PL/SQL static code analysis: "END LOOP" should be followed by a semicolon

1-2 minutes

Labeled loops are useful, especially when the code is badly indented, to match the begin and end of each loop. However, those labels, if used, must appear on the same line as the "END" keyword in order to avoid any confusion. Indeed, the label might otherwise be seen as a procedure call.

Noncompliant Code Example

```
SET SERVEROUTPUT ON

DECLARE
  PROCEDURE foo AS
  BEGIN
    DBMS_OUTPUT.PUT_LINE('foo was called!');
  END;
BEGIN
  LOOP
    EXIT;
  END LOOP -- The semicolon was forgotten

  foo; -- Noncompliant, This is interpreted as a label of the previous FOR loop, not as a procedure call to foo!
```

```
END;
/
```

Compliant Solution

```
SET SERVEROUTPUT ON

DECLARE
  PROCEDURE foo AS
  BEGIN
    DBMS_OUTPUT.PUT_LINE('foo was called!');
  END;
BEGIN

  <<myLoopLabel>>
  LOOP
    EXIT;
  END LOOP myLoopLabel;

  foo; -- Correctly interpreted as a procedure call to foo
END;
/
```