

OLYMPIA TEACHER USER GUIDE

Create Questions - Programming Question Type



Daftar Isi

1. [Navigasi ke Halaman Question Bank](#)
2. [Programming Question \(Blackbox Grading\)](#)
3. [General](#)
4. [Response Options](#)
5. [Blackbox](#)
6. [Blackbox Languages](#)
7. [Blackbox Test Cases](#)
8. [Whitebox](#)
9. [Tags](#)

Daftar Gambar

1. [Gambar 1. Menu Courses](#)
2. [Gambar 2. Menu question bank](#)
3. [Gambar 3. Tombol “Create new Question”](#)
4. [Gambar 4. Pemilihan tipe soal “Programming”](#)
5. [Gambar 5. Contoh pengisian “Question name” dan “Question text”](#)
6. [Gambar 6. Contoh konfigurasi “Default mark”](#)
7. [Gambar 7. Contoh konfigurasi “Tries limit” dan “grader file”](#)
8. [Gambar 8. Contoh penamaan “Test case”](#)
9. [Gambar 9. “Test case” langsung di zip tanpa dimasukkan ke dalam folder](#)
10. [Gambar 10. Contoh konfigurasi “Blackbox”](#)
11. [Gambar 11. Contoh bahasa pemrograman yang dipilih](#)
12. [Gambar 12. Contoh konfigurasi “Blackbox Test Cases”](#)
13. [Gambar 13. Konfigurasi “Blackbox Test Cases” otomatis](#)
14. [Gambar 14. Tombol “Save changes”](#)
15. [Gambar 15. Tombol “save changes and continue editing”](#)

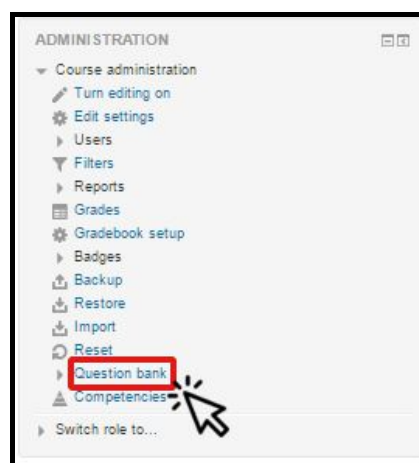
Navigasi ke Halaman Question Bank

1. Klik panah di sebelah kiri “My course” dalam menu “Navigation”.



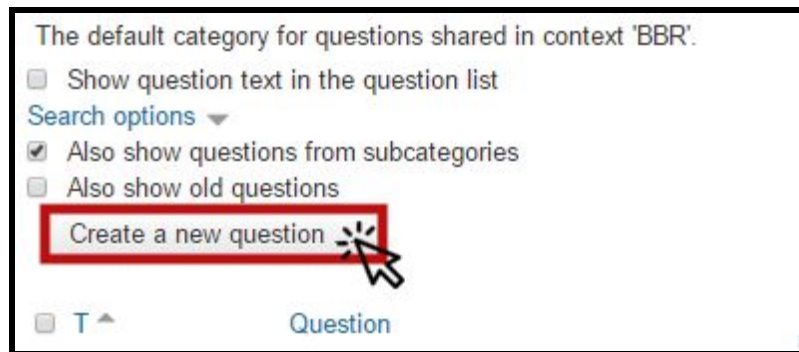
Gambar 1. Menu Courses

2. Klik “Course” yang terkait dengan soal yang akan ditambahkan.
3. Pada halaman “Course” tersebut, klik “Question bank” yang terletak pada block “Administration” disebelah kiri layar.



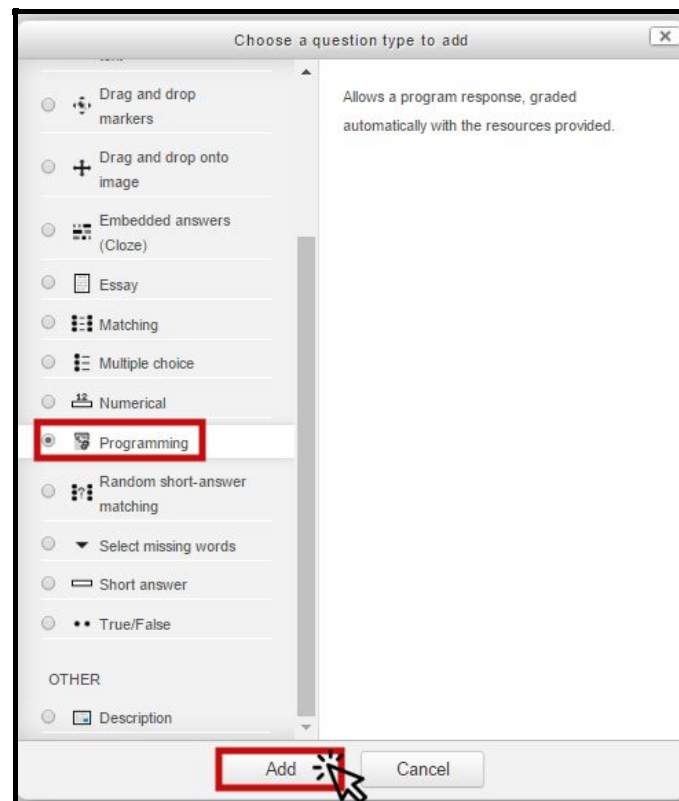
Gambar 2. Menu question bank

4. Klik tombol “Create a new question”.



Gambar 3. Tombol “Create new Question”

5. Klik menu “Programming” untuk tipe soal programming, lalu klik tombol “Add”.

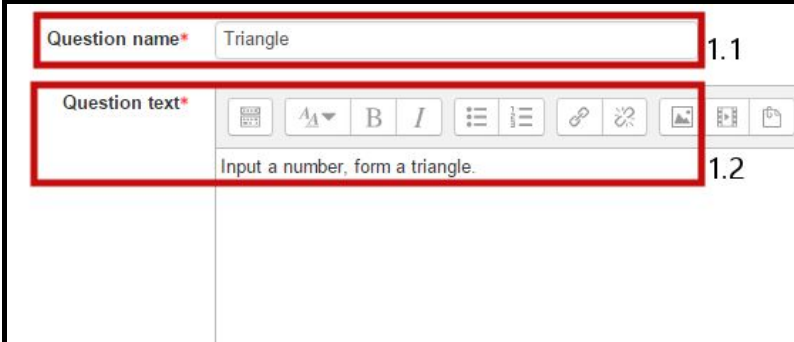


Gambar 4. Pemilihan tipe soal “Programming”

Programming Question (Blackbox Grading)

1. General

- 1.1. Isi "Question name" dengan "judul" soal .
- 1.2. Isi "Question text" dengan deskripsi soal.



The image shows a screenshot of a question editor interface. It features two main input areas, each highlighted with a red rectangular border. The first area, labeled '1.1' on the right, is for the 'Question name' and contains the text 'Triangle'. The second area, labeled '1.2' on the right, is for the 'Question text' and contains the text 'Input a number, form a triangle.' Above the text in the second area is a rich text editor toolbar with icons for text alignment, bold, italic, list creation, link, unlink, image, and document.

Gambar 5. Contoh pengisian "Question name" dan "Question text"

- 1.3. Isi bagian "Default Mark" dengan total nilai soal.
Pastikan nilai "Default Mark" adalah jumlah dari seluruh score "Exact" dan "Tolerant" (jika kedua metode digunakan) pada bagian "Blackbox Test Cases" + score maksimum dari seluruh checker yang digunakan untuk suatu bahasa pada bagian "Whitebox".



Default mark* 1.3

Gambar 6. Contoh konfigurasi “Default mark”

- 1.4. Isi bagian “Tries limit” dengan jumlah maksimum seorang peserta dapat mengulang soal.
- 1.5. Unggah “Grader Files” ke dalam form berlabel “Grader files (zip)” yang telah di-*compress* dengan format (.zip).



Tries limit* 1.4

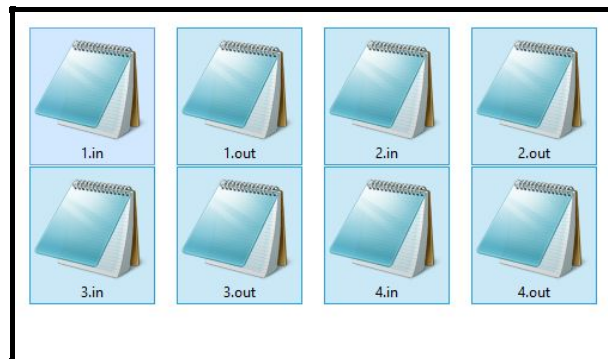
Grader files (zip)

Files

tri-tc.zip 1.5

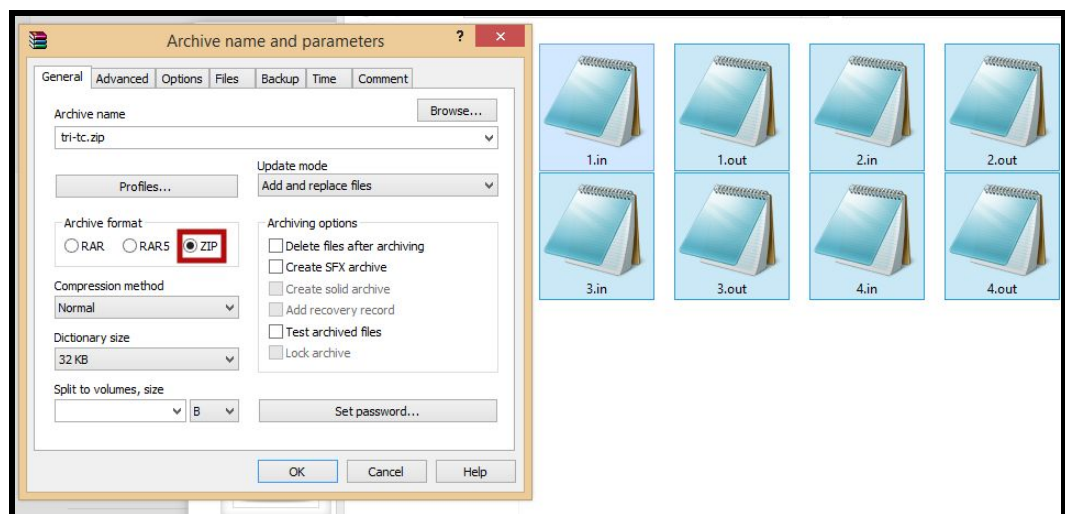
Gambar 7. Contoh konfigurasi “Tries limit” dan “grader file”

- 1.5.1. Pastikan nama file .out (output yang diharapkan) dan .in (input yang diberikan) adalah sama untuk masing-masing testcase.
Contoh : 1.in & 1.out; tca.in & tca.out



Gambar 8. Contoh penamaan “Test case”

- 1.5.2. Pastikan kompresi seluruh file “Test Case” (.in & .out) dilakukan secara langsung tanpa dimasukkan ke dalam folder terlebih dahulu.



Gambar 9. “Test case” langsung di zip tanpa dimasukkan ke dalam folder

2. Response Options

- 2.1. Tentukan respon yang akan diberikan kepada peserta saat attempt dilakukan oleh peserta pada soal ini.

3. Blackbox

Gambar 10. Contoh konfigurasi “Blackbox”

- 3.1. Isi bagian “Compile error score” dengan nilai yang diberikan jika file berisi source code yang dikumpulkan tidak dapat di-*compile*.
- 3.2. Penilaian secara “Exact” dilakukan dengan membandingkan string antara testcase (.out) dan output hasil eksekusi source code peserta yang telah diberi input testcase (.in).
- 3.3. Penilaian secara “Tolerant” dilakukan dengan membandingkan string antara testcase .out dan output hasil eksekusi source code peserta yang diberi input testcase .in, namun dengan menghilangkan *whitespace* (spasi/tab) dan *newline*.
- 3.4. Secara default, soal dinilai dengan metode “Exact”.

3.5. Penjelasan lebih rinci mengenai “Blackbox” tersedia pada dokumen berikut ini : <http://goo.gl/SxOnhD>.

4. Blackbox Languages

4.1. Pilih bahasa pemrograman yang digunakan dalam soal.

4.1	C	<input checked="" type="checkbox"/>	Compile	<input type="text"/>	Run	<input type="text"/>
	C++11	<input type="checkbox"/>	Compile	<input type="text"/>	Run	<input type="text"/>
	C++14	<input checked="" type="checkbox"/>	Compile	<input type="text"/>	Run	<input type="text"/>
	Haskell	<input type="checkbox"/>	Compile	<input type="text"/>	Run	<input type="text"/>
	Java 7	<input type="checkbox"/>	Compile	<input type="text"/>	Run	<input type="text"/>
	Java 8	<input checked="" type="checkbox"/>	Compile	<input type="text"/>	Run	<input type="text"/>
	Lisp	<input type="checkbox"/>	Compile	<input type="text"/>	Run	<input type="text"/>
	Pascal	<input type="checkbox"/>	Compile	<input type="text"/>	Run	<input type="text"/>
	Plain	<input type="checkbox"/>	Compile	<input type="text"/>	Run	<input type="text"/>
	PHP 5.5	<input type="checkbox"/>	Compile	<input type="text"/>	Run	<input type="text"/>
	PHP 5.6	<input type="checkbox"/>	Compile	<input type="text"/>	Run	<input type="text"/>
	Python 2	<input type="checkbox"/>	Compile	<input type="text"/>	Run	<input type="text"/>
	Python 3	<input type="checkbox"/>	Compile	<input type="text"/>	Run	<input type="text"/>

Gambar 11. Contoh bahasa pemrograman yang dipilih

5. Blackbox Test Cases

5.1. Manual configuration

The image shows a configuration interface for Blackbox Test Cases. It contains four test cases, each with an input field, an output field, an exact score field, and a tolerant score field. The first test case, 'Test case 1', has its input field containing '1.in', its output field containing '1.out', its exact score field containing '25', and its tolerant score field containing '0'. Red boxes highlight the input and output fields of 'Test case 1', the exact score field of 'Test case 1', and the tolerant score field of 'Test case 1'. The other three test cases (2, 3, and 4) have their input and output fields containing '2.in', '3.in', and '4.in' respectively, and their exact and tolerant score fields containing '25' and '0' respectively. The interface is titled '5.1.1' and '5.1.2' and '5.1.3' are also visible near the first test case's fields.

Test Case	Input	Output	Exact score	Tolerant score
Test case 1	1.in	1.out	25	0
Test case 2	2.in	2.out	25	0
Test case 3	3.in	3.out	25	0
Test case 4	4.in	4.out	25	0

Gambar 12. Contoh konfigurasi “Blackbox Test Cases”

- 5.1.1. Tulis nama file beserta ekstensi dari pasangan “Test Case” yang terdapat di dalam file (.zip). Lihat bagian [1.5 - Grader Files](#)
- 5.1.2. Isi nilai “Exact” dan “Tolerant” (jika keduanya digunakan) pada textbox berlabel “Exact score” dan “Tolerant score” untuk masing-masing “Test Case”.
- 5.1.3. Jika metode “Tolerant” tidak digunakan, maka secara otomatis form nilai berlabel “Tolerant score” tidak dapat diisi.

5.2. Automatic configuration

The image shows a configuration form for 'Blackbox Test Cases'. It contains four test case entries. The first entry, 'Test case 1', has its 'Input' and 'Output' fields highlighted with red boxes. Below it, the sub-section '5.2.1' has its 'Exact score' and 'Tolerant score' fields highlighted with red boxes. The other three test cases ('Test case 2', 'Test case 3', and 'Test case 4') have empty input/output fields and a 'Tolerant score' of 0.

Gambar 13. Konfigurasi “Blackbox Test Cases” otomatis

- 5.2.1. Kosongkan seluruh textbox pada form “Blackbox Test Cases”.
- 5.2.2. Grader akan secara otomatis melakukan penilaian berdasarkan testcase yang disediakan pada bagian [1.5 - Grader Files](#).
- 5.2.3. Proporsi nilai akan dibagi rata dengan hirarki sebagai berikut:
 - 5.2.3.1. Nilai pada “Default Mark” dibagi 2, masing-masing dialokasikan untuk penilaian “Whitebox” dan “Blackbox”.
 - 5.2.3.1.1. Jika penilaian “Whitebox” tidak digunakan, maka nilai pada “Default Mark” sepenuhnya dialokasikan untuk penilaian “Blackbox”.
 - 5.2.3.1.2. Jika penilaian “Blackbox” tidak digunakan, maka nilai pada “Default Mark” sepenuhnya dialokasikan untuk penilaian “Whitebox”.
 - 5.2.3.2. Pada penilaian “Blackbox”, nilai akan dibagi 2 kembali, masing-masing dialokasikan untuk metode “Exact” dan

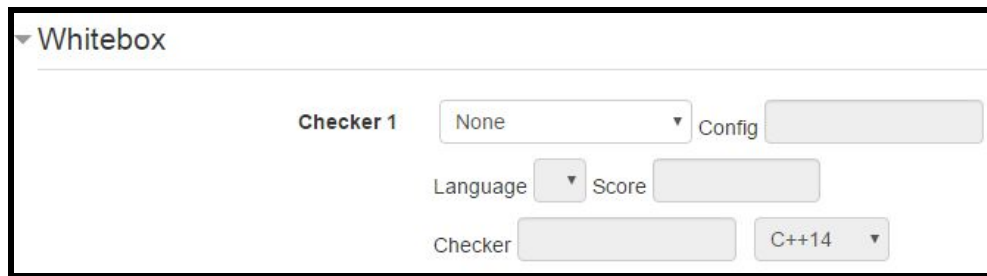
“Tolerant”. Lihat [bagian 3](#) tentang metode “Exact” dan “Tolerant”.

5.2.3.2.1. Jika salah satu metode (“Exact”/“Tolerant”) tidak digunakan, maka nilai akan sepenuhnya dialokasikan pada metode lainnya yang digunakan.

5.2.4. Pastikan jumlah nilai “Exact” dan “Tolerant” pada form “Blackbox Test Cases” sama dengan nilai pada [“Default Mark”](#).

5.3. Penjelasan lebih rinci mengenai “Blackbox Test Cases” dan teknis penilaian tersedia pada dokumen Olympia - Teacher User Guide (Detailed Guide).

6. Whitebox



Gambar 14. Whitebox configuration

- 6.1. Pilihlah “checker” yang tersedia pada dropdown menu. Terdapat 11 opsi yang dapat dipilih yaitu:
- 6.1.1. No Operation Checker
 - 6.1.1.1. Tidak dilakukan pengecekan pada “No Operation Checker.”
 - 6.1.1.2. “No operation checker” dapat digunakan apabila soal Anda dapat dijawab dengan menggunakan beberapa alternatif bahasa pemrograman, dan hanya dilakukan evaluasi “whitebox” untuk beberapa bahasa saja.
 - 6.1.1.2.1. “No Operation checker” digunakan agar total mark yang didapatkan bahasa lainnya yang tidak melalui evaluasi “whitebox” sesuai dengan “default mark”.
 - 6.1.2. Custom Checker
 - 6.1.2.1. Jika Anda hendak menggunakan “custom checker”, maka file “checker” harus diunggah pada grader files bersama kelengkapan soal lainnya.
 - 6.1.3. CPD
 - 6.1.3.1. “Copy/Paste Detector”, berfungsi untuk menemukan duplikasi kode pada source code. Dokumentasi lengkap

dapat di akses pada tautan berikut :

<https://pmd.github.io/pmd-5.4.0/usage/cpd-usage.html>

6.1.3.2. Bahasa pemrograman yang didukung dan tersedia di Olympia : C, C++11, C++14, Haskell, Java 7, Java 8, PHP 5.6, PHP 7.0, Python 2, Python 3.

6.1.4. Checkstyle

6.1.4.1. “Checkstyle” berfungsi untuk otomasi pengecekan kesesuaian kode java dengan aturan yang diberlakukan. Dokumentasi lengkap dapat di akses pada tautan berikut : <http://checkstyle.sourceforge.net/>.

6.1.4.2. Bahasa pemrograman yang didukung dan tersedia di Olympia: Java 7, Java 8.

6.1.5. PMD

6.1.5.1. PMD berfungsi untuk menemukan kekurangan dalam kode java. Contoh : dan lain-lain. Dokumentasi lengkap dapat di akses pada tautan berikut :

<https://pmd.github.io/pmd-5.5.2/index.html>

6.1.5.2. Bahasa pemrograman yang didukung dan tersedia di Olympia: Java 7, Java 8.

6.1.6. FindBugs

6.1.6.1. FindBugs berfungsi untuk menemukan bug pada kode Java. Dokumentasi lengkap dapat diakses pada tautan berikut : <http://findbugs.sourceforge.net/>

6.1.6.2. Bahasa pemrograman yang didukung dan tersedia di Olympia : Java 7, Java 8.

6.1.7. PHP_CodeSniffer

6.1.7.1. PHP_CodeSniffer berfungsi untuk menemukan ketidaksesuaian kode PHP dengan standar yang telah

disepakati. Dokumentasi lengkap dapat di akses pada tautan berikut :

https://github.com/squizlabs/PHP_CodeSniffer/wiki

6.1.7.2. Bahasa pemrograman yang didukung dan tersedia di Olympia : PHP 5.6, PHP 7.0.

6.1.8. PHPMD

6.1.8.1. PHP Mess Detector (PHPMD) berfungsi untuk menemukan kekurangan-kekurangan dalam kode PHP yang dapat menyebabkan permasalahan seperti (namun tidak terbatas kepada):

yang tidak digunakan.

Dokumentasi lengkap dapat diakses pada tautan berikut :

<https://phpmd.org/rules/index.html>

6.1.8.2. Bahasa pemrograman yang didukung dan tersedia di Olympia : PHP 5.6, PHP 7.0.

6.1.9. Cppcheck

6.1.9.1. Cppcheck berfungsi sebagai static source code analysis tool untuk kode C/C++. Dokumentasi lengkap dapat diakses pada tautan berikut :

<http://cppcheck.sourceforge.net/manual.pdf>

6.1.9.2. Bahasa pemrograman yang didukung dan tersedia di Olympia : C, C++.

6.1.10. OCLint

6.1.10.1. OClint berfungsi sebagai static source code analysis untuk kode C, C++, dan Objective-C. Dokumentasi lengkap dapat di akses pada tautan berikut :

<http://oclint-docs.readthedocs.io/en/stable/>.

- 6.1.10.2. Bahasa pemrograman yang didukung dan tersedia di Olympia : C, C++.
- 6.1.11. HLint
 - 6.1.11.1. Hlint berfungsi sebagai static source code analysis untuk kode Haskell. Dokumentasi lengkap dapat diakses pada tautan berikut :
<https://hackage.haskell.org/package/hlint>
 - 6.1.11.2. Bahasa pemrograman yang didukung dan tersedia di Olympia : Haskell.
- 6.2. “Config” merupakan file berisi rules/aturan yang akan diacu oleh checker.
 - 6.2.1.1. Jika anda hendak menentukan “Rules” untuk “Checker” terkait yang akan di / , sertakan file config rule tersebut pada “Grader Files”, lalu Tuliskan nama file “Config” yang telah Anda sertakan pada “Grader Files”.
 - 6.2.1.2. Secara default, telah disediakan default config untuk masing-masing checker. Default config tersebut dapat digunakan dengan mengosongkan field “Config”. Default config tersebut dapat dilihat pada tautan berikut :
<https://olympia.id/mod/folder/view.php?id=759>.
 - 6.2.1.3. Masing-masing checker memiliki format penulisan rules yang berbeda. Silakan mengacu kepada dokumentasi masing-masing checker yang telah disediakan pada [point 6.1](#).

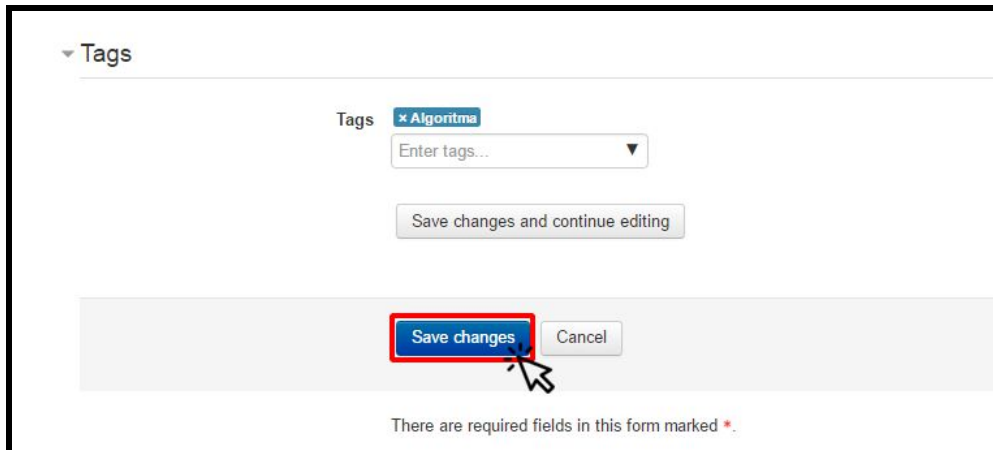
- 6.3. Pilih bahasa pemrograman yang digunakan dengan klik drop-down menu dengan label “Language”.
- 6.4. Tentukan jumlah nilai yang didapatkan apabila kode lulus uji dengan menetik nilai yang diinginkan pada “textbox” dengan label “Score”.
- 6.5. Sebagai alternatif melakukan pengisian field pada section “Whitebox”, sertakan sebuah comma separated value (csv) file bernama whitebox.csv dan kosongkan seluruh field pada section “Whitebox”. File csv yang diupload berisi tabel dengan kolom :
 - 6.5.1. Language (), score (), type (), config (), checker filename (), checker language ().
 - 6.5.1.1. “Language” adalah bahasa yang digunakan untuk menjawab soal.
 - 6.5.1.2. “Score” adalah jumlah nilai yang didapatkan apabila kode lulus uji.
 - 6.5.1.3. “Type” dapat berisi “custom” atau nama “checker” pada poin 6.1.
 - 6.5.1.3.1. Format penulisan Type Config adalah :
 - 6.5.1.3.1.1. Tulis “nop” untuk memilih No operation checker.
 - 6.5.1.3.1.2. Tulis “custom” untuk memilih custom checker.
 - 6.5.1.3.1.3. Tulis “cpd” untuk memilih CPD.
 - 6.5.1.3.1.4. Tulis “checkstyle” untuk memilih Checkstyle.
 - 6.5.1.3.1.5. Tulis “pmd” untuk memilih PMD.

- 6.5.1.3.1.6. Tulis “findbugs” untuk memilih FindBugs.
- 6.5.1.3.1.7. Tulis “phpcodesniffer” untuk memilih PHP_CodeSniffer.
- 6.5.1.3.1.8. Tulis “phpmd” untuk memilih PHPMD.
- 6.5.1.3.1.9. Tulis “cppcheck” untuk memilih Cppcheck.
- 6.5.1.3.1.10. Tulis “oclint” untuk memilih OCLint.
- 6.5.1.3.1.11. Tulis “hlint” untuk memilih HLint.
- 6.5.1.4. “Config” adalah adalah nama file yang berisi rules/aturan yang akan diacu “checker”.
 - 6.5.1.4.1. Kosongkan config apabila menggunakan config default. Lihat poin 6.2.
- 6.5.1.5. Checker filename adalah nama file custom checker. (apabila menggunakan custom checker)
- 6.5.1.6. Checker language adalah bahasa yang digunakan untuk membuat custom checker. (apabila menggunakan custom checker)

7. Tags

- 7.1. Isi “Tags” dengan kategori-kategori soal yang sesuai. Tags akan memudahkan anda dalam manajemen soal.

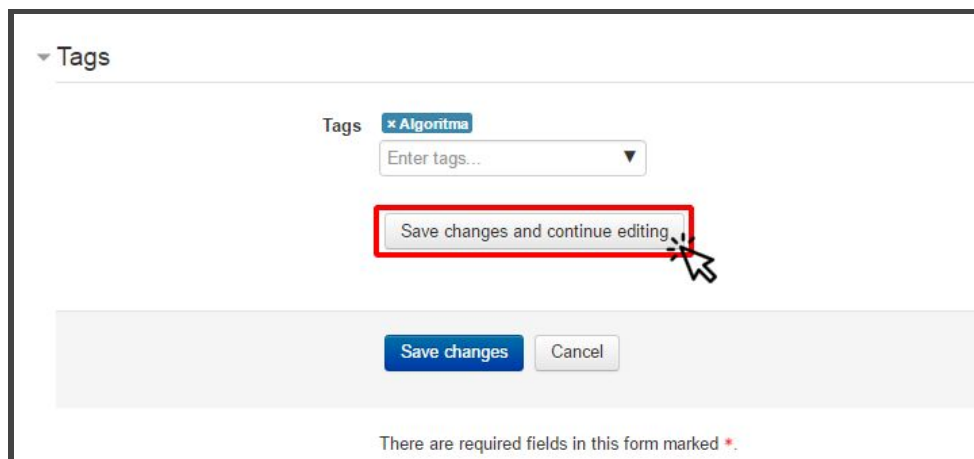
8. Klik Tombol “Save changes” untuk menyimpan soal yang telah dibuat ke bank soal.



The screenshot shows a form titled "Tags" with a dropdown menu containing "Algoritma". Below the dropdown is a text input field labeled "Enter tags...". A button labeled "Save changes and continue editing" is positioned below the input field. At the bottom of the form, there are two buttons: "Save changes" (highlighted with a red box and a mouse cursor) and "Cancel". A message at the bottom of the form reads "There are required fields in this form marked *."

Gambar 15. Tombol “Save changes”

Catatan: Klik tombol “Save changes and continue editing” pada bagian bawah halaman untuk menyimpan perubahan sementara tanpa kembali ke layar “Bank Soal”. Simpan pekerjaan Anda secara berkala.



The screenshot shows the same form as in Gambar 15. In this version, the button labeled "Save changes and continue editing" is highlighted with a red box and a mouse cursor is pointing to it. The "Save changes" button is also visible at the bottom of the form. The message "There are required fields in this form marked *." is present at the bottom.

Gambar 16. Tombol “save changes and continue editing”