

## Lampiran

### Lampiran 1 Listing

- Program Program di *module.js* Modul Dosen

```
// File name: Module.js
// Created: 01.12.2005
// Copyright (c) Websoft, 2006. All rights reserved.
//20140140008

// InitModule
function InitModule()
{
}

// ShutdownModule
function ShutdownModule()
{
}

var moduleID = 'TI 2016 UMY'; //'HTML 2016 TI UMY';
var codeID = 68;
var codeIDH = 225;
var text_assign = '';

function url_update()
{
    var url_updt = "http://gflm.omy.ac.id/_insert_text.php";
    return url_updt;
}

function url_validation()
{
    var uurl = "http://gflm.omy.ac.id/_pass_post.php";
    return uurl;
}
```

```
}
```

```
function url_send(){  
    var uuurl="http://gflm.ummy.ac.id/_send_onlinetext.php";  
    return uuurl;  
}
```

```
function url_data(){  
    var surl="http://gflm.ummy.ac.id/_data_mhs.php";  
    return surl;  
}
```

```
function callModuleID(){  
    return moduleID;  
}
```

```
function code(){  
    return codeID;  
}
```

```
function codeh(){  
    return codeIDH;  
}
```

```
function read_file()  
{  
    var Scr8 = new  
ActiveXObject("Scripting.FileSystemObject");  
    var moduleIDs = moduleID+".txt";  
    try  
    {  
        var CTF8 = Scr8.OpenTextFile(moduleIDs, 1, true);  
        encodedText = CTF8.ReadAll();  
    }  
}
```

```

        CTF8.Close();
        decodedText = s_encode(encodedText,codeID);
        natureText = hexToString(decodedText);
        return natureText;
    }
catch (err)
    {
        CTF8.Close();
        return 'Course ID is not found';
    }
}

function create_newFile(moduleID,studentId,studentName,status)
{
    var moduleIDs = moduleID + '.txt';
    var Scr2 = new
ActiveXObject("Scripting.FileSystemObject");
    var score = g_encode(score);
    var string = generate_array(100);
    array = string.split(',');

array[0]=moduleID;array[1]=studentId;array[2]=studentName;arra
y[3]=status;

    var Text = array_toText(array,100);
    var HText = stringToHex(Text);
    secureText = s_encode(HText,codeID);
    try
    {
        var CTF2 = Scr2.CreateTextFile(moduleIDs);
        CTF2.Write(secureText);
        CTF2.Close();
        return "OK";
    }
}

```

```
    catch (err)
    {
        CTF2.Close();
        alert('File saved error');
    }
}

function s_encode(str,code) {
    var encoded = "";
    for (i=0; i<str.length;i++) {
        var a = str.charCodeAt(i);
        var b = a ^ code;
        encoded = encoded+String.fromCharCode(b);
    }
    return encoded;
}

function g_encode(grade) {
    return (grade+174)*100;
}

function g_decode(higrade) {
    return ((higrade)/100 -174);
}

var myArray = new Array();

function generate_array(number) {
    var text = "*";
    for (i=1;i<number;i++){
        text = text + ',';
        myArray[i] = '*';
    }
}
```

```

        text = text + myArray[i];
    }
    return text;
}

function array_toText(myArray,length) {
    var string = myArray[0];
    for (i=1;i<length;i++){
    string = string + "," + myArray[i];
    }
    return string;
}

function stringToHex (tmp) {
    var str = '',
        i = 0,
        tmp_len = tmp.length,
        c;

    for (; i < tmp_len; i += 1) {
        c = tmp.charCodeAt(i);
        str += d2h(c) + ' ';
    }
    return str;
}

function hexToString (tmp) {
    var arr = tmp.split(' '),
        str = '',
        i = 0,
        arr_len = arr.length,
        c;

```



```
        if (result == "VALID"){
            response = updateFile();

HideObject("IMG_13","", "infinite");
            ShowObject('IMG_83');
        }
        if (result == "INVALID"){
            alert("Please check your
Student-ID and try again");

HideObject(object, "", "infinite");

HideObject("IMG_13","", "infinite");

ShowObject("IMG_85","", "infinite");

ShowObject("inputNIM","", "infinite");

ShowObject("inputNAME","", "infinite");

ShowObject("inputPASS","", "infinite");

ShowObject("TXT_54","", "infinite");

ShowObject("TXT_53","", "infinite");

ShowObject("TXT_57","", "infinite");

ShowObject("TXT_61","", "infinite");
        }
    }
    else
    {
        alert("Validation error, please get
internet connection and try again");
        HideObject(object, "", "infinite");
        HideObject("IMG_13","", "infinite");
```

```

        ShowObject("IMG_85","", "infinite");
        ShowObject("inputNIM","", "infinite");

ShowObject("inputNAME","", "infinite");

ShowObject("inputPASS","", "infinite");
        ShowObject("TXT_54","", "infinite");
        ShowObject("TXT_53","", "infinite");
        ShowObject("TXT_57","", "infinite");
        ShowObject("TXT_61","", "infinite");
    }
}
}
}

function updateFile()
{
    var text = read_file();
    var array = text.split(',');
    var moduleIDs = array[0] + '.txt';
    var Scr2 = new
ActiveXObject("Scripting.FileSystemObject");
    array[3]="VALID";
    var Text = array_toText(array,array.length);
    var HText = stringToHex(Text);
    secureText = s_encode(HText,codeID);
    try
    {
        var CTF2 = Scr2.CreateTextFile(moduleIDs);
        CTF2.Write(secureText);
        CTF2.Close();
        return "Validation is done";
    }
}

```



```

    }
catch (err)
    {
        CTF2.Close();
        alert('File saved error');
    }
}

function getStudentReport(NIM) //untuk menampilkan nilai siswa
di halaman baru
{
    var Course = moduleID;
    var url = url_send();
    var passvar= "user_id=" + NIM + "&" + "course_code=" +
Course;
    var xmlHttp = new XMLHttpRequest();
    ShowObject("IMG_13",'','infinite');
    xmlHttp.open("POST", url, true);
    xmlHttp.setRequestHeader("Content-type", "application/x-
www-form-urlencoded");
    xmlHttp.setRequestHeader("Content-length",
"passvar.length ");
    xmlHttp.setRequestHeader("Conection", "close");
    xmlHttp.send(passvar);
    xmlHttp.onreadystatechange = function()
    {
        if (xmlHttp.readyState == 4)
            {
                if(xmlHttp.status==200)
                    {
                        var text =
xmlHttp.responseText;

                        g_arVars['bufferHTML']=text;

```

```

HideObject("IMG_13","", 'infinite');
                                NextFrame();
                                }
                                else
                                {
HideObject("IMG_13","", 'infinite');
                                }
                                }
                                }
}

function sendDataPost(){
    var message = "IMG_86";
    var url = url_update();
    var json = read_file();
    var xmlhttp = new XMLHttpRequest();
    var passvar = "onlinetext="+json;
    if (xmlhttp != null){
        ShowObject(message, '', 'infinite');
        xmlhttp.open("POST", url, true);
        xmlhttp.setRequestHeader("Content-
type", "application/x-www-form-urlencoded");
        xmlhttp.setRequestHeader("Content-length",
"passvar.length");
        xmlhttp.setRequestHeader("Conection", "close");
        xmlhttp.send(passvar);
        xmlhttp.onreadystatechange = function(){
            if (xmlhttp.readyState == 4){
                if (xmlhttp.status == 200){
                    HideObject(message, '', 'infinite');
                }
            }
            else {

```



```
    var buffer = "";
var loop = "TRUE"
    while (loop){
        buffer = text.replace(',', '@@@');
        if (buffer == text) {break;}
        text = buffer;
    }
    return buffer;
}
```

```
function htmlDeCode(text){
    var buffer = "";
var loop = "TRUE"
    while (loop){
        buffer = text.replace('@@@', ',');
        if (buffer == text) {break;}
        text = buffer;
    }
    return buffer;
}
```

```
function uploadedAssignment(dom, chapter, viewer)
{
    var textfile = dom;
var array = textfile.split(',');
var i=0;
var counter = "true";
while (counter=="true")
    {
        if (array[i]==chapter)
            {
                var text = array[i+1];
```

```

        html = s_encode(text,codeh());
        html = htmlDeCode(html);
        html += " ";
        html += "</br></br>";
        html += "<p><font color='red'>Score:
"+array[i+2]+"</font></p>";
        g_arVars[viewer] = html;
        HideObject("IMG_13","", 'infinite');
        break;
    }
    i++;
    if (array[i]=="*"){break;}
    if (i==99){break;}
}
}

```

```

function saveScore(NIM, chapter, score)
{
    var Course = moduleID;
    var url = url_send();
    var passvar= "user_id=" + NIM + "&" + "course_code=" +
Course;
    var xmlHttp = new XMLHttpRequest();
    ShowObject("IMG_13","", 'infinite');
    xmlHttp.open("POST", url, true);
    xmlHttp.setRequestHeader("Content-type", "application/x-
www-form-urlencoded");
    xmlHttp.setRequestHeader("Content-length",
"passvar.length ");
    xmlHttp.setRequestHeader("Conection", "close");
    xmlHttp.send(passvar);
    xmlHttp.onreadystatechange = function()
    {
        if (xmlHttp.readyState == 4)

```

```

        {
            if(xmlHttp.status==200)
            {
                var text =
xmlHttp.responseText;

                HideObject("IMG_13","", 'infinite');

                store_score(text, chapter,
score); //text akan berubah menjadi variabel dom di
store_score

            }
            else
            {

HideObject("IMG_13","", 'infinite');

                alert("Please check your
internet connection");

            }
        }
    }
}

```

```

function store_score(dom, chapter, score) //dom diambil dari
variabel text

```

```

{
    var textfile = dom;
    var updatedStudentData = '';
    array = textfile.split(',');

    for(i=6;i<99;i++)
    {
        if (array[i] == chapter)
        {

```









```

        if (xmlHttp.readyState == 4)
            {
                if(xmlHttp.status==200)
                    {
                        var dom =
xmlHttp.responseText;

                        HideObject("IMG_13","", 'infinite');

                        findScore(dom);
                    }
                else
                    {
HideObject("IMG_13","", 'infinite');
                    }
            }
    }

function css() {
    var text = "";
        text += "<style>"+
            "table {background-color: lightblue; width: 80%;
font-size = 22px;}"+
            "th {text-align: left; }"+
            "td {background-color: white; color: black;}"+
            "</style>";
    return text;
}

function ShowAllData() //dipanggil di external url
{
    var url = url_data();

```

```

var Course = moduleID;
var passvar= "course_code=" + Course;
var xmlHttp = new XMLHttpRequest();
ShowObject("IMG_13","", 'infinite');
xmlHttp.open("POST", url, true);
xmlHttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xmlHttp.setRequestHeader("Content-length", "passvar.length ");
xmlHttp.setRequestHeader("Conection", "close");
xmlHttp.send(passvar);
xmlHttp.onreadystatechange = function()
{
    if (xmlHttp.readyState == 4)
    {
        if(xmlHttp.status==200)
        {
            var text =
xmlHttp.responseXML; //data masih dalam bentuk objek atau
xml
            var data = parseXML(text);
//dijadikan dalam bentuk text
            showAll(data);
//Menampilkan data
        }
        else
        {
            HideObject("IMG_13","", 'infinite');
            alert("Please check your internet
connection");
        }
    }
}
}

```

```

function parseXML(xml)
{
    var docoment = "";
    var tabel = xml.getElementsByTagName("coloumn");
    var informasi = "";
    var baris, kolom, score;
    for (baris = 0;baris < tabel.length; baris++){
        satubaris = tabel[baris];
        isibaris=satubaris.attributes;
        for (kolom =0;kolom<isibaris.length;kolom++){
            isikolom=isibaris[kolom].nodeValue;
            if (kolom>=(isibaris.length-1)){informasi +=
isikolom;}
                else informasi+=isikolom + ",";
            }
            informasi += ";";
        }
        return informasi;
    }

var number=0;

function showAll(data){ //langkah 1
    var array = data.split(';'); //memisahkan data yang di
dapat dari data yang di parse dari xml
    var text = "<body>" + css() + "<table
id='data'><th>No</th>"+
                "<th>NIM</th><th
width=100%>Nama</th><th>Phone</th><th>Pre-
Test</th><th>Quiz</th><th>01</th><th>02</th><th>03</th><th>04<
/th>"+
                "<th>05</th><th>06</th><th>07</th><th>08</th><th>09</th><
th>10</th><th>11</th>"+

```

"<th>12</th><th>13</th><th>14</th><th>15</th>"; //untuk mengatur tampilan di external url bagian atas

```
g_arVars['bufferHTML'] = text;
for (row=0;row<(array.length-1);row++){
    var col = array[row].split(',');
    var a =
getAllAssignment(col[1],col[2],array.length);
}
}
```

```
function getAllAssignment(NIM,phone,max) {
    var Course = moduleID;
    var url = url_send();
    var passvar= "user_id=" + NIM + "&" + "course_code=" +
Course;
    var xmlHttp = new XMLHttpRequest();
    xmlHttp.open("POST", url, true);
    xmlHttp.setRequestHeader("Content-type", "application/x-
www-form-urlencoded");
    xmlHttp.setRequestHeader("Content-length",
"passvar.length ");
    xmlHttp.setRequestHeader("Conection", "close");
    xmlHttp.send(passvar);
    xmlHttp.onreadystatechange = function()
    {
        if (xmlHttp.readyState == 4)
        {
            if(xmlHttp.status==200)
            {
                var dom =
xmlHttp.responseText;

                HideObject("IMG_13","", 'infinite');
```

```

                                                                    getScore (dom, phone, max);
//langkah 2
                                                                    }
                                                                    else
                                                                    {
HideObject("IMG_13", '', 'infinite');
                                                                    }
                                                                    }
                                                                    }
}

function startFrame(NIM) {
    var sNIM = NIM.toString();
    var nim = sNIM.substring(5);
    g_arVars['bufferNIM']=nim;
    getStudentReport(nim);
}

function getScore(dom, phone, max) {
    var chapters = "Basic HTML, Basic
Tags, Elements, Attributes, Formating, Comment, Images, " +
    "Tables, Lists, Blocks, Background, Colors, Fonts, Marquee, Layo
ut";
    var textfile = dom;
    var array = textfile.split(',');
    var nims =array[1];
//alert(nims);
    if (nims==undefined){nims=""};
    var name=array[2];
    if (name==undefined || nims== 51){name=""};
    var pretest=array[4];
    if (pretest==undefined || pretest== 51){pretest=""};

```

```

var quiz=array[6];
if (quiz==undefined || quiz== 51){quiz=""};

number++;
var nim = "12345"+nims;
g_arVars['bufferHTML'] +=      "<tr><td>" +number+ "</td>"
+
                                "<td><a href='#"
onClick='startFrame("+nim+")'>" +nims+"</a></td>" +
                                "<td>"      +name+
"</td>" +
                                "<td>" +phone+ "</td>"
+
                                "<td>" +pretest+
"</td>" +
                                "<td>" +quiz+ "</td>";
//digunakan untuk mengisi list no, nim, nama, nilai secara
otomatis

var chapter = chapters.split(','); //untuk mengambil
nilai sesuai dengan nama materi

for (c=0;c<chapter.length;c++){
    var i=7;
    var find = false;
    while (!find)
    {
        if (array[i] == chapter[c])
        {
            var score = array[i+2];
            g_arVars['bufferHTML'] +=
"<td>" +score+"</td>";
            find = true;
            break;
        }
        i++;
    }
}

```

```

        if (array[i] == "*" ) {break;}
        if (i==99) {break;}
    }
    if (!find) {g_arVars['bufferHTML'] += "<td></td>";}
}
g_arVars['bufferHTML'] += "</tr>";
if (number>=(max-1)) {
    g_arVars['bufferHTML'] += "</table></body>";
    TableScore.innerHTML = g_arVars['bufferHTML'];
    number=0;
}
}
}

```

```

function reportToExcel(tabel) {
    str = "";
    var mytable = document.getElementById(tabel);
    var rowCount = mytable.rows.length;
    var colCount =
mytable.getElementsByTagName("tr")[0].getElementsByTagName("th
").length;
    try{
        var ExcelApp = new
ActiveXObject("Excel.Application");
        catch(err){alert("Unsupport system");return;}
    var ExcelSheet = new ActiveXObject("Excel.Sheet");
    ExcelSheet.ActiveSheet.Columns("A").ColumnWidth = 4;
    ExcelSheet.ActiveSheet.Columns("B").ColumnWidth = 12;
    ExcelSheet.ActiveSheet.Columns("C").ColumnWidth = 25;
    ExcelSheet.ActiveSheet.Columns("D").ColumnWidth = 13;
    ExcelSheet.ActiveSheet.Columns("E").ColumnWidth = 3;
    ExcelSheet.ActiveSheet.Columns("F").ColumnWidth = 3;
    ExcelSheet.ActiveSheet.Columns("G").ColumnWidth = 3;
    ExcelSheet.ActiveSheet.Columns("H").ColumnWidth = 3;
}
}
}

```



```

ExcelSheet.ActiveSheet.Columns("I").ColumnWidth = 3;
ExcelSheet.ActiveSheet.Columns("J").ColumnWidth = 3;
ExcelSheet.ActiveSheet.Columns("K").ColumnWidth = 3;
ExcelSheet.ActiveSheet.Columns("L").ColumnWidth = 3;
ExcelSheet.ActiveSheet.Columns("M").ColumnWidth = 3;
ExcelSheet.ActiveSheet.Columns("N").ColumnWidth = 3;
ExcelSheet.ActiveSheet.Columns("O").ColumnWidth = 3;
ExcelSheet.ActiveSheet.Columns("P").ColumnWidth = 3;
ExcelSheet.ActiveSheet.Columns("Q").ColumnWidth = 3;
ExcelSheet.ActiveSheet.Columns("R").ColumnWidth = 3;
ExcelSheet.ActiveSheet.Columns("S").ColumnWidth = 3;
ExcelSheet.Application.Visible = true;
DisplayAlerts = true;
CollectGarbage();
for (var i = 0; i < rowCount; i++) {
    for (var j = 0; j < colCount; j++) {
        if (i == 0) {
            str =
mytable.getElementsByTagName("tr")[i].getElementsByTagName("th
") [j].innerText;
        }
        else {
            str =
mytable.getElementsByTagName("tr")[i].getElementsByTagName("td
") [j].innerText;
        }
        ExcelSheet.ActiveSheet.Cells(i + 1, j + 1).Value =
str;
    }
}
}

```

- **Program di Slide Registration**

```

HideObject('IMG_100');
var text = read_file();

```

```
array = text.split(',');

if(array[3] == 'VALID')
{
    OpenSlideById('Review Page');
}
```

- **Program di tombol *Login***

```
HideObject('IMG_100');
var text = read_file();

array = text.split(',');

if(array[3] == 'VALID')
{
    OpenSlideById('Review Page');
}
```

- **Program di *IFRAME***

```
var view = document.getElementById("TableScore");

view.style.background="white";
view.style.overflow="scroll";

ShowAllData();
```

- **Program di tombol *Send*** text =  
read\_file(); array =  
text.split(','); NIM = array[1];  
var Chat = g\_arVars['Chat']; if  
(Chat == undefined)  
{  
Chat=""; }  
Chat=""; }

```
text = g_arVars['textA']; var sendText =  
text.replace(", ", "$%$");
```

```
newText = Chat + "@M$$" + sendText + "#$%";  
sendDataPost(newText);
```

- **Program button save score**

```
var NIM = g_arVars['bufferNIM'];  
var chapter = "Basic HTML";  
var score = g_arVars['Score1'];

if (score!=""){
    saveScore(NIM, chapter, score);
}
else {alert("Please, enter the score")}
```

- **Program button show**

```
ShowObject("Home_title1","", "infinite");

var code = g_arVars['CodeArea'];

var title = "";
g_arVars['Title1'] = title;

if(code!=""){
var buffer = code.match("<title>(.*?)</title>");
if(buffer != null){title = buffer[1]}
g_arVars['Title1']= title;
}

var dom = document.getElementById('PageArea');

dom.innerHTML = code;
```

- **Program button refresh**

```
var dom = g_arVars['bufferHTML'];
var chapter = "Basic HTML";

uploadedAssignment(dom, chapter, 'CodeArea');
```

## **Lampiran 2 Listing Program**

- **Program di *module.js* Modul Mahasiswa**

```
// File name: Module.js
// Created: 01.12.2005
// Copyright (c) Websoft, 2006. All rights reserved.

// InitModule
function InitModule()
{
}

// ShutdownModule
function ShutdownModule()
{
```

```
}

var moduleID = 'TI 2016 UMY';//'HTML 2016 TI UMY';
var codeID = 68;
var codeIDH = 225;
var text_assign = '';

function url_update()
{
    var url_updt = "http://gflm.omy.ac.id/_insert_text.php";
    return url_updt;
}

function url_validation()
{
    var uurl = "http://gflm.omy.ac.id/_check_mhs.php";
    return uurl;
}

function url_send(){
    var uuurl="http://gflm.omy.ac.id/_send_onlinetext.php";
    return uuurl;
}

function url_data(){
    var surl="http://gflm.omy.ac.id/_data_mhs.php";
    return surl;
}

function callModuleID(){
    return moduleID;
}
```

```
function code(){
    return codeID;
}

function codeh(){
    return codeIDH;
}

function read_file()
{
    var Scr8 = new
ActiveXObject("Scripting.FileSystemObject");
    var moduleIDs = moduleID+".txt";
    try
    {
        var CTF8 = Scr8.OpenTextFile(moduleIDs, 1, true);
        encodedText = CTF8.ReadAll();
        CTF8.Close();
        decodedText = s_encode(encodedText,codeID);
        natureText = hexToString(decodedText);
        return natureText;

    }
    catch (err)
    {
        CTF8.Close();
        return 'Course ID is not found';
    }
}

function create_newFile(moduleID,studentId,studentName,status)
{
```

```

    var moduleIDs = moduleID + '.txt';
    var Scr2 = new
ActiveXObject("Scripting.FileSystemObject");
    var score = g_encode(score);
    var string = generate_array(100);
    array = string.split(',');

array[0]=moduleID;array[1]=studentId;array[2]=studentName;arra
y[3]=status;array[4]=0;array[5]=0;array[6]=0;
    var Text = array_toText(array,100);
    var HText = stringToHex(Text);
    secureText = s_encode(HText,codeID);
    try
    {
        var CTF2 = Scr2.CreateTextFile(moduleIDs);
        CTF2.Write(secureText);
        CTF2.Close();
        return "OK";
    }
catch (err)
    {
        CTF2.Close();
        alert('File saved error');
    }
}

function s_encode(str,code) {
    var encoded = "";
    for (i=0; i<str.length;i++) {
        var a = str.charCodeAt(i);
        var b = a ^ code;
        encoded = encoded+String.fromCharCode(b);
    }
}

```

```

        return encoded;
    }

function g_encode(grade) {
    return (grade+174)*100;
}

function g_decode(higrade) {
    return ((higrade)/100 -174);
}

var myArray = new Array();

function generate_array(number){
    var text = "*";
    for (i=1;i<number;i++){
        text = text + ',';
        myArray[i] = '*';
        text = text + myArray[i];
    }
    return text;
}

function array_toText(myArray,length){
    var string = myArray[0];
    for (i=1;i<length;i++){
        string = string + "," + myArray[i];
    }
    return string;
}

function stringToHex (tmp) {

```

```

var str = '',
    i = 0,
    tmp_len = tmp.length,
    c;

for (; i < tmp_len; i += 1) {
    c = tmp.charCodeAt(i);
    str += d2h(c) + ' ';
}
return str;
}

function hexToString (tmp) {
    var arr = tmp.split(' '),
        str = '',
        i = 0,
        arr_len = arr.length,
        c;

    for (; i < arr_len; i += 1) {
        c = String.fromCharCode( h2d( arr[i] ) );
        str += c;
    }
    return str;
}

function d2h(d) {
    return d.toString(16);
}

function h2d (h) {
    return parseInt(h, 16);
}

```



```

}

function validation(url, json, object)
{
    var xmlHttp = new XMLHttpRequest();
    var result = null;
    if (xmlHttp != null){
        ShowObject("IMG_13","", "infinite");
        xmlHttp.open("POST", url, true);
        xmlHttp.setRequestHeader("Content-
type","application/x-www-form-urlencoded");
        xmlHttp.setRequestHeader("Content-length",
"json.length");
        xmlHttp.setRequestHeader("Conection","close");
        xmlHttp.send(json);
        xmlHttp.onreadystatechange = function(){
            if ((xmlHttp.readyState == 4) & (xmlHttp.status ==
200)){
                var result = xmlHttp.responseText;
                if (result == "VALID"){
                    response = updateFile();
                    HideObject("IMG_13","", "infinite");
                    ShowObject('IMG_83');
                }
                if (result == "INVALID"){
                    alert("Please, check your Student-ID
and try again");
                    HideObject("IMG_83","", "infinite");
                    HideObject("IMG_13","", "infinite");
                }
            }
        }
        if ((xmlHttp.readyState == 4) & (xmlHttp.status !=
200)){

```

```
        alert("Validation error, please get internet
connection and try again");
        HideObject(object,"","infinite");
    }
}
}
}

function updateFile()
{
    var text = read_file();
    var array = text.split(',');
    var moduleIDs = array[0] + '.txt';
    var Scr2 = new
ActiveXObject("Scripting.FileSystemObject");
    array[3]="VALID";
    var Text = array_toText(array,array.length);
    var HText = stringToHex(Text);
    secureText = s_encode(HText,codeID);
    try
    {
        var CTF2 = Scr2.CreateTextFile(moduleIDs);
        CTF2.Write(secureText);
        CTF2.Close();
        return "Validation is done";
    }
catch (err)
{
    CTF2.Close();
    alert('File saved error');
}
}
```

```
function store_data(chapter, text)
{
    var dom = g_arVars['bufferDOM'];
    var textfile = read_file();
    array = textfile.split(',');
    var counter = 0;

    for(i=7;i<99;i++)
    {
        if ((array[i] == '*' ) | (array[i] == chapter))
        {
            counter = i;
            break;
        }
    }
    var quiz = GetObjectiveScore('total');

    var Textfilebaru = '';
    array[counter] = chapter;
    array[counter + 1] = text;
    array[counter + 2] = findScore(dom,chapter);

    for(i=0;i<array.length;i++){
        Textfilebaru += array[i] + ',';
    }

    var HText = stringToHex(Textfilebaru);
    secureText = s_encode(HText,codeID);
    var Scr6 = new
ActiveXObject("Scripting.FileSystemObject");
    var moduleIDs = moduleID + ".txt";
    try {
        var CTF6 = Scr6.CreateTextFile(moduleIDs);
```

```
        CTF6.Write(secureText);
        CTF6.Close();
            alert("Data saved");
        return "OK";
    }
    catch (err) {
        CTF6.Close();
        alert('File saved error');
    }
}

function refresh_quiz()
{
    var textfile = read_file();
    array = textfile.split(',');

    var Textfilebaru = '';

    if(array[5] == 0)
    {
        array[4] = 0;
    }
    else
    {
        array[6] = 0;
    }

    for(i=0;i<array.length;i++){
        Textfilebaru += array[i] + ',';
    }

    var HText = stringToHex(Textfilebaru);
```

```
secureText = s_encode(HText,codeID);
var Scr6 = new
ActiveXObject("Scripting.FileSystemObject");
var moduleIDs = moduleID + ".txt";
try {
    var CTF6 = Scr6.CreateTextFile(moduleIDs);
    CTF6.Write(secureText);
    CTF6.Close();
    return "OK";
}
catch (err) {
    CTF6.Close();
    alert('File saved error');
}
}
```

```
function store_quiz(quiz)
{
    var textfile = read_file();
    array = textfile.split(',');

    if(array[5] == 0)
    {
        array[4] = quiz;
    }
    else
    {
        array[6] = quiz;
    }

    var Textfilebaru = '';
```

```
for(i=0;i<array.length;i++){
    Textfilebaru += array[i] + ',';
}

var HText = stringToHex(Textfilebaru);
secureText = s_encode(HText,codeID);
var Scr6 = new
ActiveXObject("Scripting.FileSystemObject");
var moduleIDs = moduleID + ".txt";
try {
    var CTF6 = Scr6.CreateTextFile(moduleIDs);
    CTF6.Write(secureText);
    CTF6.Close();
    return "OK";
}
catch (err) {
    CTF6.Close();
    alert('File saved error');
}

}

function store_code()
{
    var textfile = read_file();
    array = textfile.split(',');

    var Textfilebaru = '';
    array[5] = 1;

    for(i=0;i<array.length;i++){
        Textfilebaru += array[i] + ',';
    }
}
```

```

    }

    var HText = stringToHex(Textfilebaru);
    secureText = s_encode(HText,codeID);
    var Scr6 = new
ActiveXObject("Scripting.FileSystemObject");
    var moduleIDs = moduleID + ".txt";
    try {
        var CTF6 = Scr6.CreateTextFile(moduleIDs);
        CTF6.Write(secureText);
        CTF6.Close();
        return "OK";
    }
    catch (err) {
        CTF6.Close();
        alert('File saved error');
    }
}

function getText(NIM)
{
    // alert(NIM);
    var Course = moduleID;
    var url = url_send();
    var passvar= "user_id=" + NIM + "&" + "course_code=" +
Course;
    var xmlHttp = new XMLHttpRequest();
    ShowObject("IMG_13","", 'infinite');
    xmlHttp.open("POST", url, true);
    xmlHttp.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");
    xmlHttp.setRequestHeader("Content-length", "passvar.length
");

```

```

xmlHttp.setRequestHeader("Conection", "close");
xmlHttp.send(passvar);
xmlHttp.onreadystatechange = function()
{
    if (xmlHttp.readyState == 4)
    {
        if(xmlHttp.status==200)
        {
            var text = xmlHttp.responseText;
            var cek = text.split(',');

            if (cek[4] != 0 || cek[5] != 0 || cek[6] != 0 ||
cek [7] != '*')
            {
                var moduleIDs = Course + '.txt';
                var Scr2 = new
ActiveXObject("Scripting.FileSystemObject");

                var score = g_encode(score);
                var string = generate_array(100);
                array = string.split(',');

                array[0]=cek[0];array[1]=cek[1];array[2]=cek[2];array[3]=
cek[3];array[4]=cek[4];array[5]=cek[5];array[6]=cek[6];array[7
]=cek[7];array[8]=cek[8];array[9]=cek[9];array[10]=cek[10];arr
ay[11]=cek[11];array[12]=cek[12];array[13]=cek[13];array[14]=c
ek[14];array[15]=cek[15];array[16]=cek[16];array[17]=cek[17];a
rray[18]=cek[18];array[19]=cek[19];array[20]=cek[20];

                array[21]=cek[21];array[22]=cek[22];array[23]=cek[23];arr
ay[24]=cek[24];array[25]=cek[25];array[26]=cek[26];array[27]=c
ek[27];array[28]=cek[28];array[29]=cek[29];array[30]=cek[30];a
rray[31]=cek[31];array[32]=cek[32];array[33]=cek[33];array[34]
=cek[34];array[35]=cek[35];array[36]=cek[36];array[37]=cek[37]
;array[38]=cek[38];array[39]=cek[39];array[40]=cek[40];

                array[41]=cek[41];array[42]=cek[42];array[43]=cek[43];arr

```



```
ay[44]=cek[44];array[45]=cek[45];array[46]=cek[46];array[47]=c
ek[47];array[48]=cek[48];array[49]=cek[49];array[50]=cek[50];a
rray[51]=cek[51];array[52]=cek[52];array[53]=cek[53];array[54]
=cek[54];array[55]=cek[55];array[56]=cek[56];array[57]=cek[57]
;array[58]=cek[58];array[59]=cek[59];array[60]=cek[60];

    var hsl = array_toText(array,100);
    var HText = stringToHex(hsl);
    secureText = s_encode(HText,codeID);
    try
    {
        var CTF2 = Scr2.CreateTextFile(moduleIDs);
        CTF2.Write(secureText);
        CTF2.Close();
        alert('WELCOME');
        HideObject("IMG_13","", "infinite");
        //OpenSlideById('SLIDE_1');
    }
    catch (err)
    {
        CTF2.Close();

        alert('File saved error');
    }
    //alert(string);

}
else
{
    HideObject("IMG_13","", "infinite");
}
}

else
{
```

```

        HideObject("IMG_13",'','infinite');
    }
}

function sendDataPost(){
    var message = "IMG_86";
    var url = url_update();
    var json = read_file();
    var xmlhttp = new XMLHttpRequest();
    var passvar = "onlinetext="+json;
    if (xmlhttp != null){
        ShowObject(message,'','infinite');
        xmlhttp.open("POST", url, true);
        xmlhttp.setRequestHeader("Content-
type", "application/x-www-form-urlencoded");
        xmlhttp.setRequestHeader("Content-length",
"passvar.length");
        xmlhttp.setRequestHeader("Conection", "close");
        xmlhttp.send(passvar);
        xmlhttp.onreadystatechange = function(){
            if (xmlhttp.readyState == 4){
                if (xmlhttp.status == 200){
                    HideObject(message,'','infinite');
                }
                else {
                    HideObject(message,'','infinite');
                }
            }
        }
        else {
            HideObject(message,'','infinite');
        }
    }
}

```

```

        }
    }
}

function assignment(chapter)
{
    var result = "";
    var textfile = read_file();
    var array = textfile.split(',');
    var i=0;
    var counter = "true";
    while (counter=="true")
    {
        if (array[i]==chapter)
        {
            var text = array[i+1];
            break;
        }
        i++;
        if (array[i]=="*"){break;}
        if (i==99){break;}
    }
    return text;
}

```

```

function htmlCode(text){
    var buffer = "";
    var loop = "TRUE"
    while (loop){
        buffer = text.replace(',','@@@');
        if (buffer == text) {break;}
        text = buffer;
    }
}

```

```
    }  
    return buffer;  
}
```

```
function htmlDeCode(text){  
    var buffer = "";  
    var loop = "TRUE"  
    while (loop){  
        buffer = text.replace('@@@',' ','');  
        if (buffer == text) {break;}  
        text = buffer;  
    }  
    return buffer;  
}
```

```
function getStudentReport (NIM)  
{  
    var Course = moduleID;  
    var url = url_send();  
    var passvar= "user_id=" + NIM + "&" + "course_code=" +  
Course;  
    var xmlHttp = new XMLHttpRequest();  
    ShowObject("IMG_13",'', 'infinite');  
    xmlHttp.open("POST", url, true);  
    xmlHttp.setRequestHeader("Content-type", "application/x-  
www-form-urlencoded");  
    xmlHttp.setRequestHeader("Content-length",  
"passvar.length ");  
    xmlHttp.setRequestHeader("Conection", "close");  
    xmlHttp.send(passvar);  
    xmlHttp.onreadystatechange = function()  
    {  
        if (xmlHttp.readyState == 4)
```

```

        {
            if(xmlHttp.status==200)
            {
                var text =
xmlHttp.responseText;
                g_arVars['bufferDOM']=text;
            }
            else
            {
HideObject("IMG_13","", 'infinite');
            }
        }
    }
}

```

```

function findScore(dom,chapter)
{
    var textfile = dom;
var array = textfile.split(',');
    var score=0;
    var i=0;
    var counter = true;
    while (counter)
    {
        if (array[i] == chapter)
        {
            score = array[i+2];
            counter=false;
            break;
        }
        i++;
        if (array[i]=="*"){break;}
    }
}

```

```
        if (i==99){break;}
    }
    return score;
}

function read_js()
{
    var ScrJs = new
ActiveXObject("Scripting.FileSystemObject");
    var moduleIDs = "1/module.js";
    try
    {
        var CTFJs = ScrJs.OpenTextFile(moduleIDs, 1,
true);

        var codeJs = CTFJs.ReadAll();
        CTFJs.Close();
        return codeJs;
    }
    catch (err)
    {
        CTFJs.Close();
        alert(err);
        return 'Error reading';
    }
}

function createJs(code)
{
    var Scr = new
ActiveXObject("Scripting.FileSystemObject");
    var path = Scr.GetAbsolutePathName(".");
    g_arVars['GPath']= path;
    var moduleIDs = "project.js";
```

```
try
{
    var CTF = Scr.CreateTextFile(moduleIDs);
    CTF.Write(code);
    CTF.Close();
    return "OK";
}
catch (err)
{
    CTF.Close();
    alert('File saved error');
}
}

function require(script) {
    $.ajax({
        url: script,
        dataType: "script",
        async: false,
        success: function () {
        },
        error: function () {
            throw new Error("Could not load script " +
script);
        }
    });
}

function get_script(html){
    var clear = html.replace(/(\r\n\t|\n|\r\t)/gm, "");
    var script = "";
    var cont = true;
    while (cont){
```

```

        var scr = clear.match("<script>(.*?)</script>");
        if (scr!=null){
            script += " " + scr[1];
            var deletestring = "<script>" + scr[1] +
"</script>";
            var newClear = clear.replace(deletestring, '');
            clear = newClear;
        }
        else {break;}
        //alert(clear);
    }
    return script;
}

```

- **Program button Register Now**

```

var NIM = g_arVars['inputNIM'];
var Name = g_arVars['inputNAME'];
var url = url_update();
var moduleID = callModuleID();
var uurl = url_validation();
if (NIM != '' )
{
    var status =
create_newFile(moduleID,NIM,Name, 'INVALID');
    if (status == 'OK')
    {
        alert('New data is saved');
        var text = read_file();
        var array = text.split(',');
        var json = "username="+array[1];
        validation(uurl, json, 'IMG_83');
        HideObject("IMG_85","", "infinite");
        HideObject("inputNIM","", "infinite");
        HideObject("inputNAME","", "infinite");
        HideObject("TXT_54","", "infinite");
        HideObject("TXT_53","", "infinite");
        HideObject("TXT_57","", "infinite");
    }
    else {alert ("Error!");}
}
else
{
    alert('Enter NIM and First Name');
}

```



```
var data = read_file();
validation(url, data, "IMG_83");
getText(NIM);
```

- **Program button Show**

```
ShowObject("Home_title1","", "infinite");
var code = g_arVars['CodeArea'];

var title = "";
g_arVars['Title1'] = title;

if(code!=""){
var buffer = code.match("<title>(.*?)</title>");
if(buffer != null){title = buffer[1]}
g_arVars['Title1']= title;
}
var dom = document.getElementById('PageArea');

dom.innerHTML = code;
```

- **Program button Save**

```
var text = g_arVars['CodeArea'];

text = s_encode(text,codeh());
textcoded = htmlCode(text);
store_data('Basic HTML', textcoded);
sendDataPost();
```

- **Program button Submit Quiz**

```
refresh_quiz();

var a = GetObjectiveScore('total');
store_quiz(a);
sendDataPost();
```

- **Program di Slide Registration**

```
var text = read_file();
array = text.split(',');

if(array[3] == 'VALID')
{
    g_arVars['bufferNIM']=array[1];
```

```

        OpenSlideById('SLIDE_1');
    }

```

- **Program di Slide Assignment**

```

var text = read_file();
var array = text.split(',');
var NIM = array[1];
var name = array[2];
var title = document.title;
document.title = title + "          " + NIM + "          " +
name;

getStudentReport (NIM);

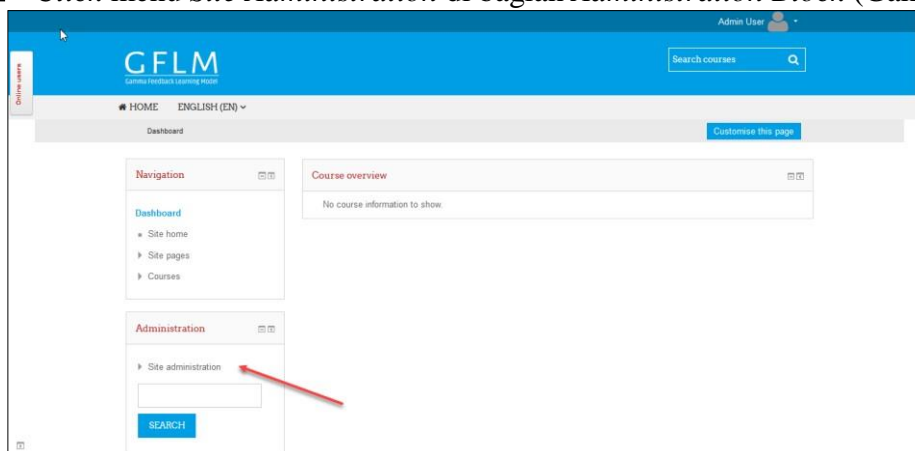
```

### Lampiran 3 GFLM Moodle

- **Langkah pertama: Membuat *Category***

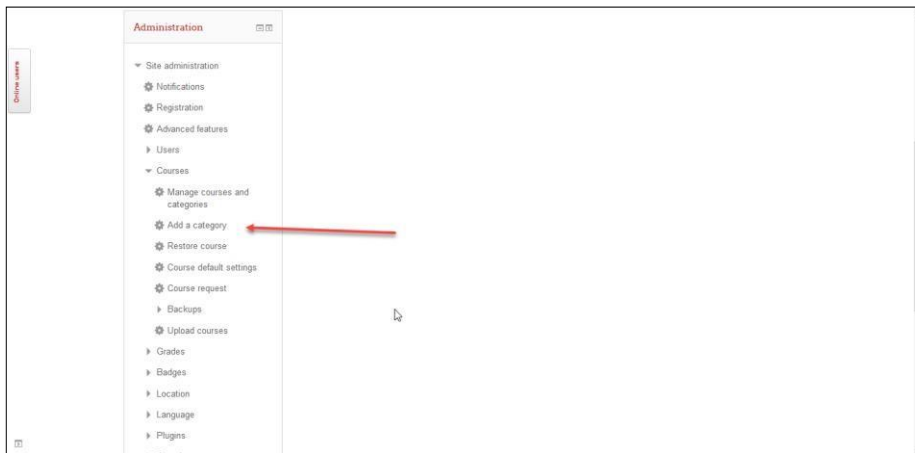
Langkah pertama adalah *Category setting*, langkah ini diperlukan untuk menyiapkan *Moodle* supaya dapat digunakan beberapa program studi.

- Login sebagai *Administrator*
- Click* menu *Home* pada *toolbar menu*
- Click* menu *Site Administration* di bagian *Administration Block* (Gambar 1)



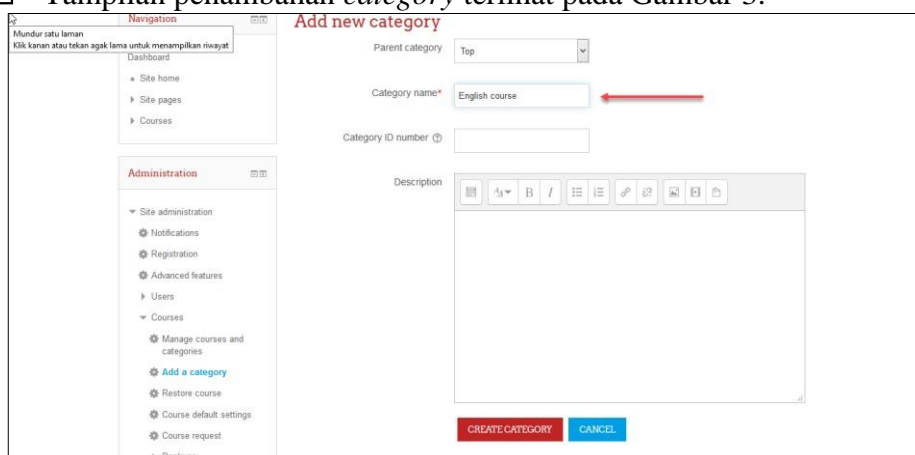
**Gambar 1** *Administration Block*

- Pilih menu *New Category*

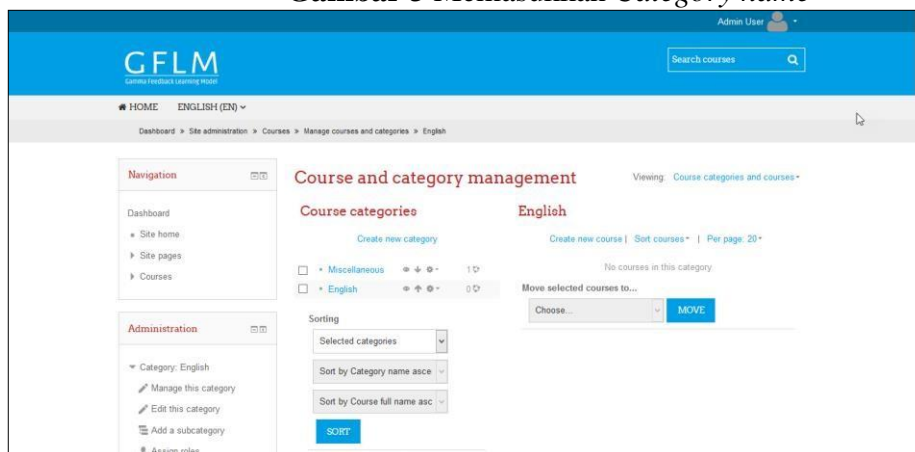


**Gambar 2** Pilih menu untuk membuat *Category*

- Masukkan nama Program Studi sebagai *Category name* dan klik tombol *Create Category* di sisi bawah warna merah (Gambar 2). Jika direncanakan Moodle hanya untuk satu program studi, maka kategori digunakan sebagai nama blok kuliah. Misalkan, blok teori, blok praktikum, blok tugas akhir, dll.
- Tampilan penambahan *category* terlihat pada Gambar 3.



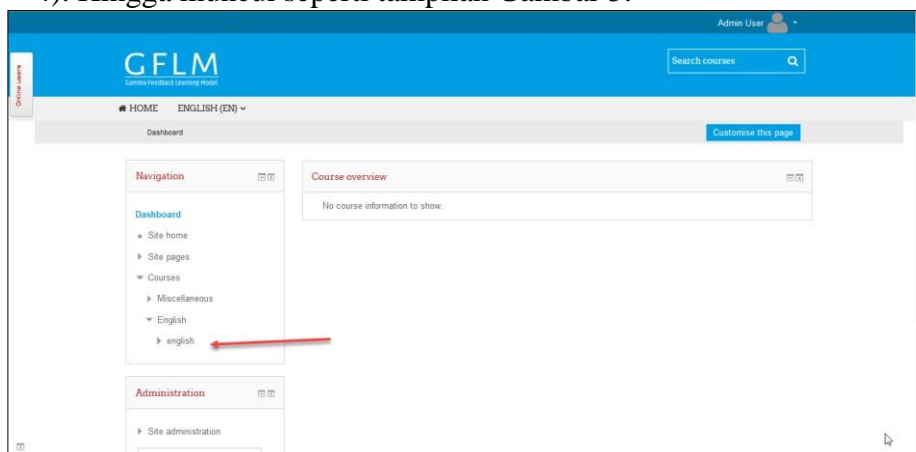
**Gambar 3** Memasukkan *Category name*



**Gambar 4** Kategori *English* berhasil dibuat

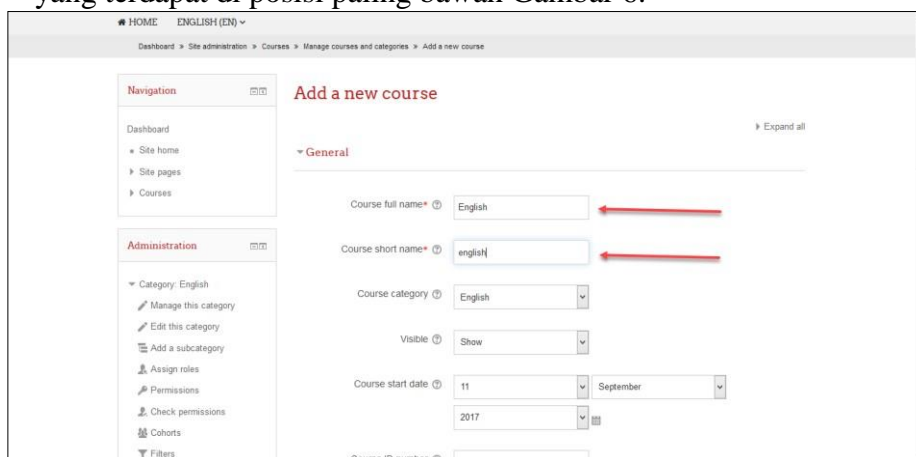
Selesai membuat *Category*, dilanjutkan membuat matakuliah.  
Langkah-langkah:

- Pada *Dashboard block* klik *Courses*., klik *Category*, kemudian *Click English* (Gambar 4). Hingga muncul seperti tampilan Gambar 5.

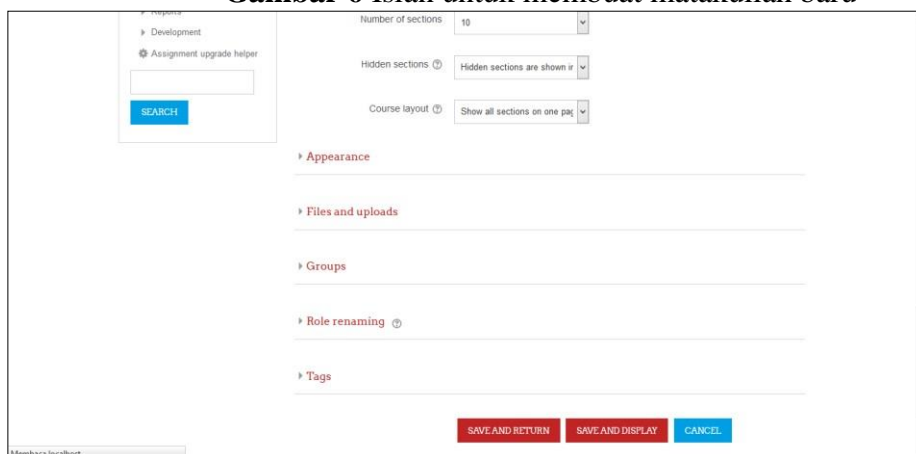


**Gambar 5** Link untuk membuat matakuliah pada kategori tertentu

- Setelah tampil seperti Gambar 5, di *Input Form* ketik nama matakuliah, dalam contoh ini nama matakuliah adalah “*english*”, sama dengan nama kategorinya. Setelah pengisian data cukup, data disimpan dengan menekan tombol “*SAVE AND RETURN*” yang terdapat di posisi paling bawah Gambar 6.



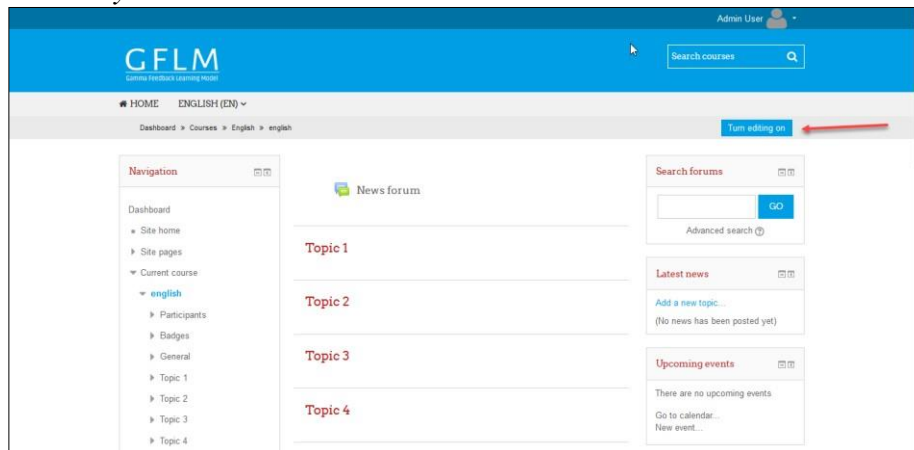
**Gambar 6** Isian untuk membuat matakuliah baru



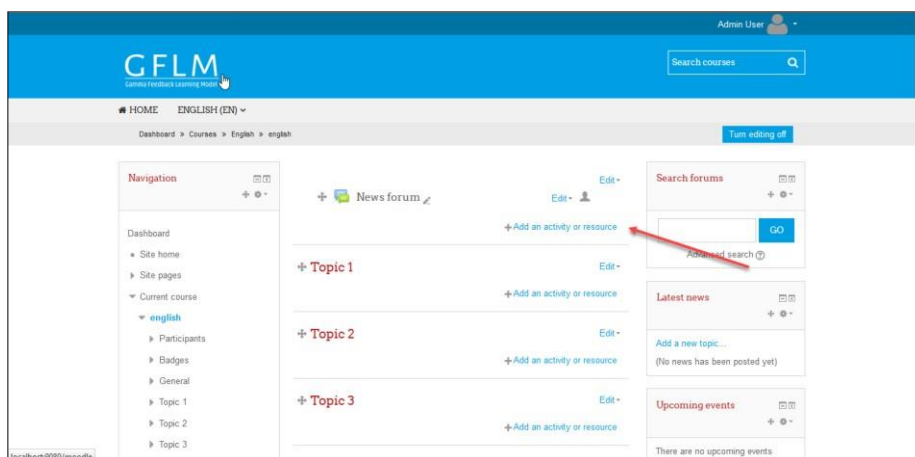
**Gambar 7** Tombol *Save and Return* di posisi paling bawah

□ **Langkah kedua: Membuat Assignment**

- Dimulai dari *Dashboard*, click *Site Administration*, klik *Courses*, dan klik *english* maka akan muncul tampilan seperti tampak pada Gambar 7. Tampilan ini menunjukkan sedang berada di ruang kuliah maya “ENGLISH”.
- Kemudian tombol *Turn edit on* (diisi atas kanan warna biru) ditekan sehingga antarmuka berubah menjadi Gambar 8. Pada layar terlihat ada tampilan *link* untuk *Add activity or resources*.



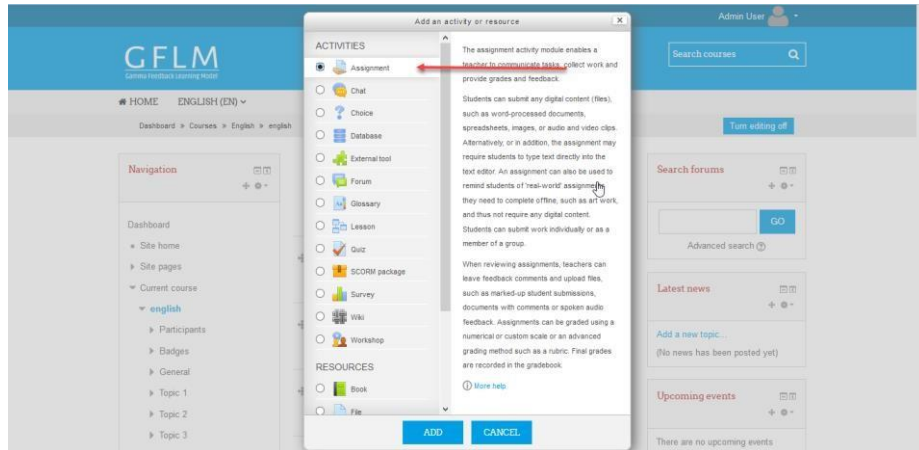
**Gambar 8** Tampilan untuk membuat *Assignment*



**Gambar 9** Tampilan antarmuka matakuliah pada *mode editing on*

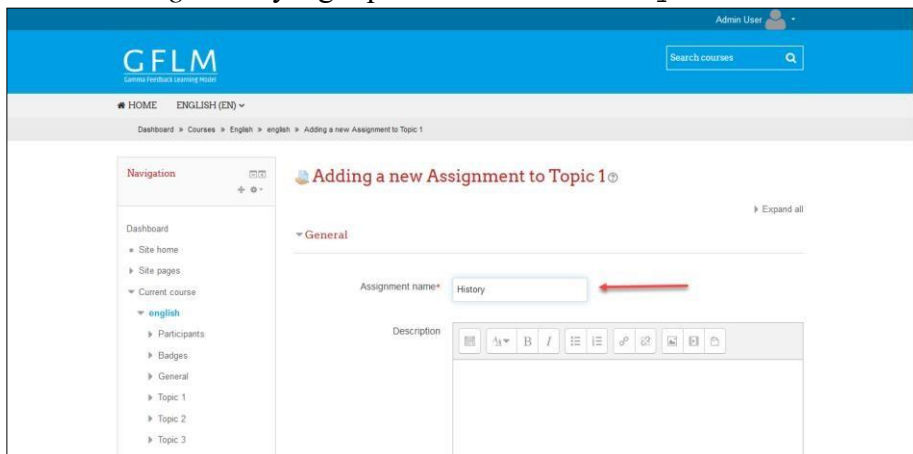
- Langkah berikutnya, menekan tombol *Add on activity or resource* sehingga muncul tampilan seperti Gambar 9. □ Kemudian klik *Assignment*.

*Assignment* ini dipilih karena menyesuaikan dengan data yang akan dikirim dari modul *CAI* yang berbentuk teks.



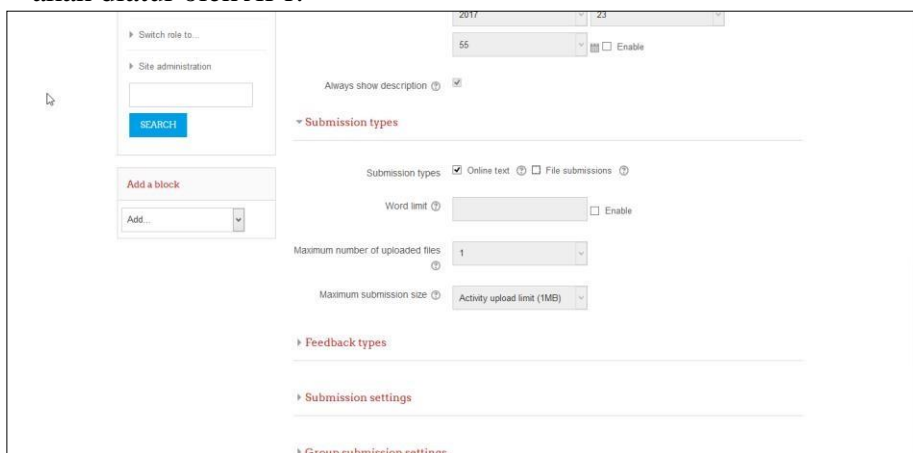
**Gambar 10** Tampilan *activity* dengan pilihan *Assignment*

- Kemudian tombol *ADD* (warna biru) ditekan sehingga muncul tampilan seperti Gambar 10.
- Nama *Assignment* yang dipilih adalah “History”.



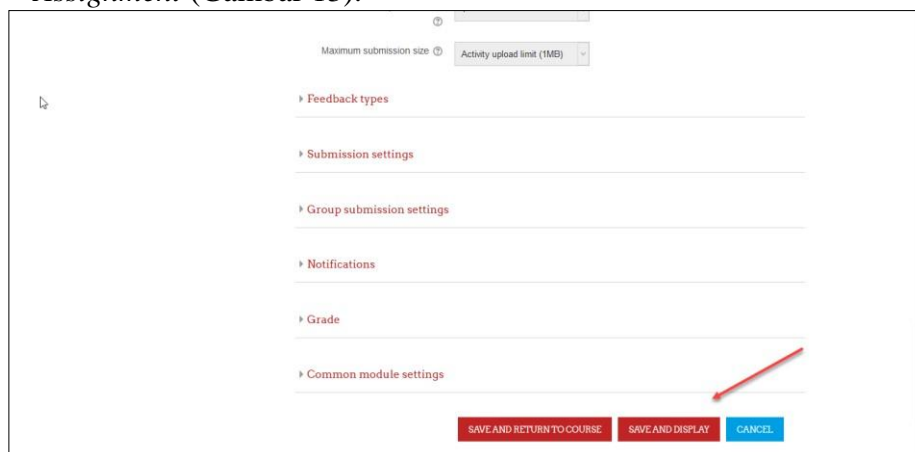
**Gambar 11** Tampilan untuk memasukkan nama *Assignment*

- Pada Gambar 11 terdapat pilihan cara pengiriman *Assignment*, *online* atau *submission* (unggah *file* dokumen). Dipilih cara *online* karena data teks dikirim secara *online* yang akan diatur oleh *API*.



**Gambar 12** Pilihan cara pengiriman *online* atau *upload file*

- Diisi paling bawah terdapat tombol “SAVE AND DISPLAY” untuk menyimpan *Assignment* (Gambar 13).



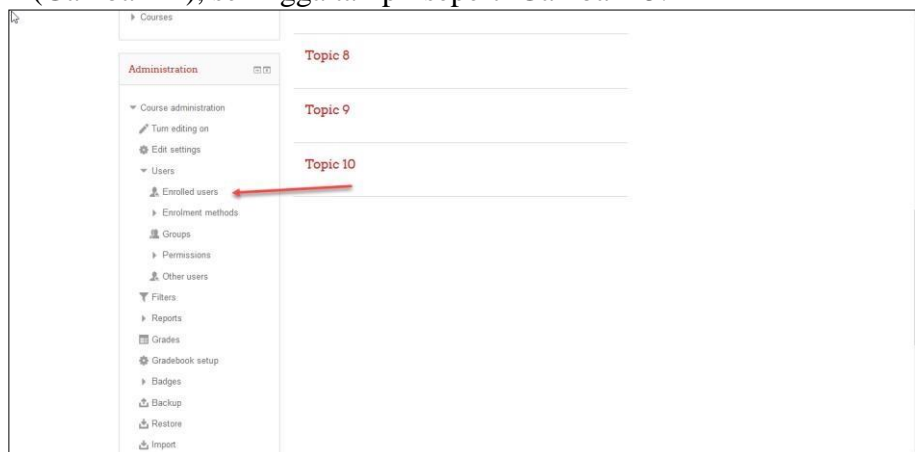
**Gambar 13** Tombol untuk menyimpan *Assignment*

Langkah berikutnya adalah mendaftarkan mahasiswa ke dalam matakuliah. Tatacara *upload* data peserta matakuliah dapat dilihat pada buku panduan *LMS Moodle*.

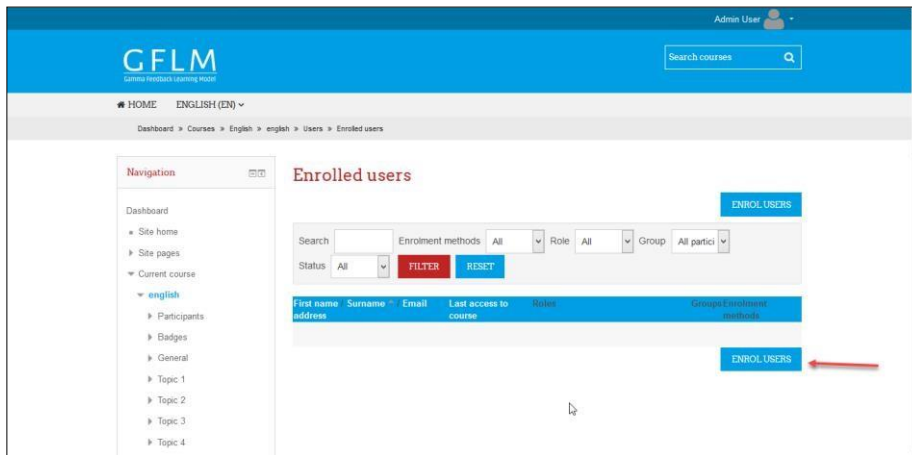
### □ **Langkah ketiga: *Student enrollment***

Pendaftaran mahasiswa sebagai peserta matakuliah berguna sebagai *trigger* untuk tabel *Assignment*. Langkah-langkah:

- Klik *Home*, pada *Navigation block* klik *Course*
- Jika sudah muncul *Course Administration*, klik *Users* kemudian klik *Enrolled users* (Gambar 14), sehingga tampil seperti Gambar 15.

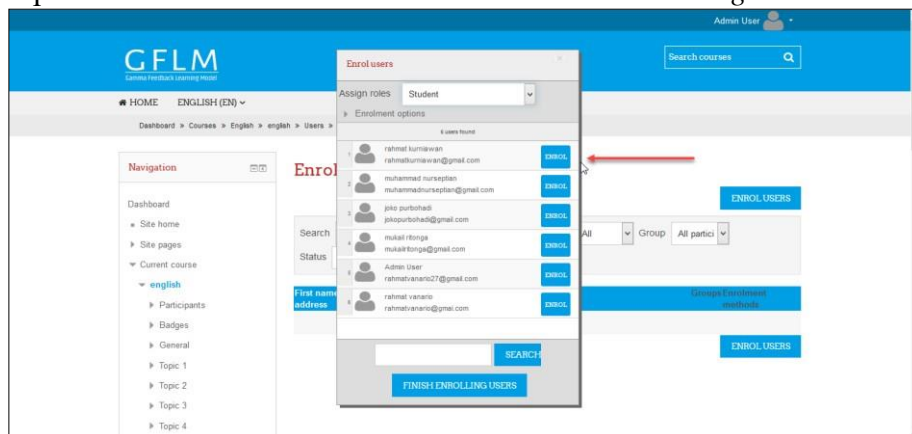


**Gambar 14** Antarmuka untuk *Enrolled users*

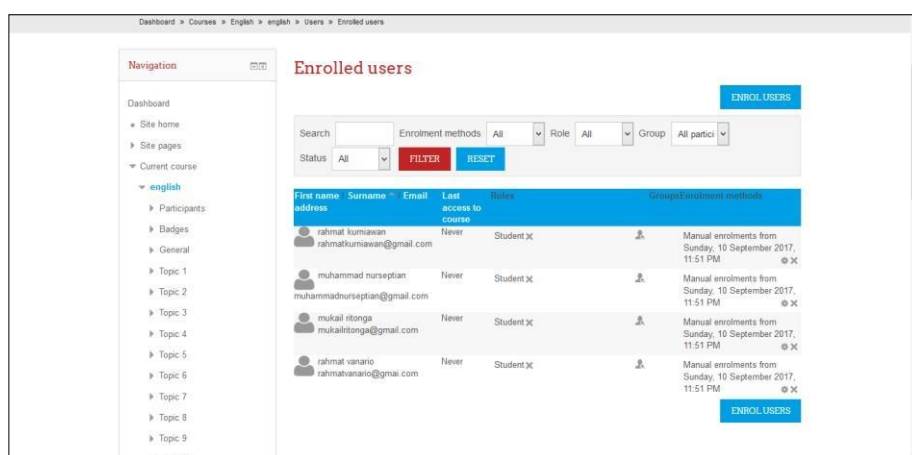


**Gambar 15** Antarmuka untuk mendaftarkan mahasiswa

- Klik tombol “*Enrol user*” sehingga muncul tampilan seperti Gambar 16. Pilih mahasiswa yang ingin didaftarkan ke matakuliah “*ENGLISH*” dengan cara klik pada tombol “*Enroll*”. Kemudian klik “*Finish enrolling user*”.



**Gambar 16** Mendaftarkan mahasiswa pada matakuliah



**Gambar 17** Mahasiswa peserta matakuliah *English*

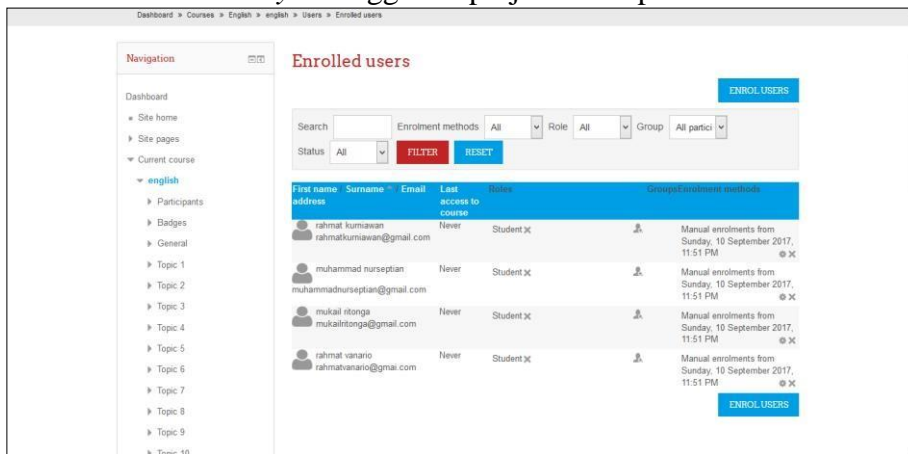
Gambar 17 memperlihatkan ada sembilan mahasiswa yang telah didaftarkan pada matakuliah English. Berikutnya memberi *trigger* untuk tabel *Assignment*.



□ **Langkah keempat: Menginisiasi *Assignment data***

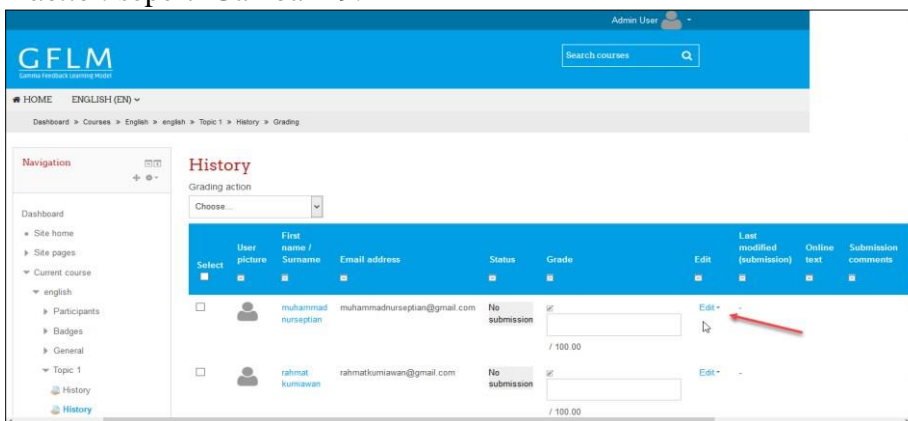
Langkah ini diperlukan untuk inisiasi nilai tugas. Seperti telah dibahas sebelumnya, bahwa *Assignment* telah ditetapkan sebagai tugas mahasiswa yang dikumpulkan secara *online*. Dalam sistem *Extended-CAI*, tugas dikirim melalui modul *CAI* yang dijalankan pada komputer atau piranti *mobile* milik mahasiswa. Tugas tersebut dikirim juga dalam bentuk teks. Namun tugas dapat dikirim setelah terlebih dahulu dilakukan proses inisiasi dengan mengisi tugas setiap mahasiswa dengan teks, minimalnya sebuah karakter (bukan spasi). Selain nilai tugas juga perlu diinisiasi dengan mengisi angka minimal 0. Langkah ini hanya dapat dilakukan oleh dosen dan *administrator*.

- Pada *Navigation block*, klik *Current Course*, klik *English*, kemudian klik *Topic 1*, dan terakhir klik *History* sehingga tampil jendela seperti Gambar 18.



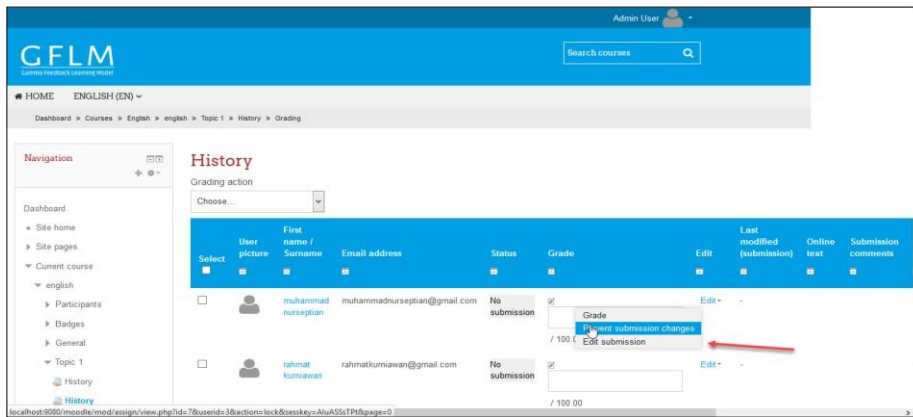
**Gambar 18** Antarmuka untuk menampilkan teks terkirim *CAI*

- Pada antarmuka *History*, klik link *View grade all submissions* sehingga tampil *Grade action* seperti Gambar 19.



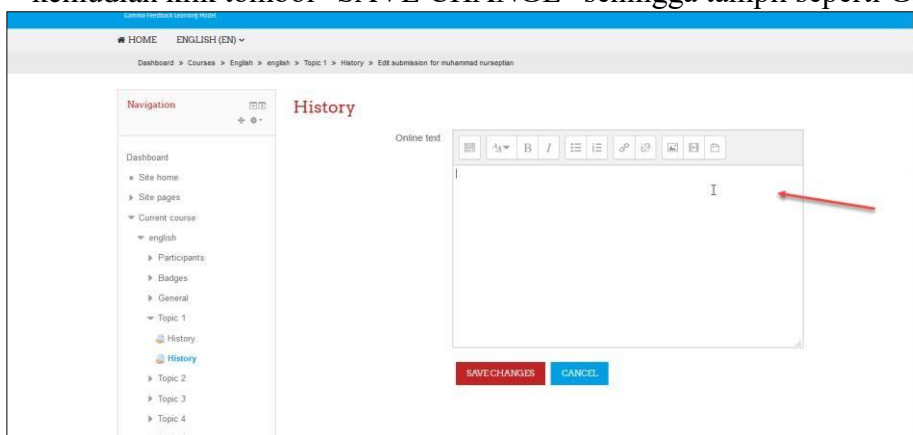
**Gambar 19** Tampilan antarmuka *Grading Action*

- Dari antarmuka *Grade Action*, klik menu *Edit* sehingga tampil seperti Gambar 20.

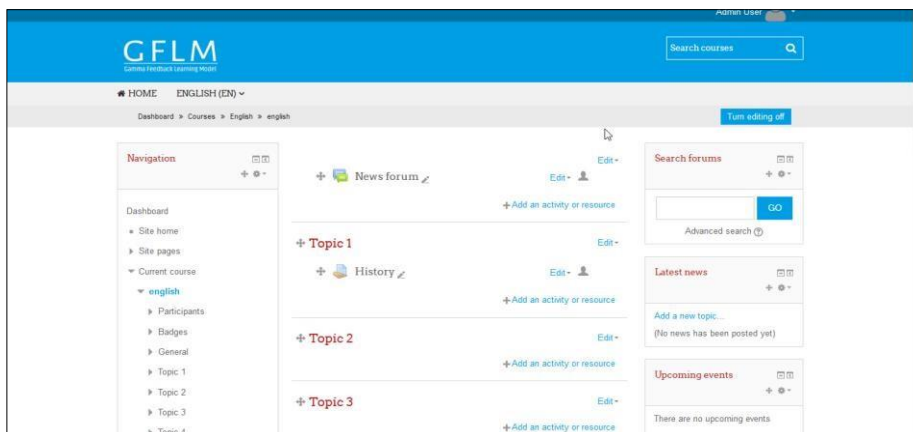


**Gambar 20** Edit tugas secara *online*

- Masukkan sembarang *character* pada *text box* (Gambar 20), minimal 1 buah *character*, kemudian klik tombol “SAVE CHANGE” sehingga tampil seperti Gambar 21.



**Gambar 21** Inisiasi teks tugas secara *online*



**Gambar 22** Tampilan akhir matakuliah “English”

Gambar 22 memperlihatkan tampilan akhir matakuliah “English” berikut sebuah *Assignment* yang diberi nama “History”.