

LAMPIRAN

Lampiran 1 Listing Program

Program di module.js Mahasiswa

```
// InitModule
function InitModule()
{
}

// ShutdownModule
function ShutdownModule()
{
}

function NoID(){
    return "20140140003";
}

function ModulID(){
    return "Sistem Operasi 2018";
}

function status(){
    return "VALID";
}

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

```

function url_validation()
{
    var uurl = "http://gflm.ummy.ac.id/_check_mhs.php";
    //var uurl = "http://gflm.ummy.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(){
    var id = "Sistem Operasi 2018";
    return id;
}

var codeID = 68;
var moduleID = 'Sistem Operasi 2018';

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;array[3]=sta
tus;

    var Text = array_toText(array,100);
    var HText = stringToHex(Text);
    secureText = s_encode(HText,codeID);
    try
        {
            var CTF2 = Scr2.CreateTextFile(moduleIDs);

```

```

        CTF2.Write(secureText);fs
        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);
}

function generate_array(number){

```

```

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

var myArray = new Array();

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 = readyAJAX();
    var result = null;
    if (xmlHttp != null)
    {
        ShowObject(object,"","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();
                alert(response);
                HideObject(object,"","infinite");
                ShowObject('OBJ_5');
                NextSlide();
            }
        if (result === "INVALID"){
            alert("Please, check your Student-ID and try again");
            HideObject(object,"","infinite");
        }
    }
    if ((xmlHttp.readyState == 4) & (xmlHttp.status != 200)){
        alert("Validation error, please get internet connection and
try again");
        HideObject(object,"","infinite");
    }
}
}
}
}

```

```

new XMLHttpRequest();

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 readyAJAX()
{
    try {
        return new XMLHttpRequest();
    } catch(e) {

```



```

    try {
        return new ActiveXObject("Msxml2.XMLHTTP");
    } catch(e) {
        try {
            return new ActiveXObject("Microsoft.XMLHTTP");
        } catch(e) {
            return "A newer browser is needed.";
        }
    }
}
}
/*
function url_send(){
    var url_update = "http://gflm.ummy.ac.id/_insert_text.php";
    return url_update;
}

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

function url_check(){
    var url_n = "http://gflm.ummy.ac.id/_check_mhs.php";
    return url_n;
}*/

function nim(){
    return NIM;
}

```

```

function checkMhs(NIM){
    var url_NIM = url_check();
    var passvar= "username="+NIM;

    var url = "http://gflm.umy.ac.id/_check_mhs.php";
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.open("POST", url_NIM, 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 resp = xmlhttp.responseText;
                if(resp == "VALID"){
                    alert ("Login
Berhasil")
                    NextSlide();}
                else { alert("NIM Tidak Dikenal,
ulang");}
            }
            else
            {
                alert("PENYUSUP");
            }
        }
    }
}

```

```

function sendDataPost(text)
{
    var url = url_send();
    var json = ModulID()+","+NIM+","+status() + ","+text;
    var xmlhttp = readyAJAX();
    var passvar = "onlinetext="+json;
    if (xmlhttp != null)
    {
        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)
                {
                    getStudentChat(NIM);
                }
                else
                {
                    alert("No Internet Connection");
                }
            }
        }
    }
}

```

```

}

function getStudentChat(NIM)
{
    var Course = ModulID();
    var htmlChat = '';
    var url = url_get();
    var passvar= "user_id=" + NIM + "&" + "course_code=" + Course;
    var xmlhttp = readyAJAX();
    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 array = text.split(",");
                var chat = array[3];
                g_arVars['Chat'] = chat;
                if (chat != undefined)
                {
                    htmlChat = newsDecode(chat);
                }
                //alert(chat);
                var view = document.getElementById('Chatbox');

```

```

        view.style.background = "lightgray";
        view.style.overflow = 'scroll';
        view.innerHTML = "<HTML>" +htmlChat+"</HTML>";
        view.scrollTop = 999999;

        // view.innerHTML = newsDecode (chat);
        // down = setTimeout(bottom, 100);
            }
else
        {
            alert("No internet connection");
        }
    }
}
}
}

```

```

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

```

```

function newsDecode(text){
    var buffer = "";
    var newText1 = "";
    var newText2 = "";
    var newText3 = "";
    var newText4 = "";
    var newText5 = "";
    var loop = "TRUE";

    while (loop){
        buffer = text.replace('%$',',');
        if (buffer == text) {break;}
        text = buffer;
    }
    newText1=text;
    var loop = "TRUE";
    while (loop){
        buffer = newText1.replace('#$', '<br>');
        if (buffer == newText1) {break;}
        newText1 = buffer;
    }
    newText2=newText1;
    var loop = "TRUE";
    while (loop){
        buffer = newText2.replace("@D$$", "<font color='red'><p
style='font-family:Verdana;', align = left><font size='5'>");
        if (buffer == newText2) {break;}
        newText2 = buffer;
    }
    newText3=newText2;
    var loop = "TRUE";

```

```

while (loop){
    buffer = newText3.replace("$@@","</font>");
    if (buffer == newText3) {break;}
        newText3 = buffer;
}
newText4=newText3;
var loop = "TRUE";
while (loop){
    buffer = newText4.replace("@M$$","<font color='blue'><p
style='font-family:Verdana;', align = right><font size='5'>");
    if (buffer == newText4) {break;}
        newText4 = buffer;
}
newText5=newText4;
var loop = "TRUE";
while (loop){
    buffer = newText5.replace("$$@", "</font>");
    if (buffer == newText5) {break;}
        newText5 = buffer;
}
return newText5;
}

```

```

function chsword(a, b){
    textD = "";
    textM = "";
    var inputDosen = new Array();
    var inputMhs = new Array();
    var inputD = new Array();
    var inputM = new Array();
    var hasil;
    var total = [];

```

```

var bufferrem = '';
    var bufferremNilai = 0;
    var nilaiJarakArray = 0;
var reference = 0;
inputDosen = a.split(".");
inputMhs = b.split(".");
buffer = [];
    buffer2 = [];

    lebihDosen = inputDosen.length - inputMhs.length;
    lebihMhs = inputMhs.length - inputDosen.length;

    if(inputMhs.length < inputDosen.length){
        for(i = 0; i < inputDosen.length; i++){
            //Menambahkan array yang kurang di jawaban Mahasiswa
            for(h = 0; h < lebihDosen; h++){
                inputMhs.push(inputDosen[i]);
            }
            //Menghitung jarak terbesar setiap array
            for(j = 0; j < inputMhs.length; j++){
                hasil = levenshtein(inputDosen[i],
inputMhs[j]);
                nilaiJarakArray += hasil;
            }
            total[i] = nilaiJarakArray;
            nilaiJarakArray = 0;
            //Menghapus array yang di tambahkan ke jawaban
mahasiswa
            for(k = 0; k < lebihDosen; k++){
                inputMhs.pop();
            }
        }
        //alert(total);

```



```

//Mengurutkan
var len = inputDosen.length;
for (var i = len-1; i>=0; i--){
    for(var j = 1; j<=i; j++){
        if(total[j-1]>total[j]){
            var tempNilai = total[j-1];
            total[j-1] = total[j];
            total[j] = tempNilai;

            var temp = inputDosen[j-1];
            inputDosen[j-1] = inputDosen[j];
            inputDosen[j] = temp;
        }
    }
}
for(i = 0; i < lebihDosen; i++){
    //delete inputDosen[i];
    inputDosen.pop();
}
//alert(inputDosen + " dor");
for(i=0;i<inputDosen.length;i++){
    textD += inputDosen[i] + ".";
    textM += inputMhs[i] + ".";
}
//alert(textD + " pisah " + textM);
}
else if(inputDosen.length < inputMhs.length){
    for(i = 0; i < inputMhs.length; i++){
        //Menambahkan array yang kurang di jawaban Dosen
        for(h = 0; h < lebihMhs; h++){
            inputDosen.push(inputMhs[i]);
        }
    }
}

```

```

    }
    //Menghitung jarak terbesar setiap array
    for(j = 0; j < inputDosen.length; j++){
        hasil = levenshtein(inputDosen[j],
inputMhs[i]);
        nilaiJarakArray += hasil;
    }
    total[i] = nilaiJarakArray;
    nilaiJarakArray = 0;
    //Menghapus array yang di tambahkan ke jawaban
mahasiswa
    for(k = 0; k < lebihMhs; k++){
        inputDosen.pop();
    }
}
//Mengurutkan
var len = inputMhs.length;
for (var i = len-1; i>=0; i--){
    for(var j = 1; j<=i; j++){
        if(total[j-1]>total[j]){
            var tempNilai = total[j-1];
            total[j-1] = total[j];
            total[j] = tempNilai;

            var temp = inputMhs[j-1];
            inputMhs[j-1] = inputMhs[j];
            inputMhs[j] = temp;
        }
    }
}
for(i = 0; i < lebihMhs; i++){
    //delete inputDosen[i];

```

```

        inputMhs.pop();
    }
    for(i=0;i<inputMhs.length;i++){
        textD += inputDosen[i] + ".";
        textM += inputMhs[i] + ".";
    }
    //alert(textD + " pisah " + textM);
}
else if(inputDosen.length == inputMhs.length){
    for(i=0;i<inputDosen.length;i++){
        textD += inputDosen[i] + ".";
        textM += inputMhs[i] + ".";
    }
}
return [textD, textM];
}

```

```

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

```

```

function score(a, b, c){

```

```

jumArray = c.length - b.length;

arrayKunci = c.split(".");
arrayD = a.split(".");
arrayM = b.split(".");
arrayD.pop();
arrayM.pop();

var total = 0;
var hasilScore = 0;
var totalkarakter = maksdist(c);

for(d=0; d < arrayD.length; d++){
    mindist = 100;
    //for(m=0; m < arrayD.length; m++){
        dist = levenshtein(arrayD[d], arrayM[d]);
        if(dist<mindist){
            mindist = dist;
        }
        total += dist;
    //}
}
var score = ((totalkarakter - total)/totalkarakter)*100;
    if(score < 0){
        res = score * score;
        nilai = 100 - res;
        nilaiAkhir = nilai.toFixed(0);
    }
    else if(b.length < jumArray){
        nilai = 100 - score
        nilaiAkhir = nilai.toFixed(0);
    }

```

```

    }
    else{
        nilai = score;
        nilaiAkhir = nilai.toFixed(0);
    }
    return nilaiAkhir;
}

```

```

function levenshtein(a, b) {
    if(a.length === 0) return b.length;
    if(b.length === 0) return a.length;

    var matrix = [];

    // increment along the first column of each row
    var i;
    for(i = 0; i <= b.length; i++){
        matrix[i] = [i];
    }

    // increment each column in the first row
    var j;
    for(j = 0; j <= a.length; j++){
        matrix[0][j] = j;
    }

    // Fill in the rest of the matrix
    for(i = 1; i <= b.length; i++){
        for(j = 1; j <= a.length; j++){
            if(b.charAt(i-1) == a.charAt(j-1)){
                matrix[i][j] = matrix[i-1][j-1];
            }
        }
    }
}

```

```

    } else {
        matrix[i][j] = Math.min(matrix[i-1][j-1] + 1, // substitution
                                Math.min(matrix[i][j-1] + 1, // insertion
                                           matrix[i-1][j] + 1)); // deletion
    }
}
}
return matrix[b.length][a.length];
}

```

```

function mainSetence(words){
    if (words!=''){
        var otext = "";
        var wArray = words.split(' ');
        var aLength = wArray.length;
        for (w=0;w<aLength;w++){
            var itext = wArray[w];
            itext = itext.replace(/\s/g, '');
            itext = itext.toLowerCase();
            var ichar = itext.substr(0,1);
            var gchar = find_base(itext,ichar);
            if (gchar==""){otext+=itext + ' '}
            else {otext+=gchar + ' '}
        }
    }
    return otext;
}

```

```

function firstFunction(_callback){
    // do some asynchronous work
    // and when the asynchronous stuff is complete

```

```

        _callback();
    }
function secondFunction(){
    // call first function and pass in a callback function which
    // first function runs when it has completed
    firstFunction(function() {
        console.log('huzzah, I\'m done!');
    });
}

function read_text()
{
    var ScrText = new ActiveXObject("Scripting.FileSystemObject");
    var currpath = ScrText.GetAbsolutePathName(".");
    var moduleIDs = currpath.concat("\\kata.txt");
    try
    {
        var CTFText = ScrText.OpenTextFile(moduleIDs, 1, false);
        var text = CTFText.ReadAll();
        CTFText.Close();
    }
    catch (err)
    {
        CTFText.Close();
        alert(err);
    }
    return text;
}

function save_text(text,filename)
{

```

```

var Scr = new ActiveXObject("Scripting.FileSystemObject");
var moduleIDs = "var.txt";
try
{
    var CTF = Scr.CreateTextFile(moduleIDs);
    CTF.Write(text);
    CTF.Close();
    return "OK";
}
catch (err)
{
    CTF.Close();
    alert('File saved error');
}
}

```

```

function get_array(text){
    var clear = text.replace(/(\r\n\t|\n|\r\t)/gm,"");
    var json='';
    var array = [];
    var counter=0;
    var cont = true;
    while (cont){
        var start_pos = clear.indexOf('(') + 1;
        var end_pos = clear.indexOf(')',start_pos);
        var scr = clear.substring(start_pos,end_pos);
        array = scr.split(',');
        if ((scr!=null)|(scr!=undefined)|(scr!='')){
            var deletestring = "(" + scr + ")," ;
            var newClear = clear.replace(deletestring,'');
            if (clear==newClear){break;}
        }
    }
}

```



```

        if (counter<4){
            json += array[1]+' '+array[2]+' ';
            counter++;
        }else{
            json += array[1]+' '+array[2]+' '+' '+'\n';
            json+=' ';
            counter=0;
        }
        clear = newClear;
        clear = clear.replace(/(\r\n\t|\n|\r\t)/gm, "");
    }
    else {break;}
}
json = remcom(json);
return json + ' ';
}

```

```

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

```

```

function dasar_akhiran(text){
    var result = "";
    var buffer = "";

```

```

var vari = text.substring(0,1);
var load = "";
try{
    load = window[vari]();
} catch (e){result=text;return result;}
var textL = text.length;
var row = load.split(';');
var dist = 100;
var found = false;
for (rb=0;rb<row.length;rb++){
    word = row[rb].split(',');
    word[0]=word[0].replace(/\s+/, "");
    if (word[0]==text){
        result = word[0];//+', '+word[1];
        found = true;
        break;
    }
    if ((!found)&&(rb==(row.length-1))){
        tLength=text.length;
        if ((text.substring(tLength-1)=="i")&(text.substring(tLength-4)!="wati")){
            text=text.substring(0,tLength-1);
            result = text;
            found = true;
            break;
        };
        if ((text.substring(tLength-2)=="an")&(text.substring(tLength-3)!="wan")
            &(text.substring(tLength-3)!="kan")){
            text=text.substring(0,tLength-2);
            result = text;
            found = true;

```

```

        break;
    };
    if (text.substring(tLength-3)=="kan"){
        text=text.substring(0,tLength-3);
result = text;
        found = true;
        break;
    };
    if (text.substring(tLength-2)=="ku"){
        text=text.substring(0,tLength-2);
result = text;
        found = true;
        break;
    };
    if (text.substring(tLength-2)=="mu"){
        text=text.substring(0,tLength-2);
result = text;
        found = true;
        break;
    };
    if (text.substring(tLength-3)=="nya"){
        text=text.substring(0,tLength-3);
result = text;
        found = true;
        break;
    };
    if (text.substring(tLength-3)=="kah"){
        text=text.substring(0,tLength-3);
result = text;
        found = true;
        break;
    };

```

```

};
if (text.substring(tLength-3)== "pun"){
    text=text.substring(0,tLength-3);
result = text;
    found = true;
    break;
};
if (text.substring(tLength-3)== "lah"){
    text=text.substring(0,tLength-3);
result = text;
    found = true;
    break;
};
if (text.substring(tLength-3)== "tah"){
    text=text.substring(0,tLength-3);
result = text;
    found = true;
    break;
};
if (text.substring(tLength-4)== "wati"){
    text=text.substring(0,tLength-4);
result = text;
    found = true;
    break;
};
if (text.substring(tLength-3)== "wan"){
    text=text.substring(0,tLength-3);
result = text;
    found = true;
    break;
};

```

```

        if (text.substring(tLength-3)=="man"){
            text=text.substring(0,tLength-3);
        result = text;
            found = true;
            break;
        };
    }
}
return result;
}

```

```

function find_base(text,vari){
    var result = "tidak ketemu";
    var buffer = "";
    try{
        load = window[vari]();
    } catch (e){result=text;return result;}
    var textL = text.length;
    var row = load.split(';');
    var dist = 100;
    var found = false;
    for (rb=0;rb<row.length;rb++){
        word = row[rb].split(',');
        word[0]=word[0].replace(/\s+/, "");
        if (word[0]===text){
            result = word[0];//+', '+word[1];
            found = true;
            break;
        }
    }
    if ((!found)&&(rb==(row.length-1))){
        if (text.substring(0,2)=="se"){

```

```

        text=text.substring(2);
        var fChar = text.substring(0,1);
var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    };
    if (text.substring(0,3)=="per"){
        text=text.substring(3);
        var fChar = text.substring(0,1);
var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    };
    if (text.substring(0,3)=="ter"){
        text=text.substring(3);
        var fChar = text.substring(0,1);
var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    };
    if
((text.substring(0,2)=="be")&(text.substring(0,3)!="ber")){
        text=text.substring(2);
        var fChar = text.substring(0,1);
var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    };
    if (text.substring(0,3)=="ber"){
        text=text.substring(3);
        var fChar = text.substring(0,1);
var dasar = dasar_akhiran(text);

```

```

        result = find(dasar,fChar);
        break;
};
if (text.substring(0,2)=="me"){
    if (text.substring(2,4)=="mb"){
        text=text.substring(3);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if (text.substring(2,4)=="mu"){
        text="p"+text.substring(3);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if
((text.substring(2,4)=="ru")|(text.substring(2,4)=="ra")){
        text="r"+text.substring(3);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if
((text.substring(2,5)=="maa")|(text.substring(2,5)=="mak")){
        text=text.substring(2);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);

```

```

        break;
    }
    if (text.substring(2,4)=="ns"){
        text=text.substring(3);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if
((text.substring(2,4)=="ni")|(text.substring(2,4)=="ne")){
        text="t"+text.substring(3);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if (text.substring(2,4)=="li"){
        text="l"+text.substring(3);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if (text.substring(2,6)=="mper"){
        text=text.substring(6);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if (text.substring(2,6)=="mpe1"){

```



```

        text=text.substring(6);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if
((text.substring(2,5)=="mpr")|(text.substring(2,5)=="mpe")){
        text=text.substring(3);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if (text.substring(2,5)=="nas"){
        text="n"+text.substring(3);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if
((text.substring(2,4)=="nu")|(text.substring(2,4)=="na"
|(text.substring(2,4)=="no")){
        text= "t" + text.substring(3);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if (text.substring(2,5)=="nya"){
        text="s"+text.substring(4);

```

```

        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if
    (((text.substring(2,4)=="ny")&(text.substring(2,5)!="nya"))
    |(text.substring(2,6)=="nyad")|(text.substring(2,6)=="nyan")){
        text= "s" + text.substring(4);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if
    ((text.substring(2,6)=="nyan")|(text.substring(2,6)=="manj"))
    |(text.substring(2,5)=="maj")|(text.substring(2,6)=="mand")){
        text= text.substring(2);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if
    ((text.substring(2,4)=="nj")|(text.substring(2,4)=="nz"))
    |(text.substring(2,4)=="nd")|(text.substring(2,4)=="nc"))
    |(text.substring(2,5)=="ndr")){
        text=text.substring(3);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);

```

```

        break;
    }
    if ((text.substring(2,6)=="nget")){
        text=text.substring(5);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if (text.substring(2,8)=="ngemuk"){
        text=text.substring(5);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }
    if
((text.substring(2,5)=="ngh")|(text.substring(2,6)=="ngak")
    |(text.substring(2,5)=="ngg")|(text.substring(2,7)=="nging")|
    (text.substring(2,5)=="ngi")|
    ((text.substring(2,5)=="nga")&(text.substring(2,6)!="ngag")&(text
.substring(2,7)!="ngand")))
    |((text.substring(2,5)=="ngo")&(text.substring(2,6)!="ngor"))|(te
xt.substring(2,5)=="ngu")){
        text=text.substring(4);
        var fChar = text.substring(0,1);
        var dasar = dasar_akhiran(text);
        result = find(dasar,fChar);
        break;
    }

```

```

        if
((text.substring(2,5)== "nga")|(text.substring(2,5)== "ngi")&((text.subs
tring(2,7)!= "nging")))

        |(text.substring(2,5)== "ngr")|(text.substring(2,5)== "nge")|(text.
substring(2,6)!= "nget")

        |(text.substring(2,6)== "ngag")|(text.substring(2,7)== "ngand")|(te
xt.substring(2,6)== "ngor")){

                text="k"+text.substring(4);
                var fChar = text.substring(0,1);
                var dasar = dasar_akhiran(text);
                result = find(dasar,fChar);
                break;
        }

        if
((text.substring(2,4)== "ma")&(text.substring(2,5)!= "mak")){

                text="p" + text.substring(3);
                var fChar = text.substring(0,1);
                result = find(text,fChar);
                break;
        }

        else {

                text=text.substring(2);
                var fChar = text.substring(0,1);
                result = find(text,fChar);
                break;
        }

        };

        // if
((text.substring(0,2)== "ke")&(text.substring(2,4)!= "ked")&(text.substr
ing(2,4)!= "kedu")){

                // text = dasar_akhiran(text);
                // var fChar = text.substring(0,1);
                // var dasar = dasar_akhiran(text);

```

```

        // result = find(dasar,fChar);
        // break;
    // }
    if (text.substring(0,2)=="ke"){
        if (text.substring(2,4)=="mb"){
            text="kem" + text.substring(3);
            var fchar = text.substring(0,1);
            result = find(text,fchar);
            break;
        };
        if (text.substring(2,4)=="rj"){
            text="ker".substring(3);
            var fchar = text.substring(0,1);
            result = find(text,fchar);
            break;
        };
        if (text.substring(2,4)=="na"){
            text="ken".substring(3);
            var fchar = text.substring(0,1);
            result = find(text,fchar);
            break;
        };
        if
        ((text.substring(2,4)=="du")|(text.substring(2,4)=="ti")|(text.substri
        ng(2,4)=="em")

        |(text.substring(2,4)=="li")|(text.substring(2,4)=="en")|(text.su
        bstring(2,4)=="tu")

        |(text.substring(2,4)=="de")|(text.substring(2,4)=="se")){
            text=text.substring(2);
            var fchar = text.substring(0,1);
            result = find(text,fchar);

```

```

        break;
    }
    // else{
        // text = dasar_akhiran(text);
        // var fChar = text.substring(0,1);
        // var dasar = dasar_akhiran(text);
        // result = find(dasar,fChar);
        // break;
    // }
};
// if (text.substring(0,2)=="ke"){
    // text=text.substring(2);
    // var fchar = text.substring(0,1);
// result = find(text,fchar);
    // break;
// };
if (text.substring(0,2)=="di"){
    text=text.substring(2);
    var fChar = text.substring(0,1);
    var dasar = dasar_akhiran(text);
result = find(dasar,fChar);
    break;
};
else {
    text = dasar_akhiran(text);
    var fChar = text.substring(0,1);
    var dasar = dasar_akhiran(text);
result = find(dasar,fChar);
    break;
}
}

```

```

        if (found){break;}
    }
    return result;
};

function find(text,vari){
    var result = "tidak ketemu";
    var buffer = "";
    try{
        load = window[vari]();
    } catch (e){result=text;}
    var textL = text.length;
    var row = load.split(';');
    var dist = 100;
    var found = false;
    for (rb=0;rb<row.length;rb++){
        word = row[rb].split(',');
        word[0]=word[0].replace(/\s+/, "");
        if (word[0]==text){
            result = word[0];//+', '+word[1];
            found = true;
            break;
        }
        if ((!found)&&(rb==(row.length-1))){
            found=false;
            for (col=1;col<text.length;col++){
                findText = text.substr(0, text.length-col);
                for (nr=0;nr<row.length;nr++){
                    word = row[nr].split(',');
                    word[0]=word[0]
                    .replace(/\s+/, "");

```

```

        if (word[0].length>=findText.length){
            var compareText =
word[0].substr(0, findText.length);
            if
((compareText==findText)&&!found)){
                result =
word[0];//+', '+word[1];
                found=true;
                break;
            }
        }
        if(found){break;}
    }
    if(found){break;}
}
if(found){break;}
}
if (found){break;}
}
return result;
};

```

```

function compare(strA,strB){
    for(var result = 0, i = strA.length; i--;){
        if(typeof strB[i] == 'undefined' || strA[i] == strB[i]);
        else if(strA[i].toLowerCase() == strB[i].toLowerCase())
            result++;
        else
            result += 4;
    }
    return 1 - (result + 4*Math.abs(strA.length -
strB.length))/(2*(strA.length+strB.length));
}

```


- **Program di *Slide Login***

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

//alert(array[3]);

if(array[3] == 'VALID')
{
    OpenSlideById('SLIDE_28');
}
```

- **Program di tombol *Login***

```
//var nim = g_arVars['input_nim'];
//checkMhs(nim);

var NIM = g_arVars['nim'];
var Name = g_arVars['nama'];
//var PAS = g_arVars['pas'];
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,'OBJ_5');
    }
}
else
{
    alert('Enter NIM and First Name');
}
var data = read_file();
validation(url, data, "OBJ_5");
```

- **Program di tombol *Jawab***

```
var text = g_arVars['inputDosen'];
var text2 = g_arVars['inputMhs'];

//var result = mainSetence(text);
//var result2 = mainSetence(text2);
```

```

//g_arVars['hasilDosen']=result;
//g_arVars['hasilMhs']=result2;

var hasil = chsword(text, text2);

var result = mainSetence(hasil[0]);
var result2 = mainSetence(hasil[1]);

g_arVars['hasilDosen']=result;
g_arVars['hasilMhs']=result2;

alert(text2);

var akhir = score(result, result2, text);
//var akhir = score(hasil[0], hasil[1]);
//var akhir = score(text, text2);

text2 = s_encode(text2,codeh());
textcoded = htmlCode(text2);

if(text2 != ""){
alert(akhir);
store_data('Sistem Operasi', textcoded, akhir);

sendDataPost();
}
else
{
var s = 0;
alert(s);
store_data('Sistem Operasi', textcoded, s);

sendDataPost();
}

```

- **Program di *module.js* Dosen**

```

// InitModule
function InitModule()
{
}

// ShutdownModule
function ShutdownModule()
{
}

var moduleID = 'Sistem Operasi 2018';
var codeID = 68;
var codeIDH = 225;
var text_assign = '';

function url_update()
{

```

```

    var url_updt = "http://gflm.ummy.ac.id/_insert_text.php";
    return url_updt;
}

function url_validation()
{
    var uurl = "http://gflm.ummy.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;array[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;

    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){
                if (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(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)
{
  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 {
                    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 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 += "</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);
                        }
                    else
                        {
                            HideObject("IMG_13",'', 'infinite');
                            alert("Please check your
internet connection");
                        }
                }
        }
}

function store_score(dom, chapter, score)
{
    var textfile = dom;
    var updatedStudentData = '';
    array = textfile.split(',');
    for(i=4;i<99;i++)
    {
        if (array[i] == chapter)
        {
            array[i+2] = score;
            break;
        }
    }

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

```

```

    }
    sendScoredData(updatedStudentData);
}

function sendScoredData(data){
    var message = "IMG_86";
    var url = url_update();
    var json = data;
    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');
                    alert("Score sent");
                }
                else {
                    HideObject(message, '', 'infinite');
                }
            }
            else {
                HideObject(message, '', 'infinite');
            }
        }
    }
}

function getStudentData()
{
    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.responseText;
            HideObject("IMG_13",'', 'infinite');
            ViewerData.innerHTML =
parsXML(text);
        }
        else
        {
HideObject("IMG_13",'', 'infinite');
            alert("Please check your internet
connection");
        }
    }
}

function parsXML(xml)
{
    var document = "";
    var tabel = xml.getElementsByTagName("coloumn");
    var informasi = "<body>" + css() + "<table
id='data'>"+"<tr>Student List<tr>"+"<th>No</th>"
"<th>Nama</th><th>NIM</th><th>Telepon</th><th>Assignment</th>";
    var baris, kolom, score;
    for (baris = 0;baris < tabel.length; baris++){
        informasi = informasi + "<tr><td>" + (baris+1) + "</td>";
        satubaris = tabel[baris];
        isibaris=satubaris.attributes;
        for (kolom =0;kolom<isibaris.length;kolom++){
            isikolom=isibaris[kolom].nodeValue;
            informasi=informasi + "<td>" + isikolom + "</td>";
            var sNIM = isibaris[1].nodeValue;
        }
        informasi += "<td><a href='#'
onClick='startFrame("+sNIM+")'>Assignment</a></td>";
        informasi += "<td><a href='#'
onClick='getAllScore("+sNIM+")'>Score<a/></td>";
        informasi += "</tr>";
    }
    return informasi+"</body>";
}

function getAllScore(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);
}

```

```

xmlHttpRequest.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");
xmlHttpRequest.setRequestHeader("Content-length", "passvar.length ");
xmlHttpRequest.setRequestHeader("Connection", "close");
xmlHttpRequest.send(passvar);
xmlHttpRequest.onreadystatechange = function()
{
    if (xmlHttpRequest.readyState == 4)
    {
        if(xmlHttpRequest.status==200)
        {
            var dom =
xmlHttpRequest.responseText;

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

function findScore(dom){
    var chapters = "Sistem Operasi, Migrating, Elements,
Attributes, Formating, Comment, Images, Tables" +
"Lists, Blocks, Background, Colors,
Fonts, Marquee, Layout";
    var textfile = dom;
    var array = textfile.split(',');
    var text = "<body>" + css() +
"<table>"+<tr>"+array[2]+<tr>"+<th>No</th>"+
"Chapter</th><th>Score</th>";
    var chapter = chapters.split(',');
    for (c=0;c<chapter.length;c++){
        var i=0;
        var counter = "true";
        while (counter == "true")
        {
            if (array[i] == chapter[c])
            {
                var score = array[i+2];
                text +=
"<tr><td>"+(c+1)+"</td><td>"+chapter[c]+"</td><td>"+score+"</td></t
r>";
                break;
            }
            i++;
            if (array[i]=="*"){break;}
            if (i==99){break;}
        }
    }
}

```

```

        text += "</body></table>";
        TableScore.innerHTML=text;
    }

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()
{
    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;
                                //alert(text);
                                var data = parseXML(text);
                                alert(data);
                                showAll(data);
                        }
                    else
                        {
                            HideObject("IMG_13",'', 'infinite');
                            alert("Please check your internet
connection");
                        }
                }
        }
}

function parseXML(xml)
{

```

```

    var document = "";
    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){
    var array = data.split(';');
    var text = "<body>" + css() + "<table id='data'><th>No</th>" +

        "<th>NIM</th><th>Nama</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>";

    g_arVars['bufferHTML'] = text;
    for (row=0;row<(array.length-1);row++){
        var col = array[row].split(',');
        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);
        }
        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 = "Sistem
Operasi,Migrating,Elements,Attributes,Formating,Comment,Images," +
    "Tables,Lists,Blocks,Background,Colors,Fonts,Marquee,Layout";
    var textfile = dom;
    var array = textfile.split(',');
    var nims =array[1];
    if (nims==undefined){nims=""};
    var name=array[2];
    if (name==undefined){name=""};
    var a = "";
    number++;
    var nim = "12345"+nims;
    g_arVars['bufferHTML'] +=      "<tr><td>" +number+ "</td>" +
                                   "<td><a href='#"
onClick='startFrame("+nim+")'>"+nims+"</a></td>" +
                                   "<td>"      +name+
"</td>" ;

    var chapter = chapters.split(',');
    for (c=0;c<chapter.length;c++){
        var i=4;
        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 *IFRAME***

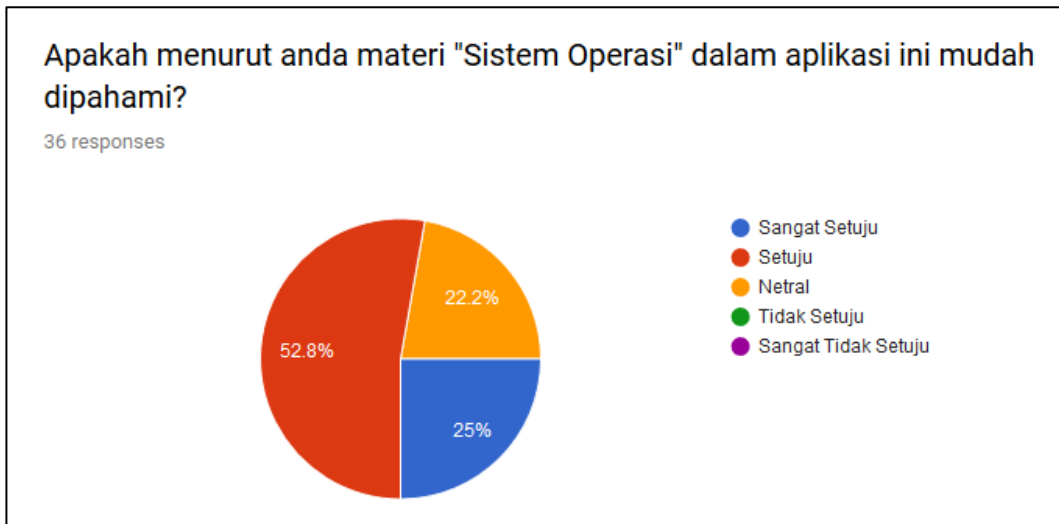
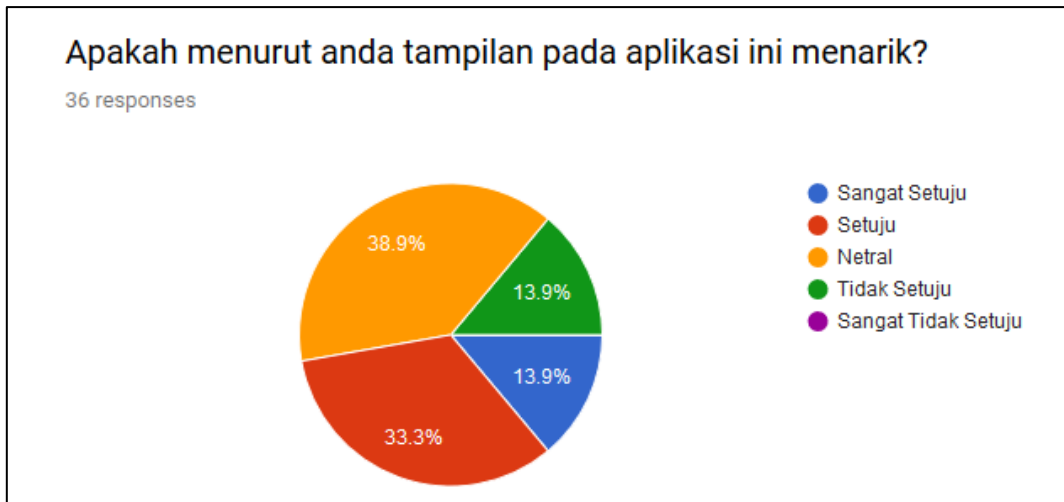
```
var view = document.getElementById("TableScore");
```

```
view.style.background="white";
```

```
view.style.overflow="scroll";
```

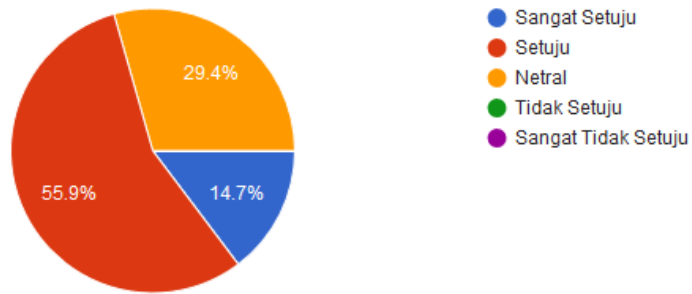
```
ShowAllData();
```

Lampiran 2 Hasil Kuesioner



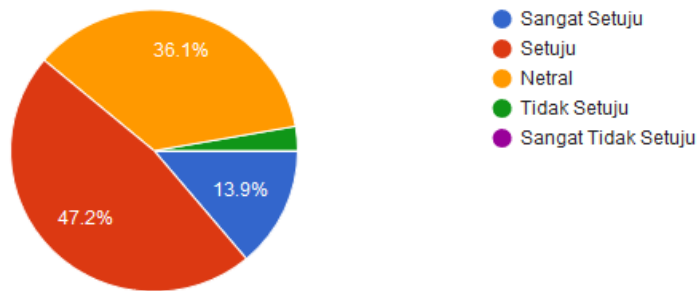
Apakah menurut anda pembelajaran dengan aplikasi ini lebih menyenangkan dibanding hanya dengan membaca modul?

34 responses



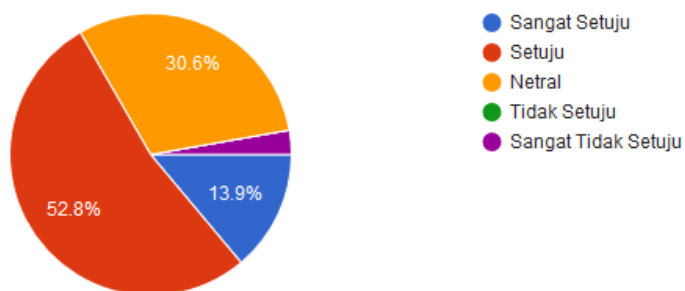
Apakah dengan CAI ini mendorong keinginan anda untuk belajar?

36 responses



Apakah anda mendapatkan kemudahan saat mengakses aplikasi ini karena tidak tergantung adanya internet?

36 responses



Apakah aplikasi ini membantu proses pembelajaran lebih mudah dan praktis?

36 responses

