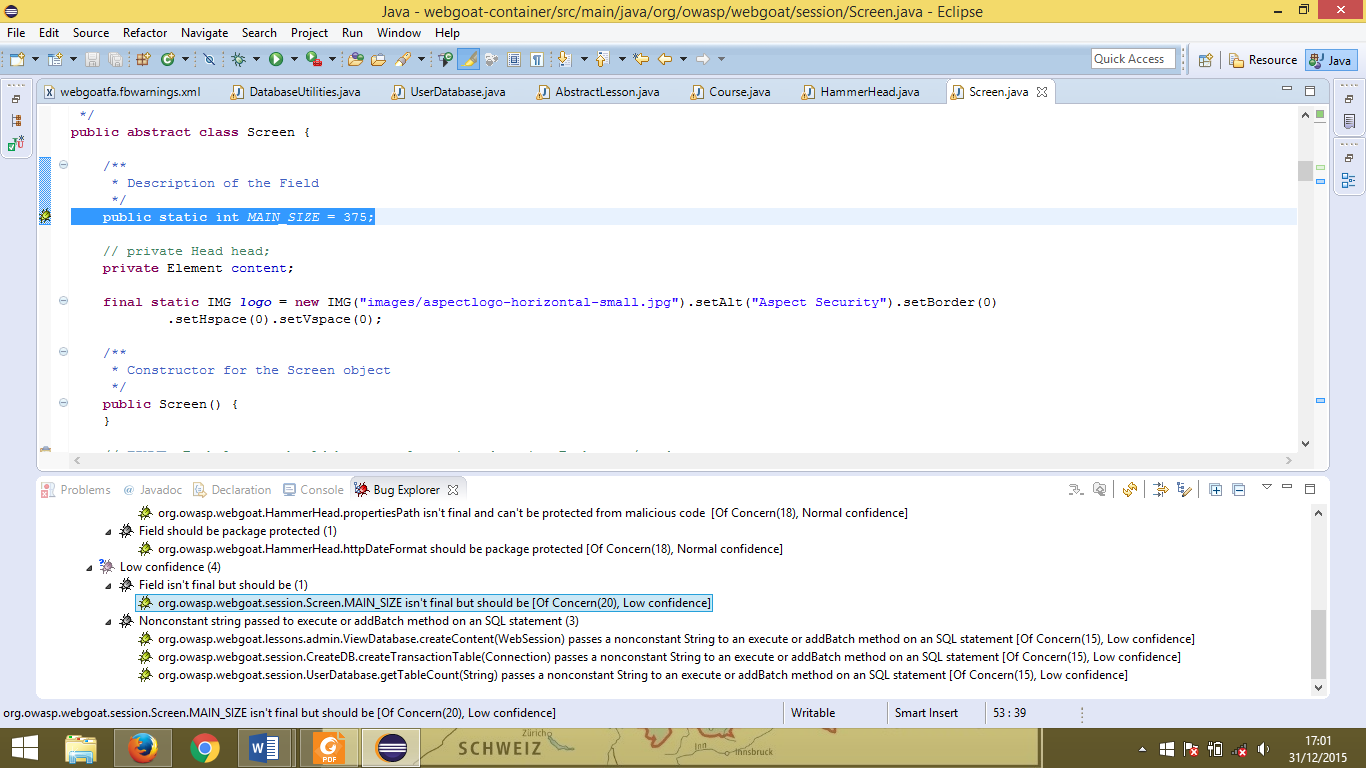
**Perbaikan:**

Mengubah protected menjadi private.



**#6 Field isn’t final but should be (Screen.java)**

Kategori: malicious code vulnerability – of concern – low confidence

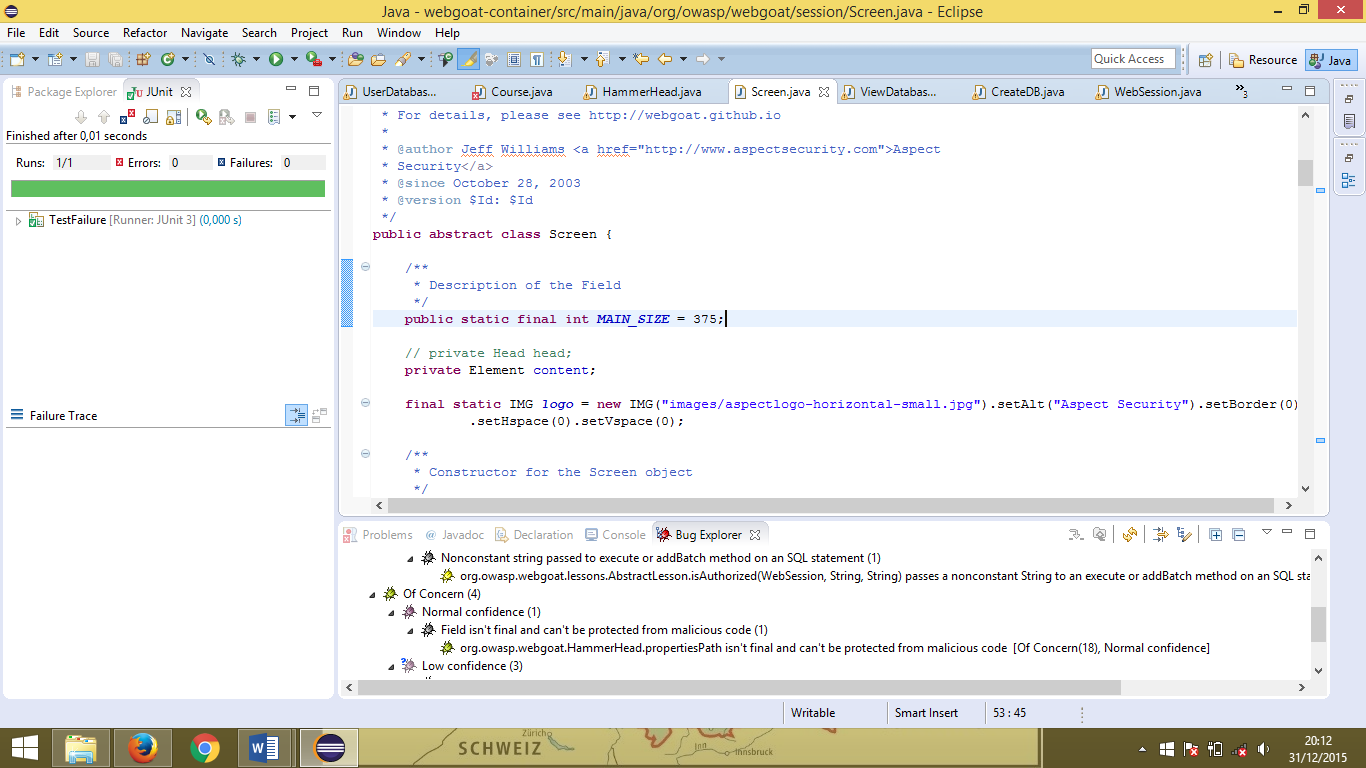


**Problem:**

Variabel *MAIN\_SIZE* seharusnya merupakan variabel final.

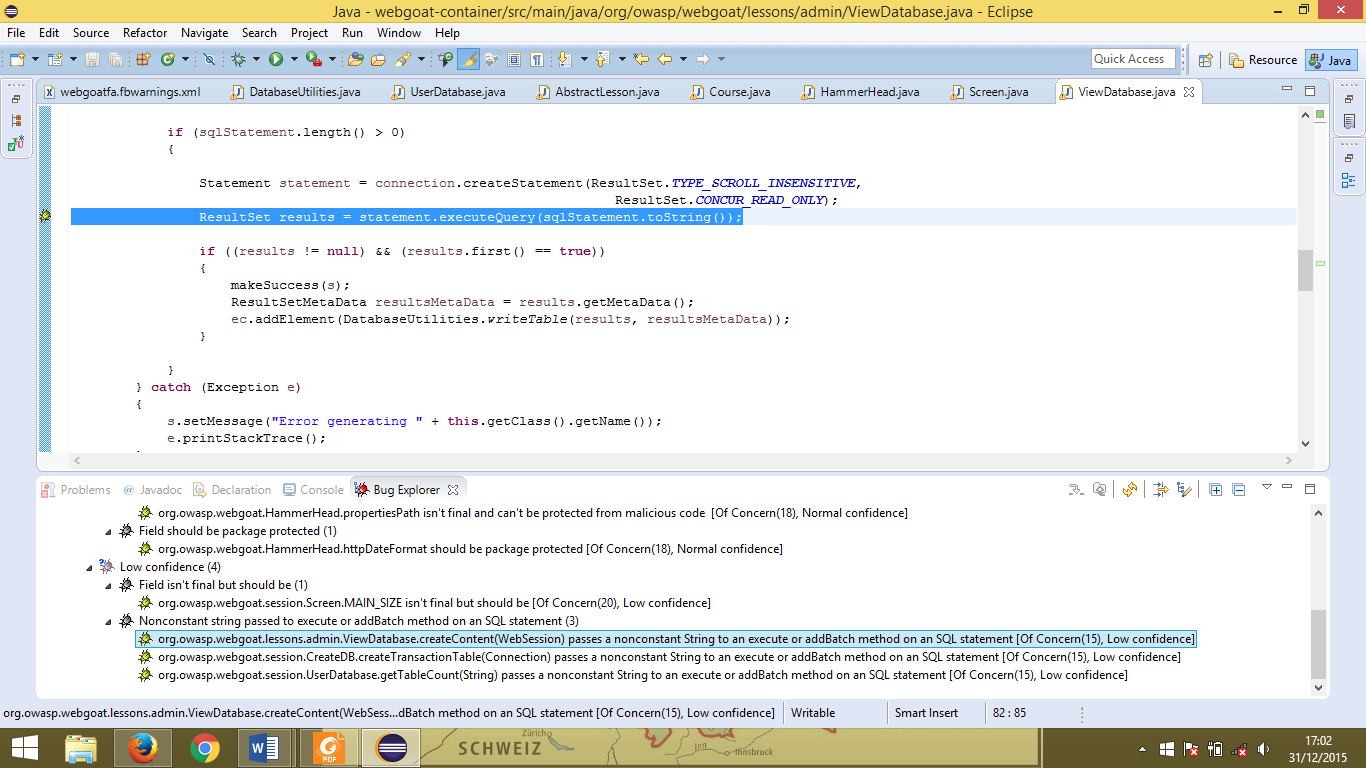
**Perbaikan:**

Mengubah variabel *MAIN\_SIZE* menjadi final.



**#7 Nonconstant string passed to executed or addBatch method on an SQL statement (ViewDatabase.java)**

Kategori: security – of concern – low confidence



**Problem:**

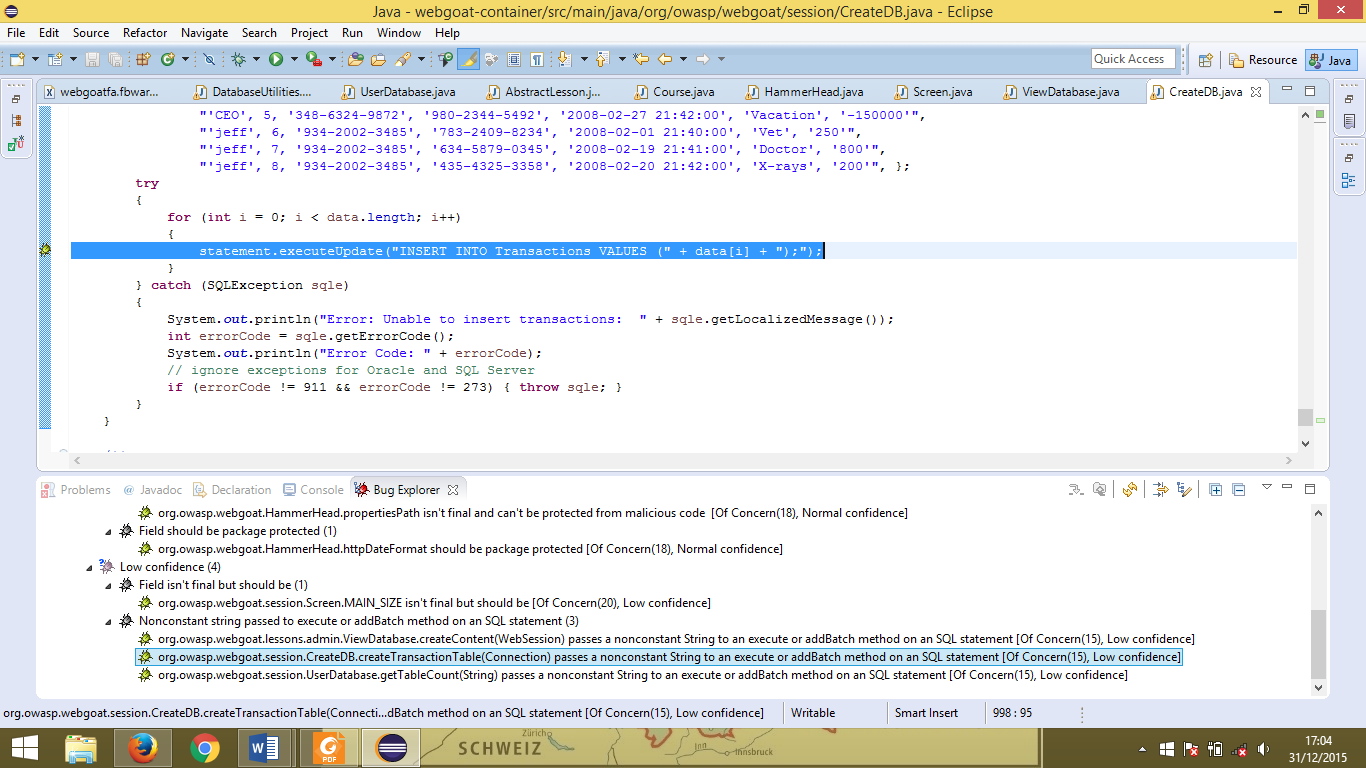
Bug terjadi karena eksekusi query statement SQL (yaitu **results**) yang berasal dari ‘user input’ (yaitu variabel **sqlStatement** yang diperoleh dari WebSession.java).

**Perbaikan:**

Tidak dapat dilakukan perbaikan, karena SQL query diperoleh dari parameter WebSession.

**#8 Nonconstant string passed to executed or addBatch method on an SQL statement (CreateDB.java)**

Kategori: security – of concern – low confidence

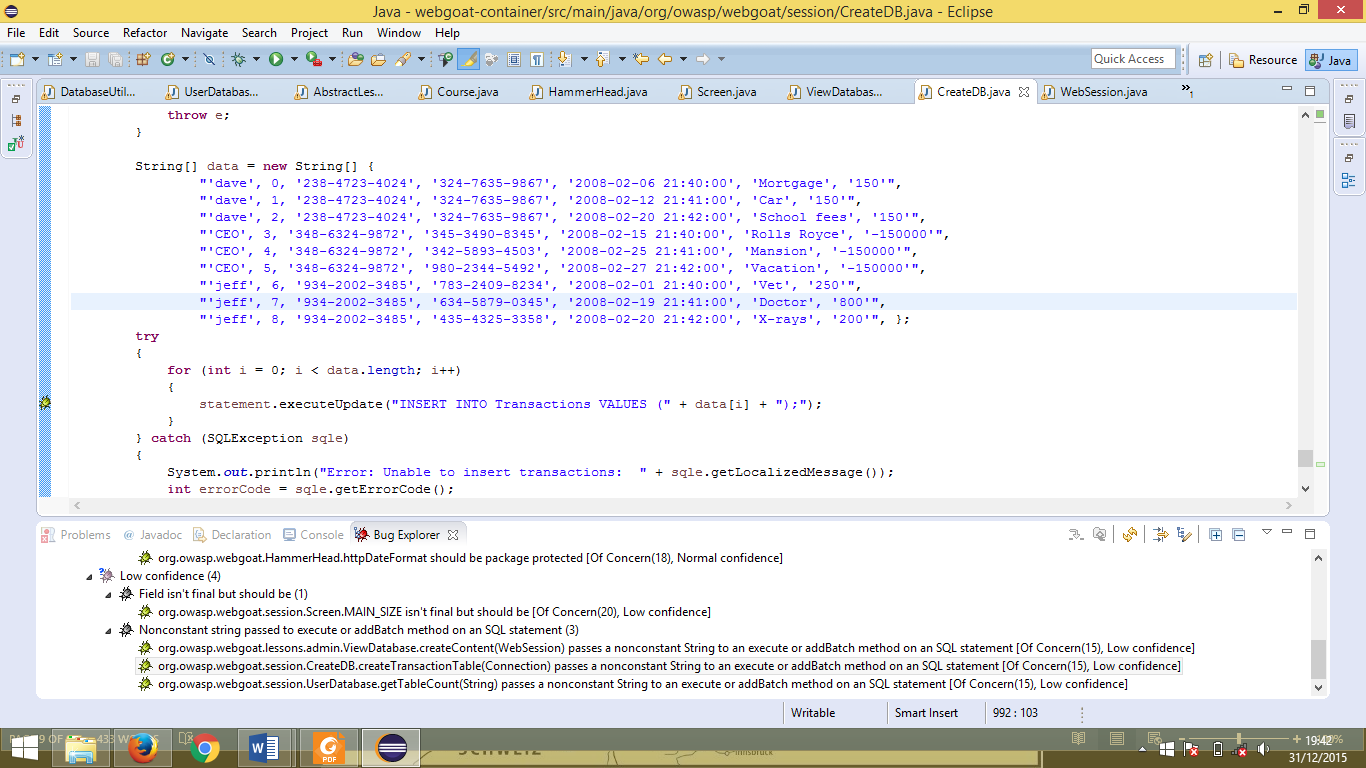


**Problem:**

Bug terjadi karena statement untuk query SQL terdapat variabel (yaitu **data[i]**) yang dianggap sebagai user input tanpa sanitasi, sehingga oleh FindBugs dianggap vulnerable.

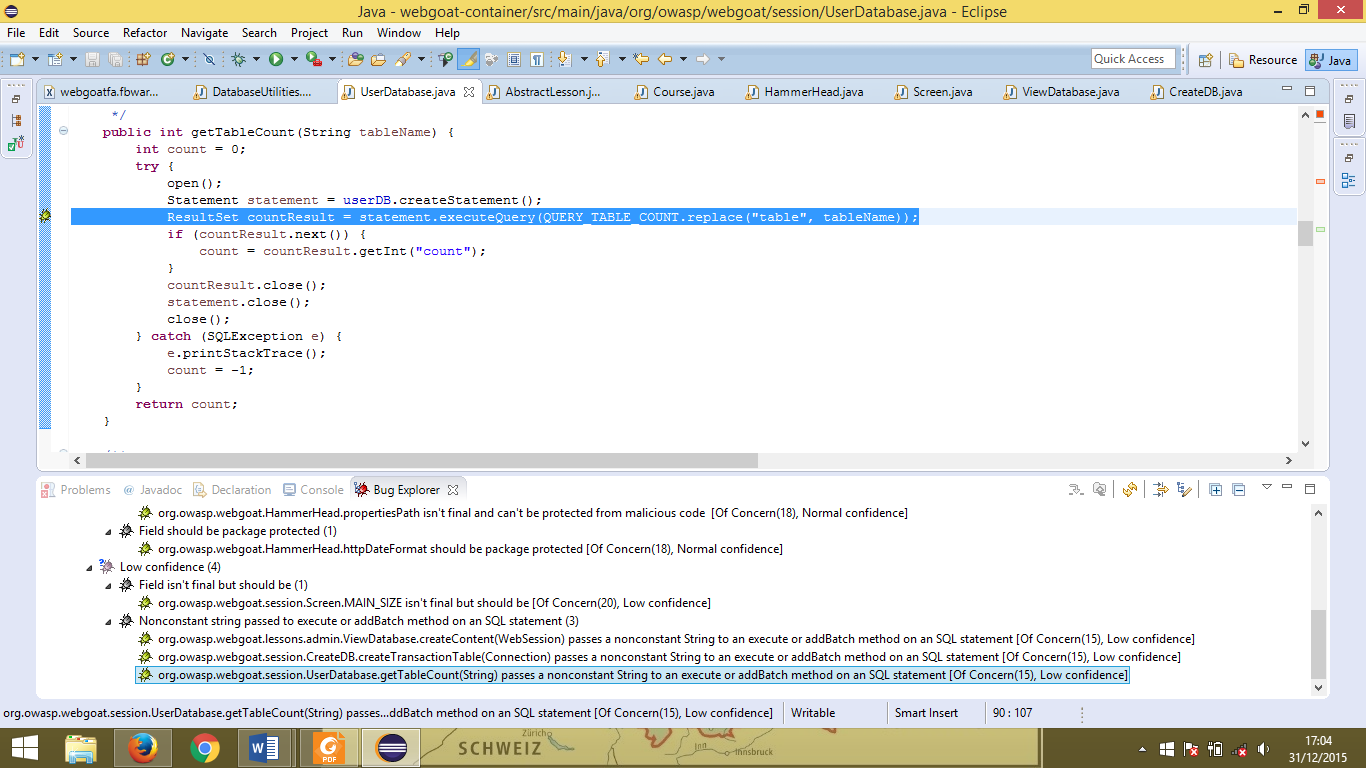
**Perbaikan:**

Tidak dilakukan perbaikan, karena variabel **data[i]** bukan user input, melainkan hardcoded dari array yang dideklarasikan sebelumnya.



**#9 Nonconstant string passed to executed or addBatch method on an SQL statement (UserDatabase.java)**

Kategori: security – of concern – low confidence



**Problem:**

Bug terjadi karena statement untuk query SQL yang dianggap menerima masukan dari user (yaitu variabel tableName yang me-replace **table** pada QUERY\_TABLE\_COUNT yang merupakan query SQL “SELECT count(id) AS count FROM **table**;” ) tanpa sanitasi.

**Perbaikan:**

Menggunakan prepared statement untuk sanitasi tableName.

