

LAMPIRAN A

LISTING PROGRAM MACROMEDIA FLASH 8.0

//1: Inisialisasi Text yang masuk

```
function onFBInitText(){var router =
_parent.SessionArray[_parent.session];

                                router.assets_path[router.Assets.UserEntry].
text = "";

                                router.assets_path[router.Assets.UserEntry].
enabled = true;}
```

// 2: Tombol

```
function onFBButton(){var router =
_parent.SessionArray[_parent.session];

if(!router.buttonFlag || router.buttonFlag == false){evalFB();

var result = (router.result=="C")?2:3;
router.setFeedback(result);}if(router.result == "C"){
if(!router.scoreFlag || router.scoreFlag == false){

router.sessionStop();

router.scoreFlag = true;

router.assets_path[router.Assets.UserEntry].
enabled = false;}

router.setNavigation();

}else if (router.result == "W"){ if(router.numOfTries > 1){

if (!router.resetFlag || router.resetFlag == false){

                                router.buttonFlag = true;

                                router.resetFlag = true;

router.assets_path[router.Assets.UserEntry].enabled = false;

router.setComponentLabel(router.Assets.ControlButton,
router.buttonLabels[4]); }else{router.numOfTries--;
```

```

onFBInitText();

router.setFeedback(0);

router.buttonFlag = false;

router.resetFlag = false;

                router.setComponentLabel(router.Assets.Contr
olButton, router.label_state);}}}

else{if(!router.scoreFlag || router.scoreFlag == false){
router.sessionStop();

router.scoreFlag = true;

router.assets_path[router.Assets.UserEntry].
enabled = false;} router.setNavigation();}}}

// 3: Evaluasi input dan hasil

function evalFB(){var router = _parent.SessionArray[_parent.session];
var rNameRef = new Array();
var rValueRef = new Array();
var rCorrectRef = new Array();
var response_results = new Array();
var i=0,j=0,count=0;

// Collect user data from parameter Objects and build Arrays for
evaluation

for(var x in router.response){rNameRef[i] = x;
rValueRef[i] = router.response[x];i++;}

for (var y in router.response_value){rCorrectRef[j] =
router.response_value[y];j++;}

// Membuat array untuk evaluasi

rNameRef = rNameRef.reverse();

rValueRef = rValueRef.reverse();

rCorrectRef = rCorrectRef.reverse();

```

```

// Cek kesalahan

function checkFormat(string){var newFormat =
string.toString().split(";").toString();return newFormat;}

function removeChar (string){var stringFormat = string.toString();
var len = stringFormat.length; var i=0; while (i < len) {
if(stringFormat.substr(len-1,1) == "," || stringFormat.substr(len-1,1)
== ";"){
stringFormat = stringFormat.slice(0,len-1) len = stringFormat.length;
continue;}else{
break;}}return stringFormat;}

// pengecekan input yang dimasukkan user
router.correct_response = "{"+checkFormat(removeChar(rValueRef))+"}";
router.student_response =
"{"+checkFormat(router.assets_path[router.Assets.UserEntry].text)+"}";

// Cek respon dari 3 pilihan
for(var n=0; n < rValueRef.length; n++){
if(router.case_sensitive == false){
if (typeof router.response[rNameRef[n]] == "number"){
var response_temp = router.response[rNameRef[n]];
} else {var response_temp =
router.response[rNameRef[n]].toLowerCase();}

if(response_temp ==
router.assets_path[router.Assets.UserEntry].text.toLowerCase()) {if
(rCorrectRef[n] == true) {response_results[n] = true;}

else if (rCorrectRef[n] == false) {response_results[n] = false;}}
else {response_results[n] = "No Match";}} else {
if(router.response[rNameRef[n]] ==
router.assets_path[router.Assets.UserEntry].text) {if (rCorrectRef[n]
== true) {response_results[n] = true;}else if (rCorrectRef[n] ==

```

```

false) {response_results[n] = false;}}else{response_results[n] = "No
Match";}}}}

// cek kesamaan jawaban

if (router.exact_match == false) {

var getText = router.assets_path[router.Assets.UserEntry].text.split("
");

var textLen = getText.length;

for (var t=0; t< textLen; t++){

if(getText[t].toLowerCase() == rValueRef[0].toLowerCase() &&
rValueRef[0] <> ""){response_results[0] = true;

break;}

else if(getText[t].toLowerCase() == rValueRef[1].toLowerCase() &&
rValueRef[1] <> ""){

response_results[1] = true; break;}

else if(getText[t].toLowerCase() == rValueRef[2].toLowerCase() &&
rValueRef[2] <> ""){response_results[2] = true;

break;}}}}

// Respon jika jawaban berbeda

if (response_results[0] == "No Match" && response_results[1] == "No
Match" && response_results[2] == "No Match") {

if (router.response_value.other_responses == true) {router.result =
"C";}

else if (router.response_value.other_responses == false)
{router.result = "W";}}

else if (response_results[0] == true || response_results[1] == true
|| response_results[2] == true) {router.result = "C";}

else if (response_results[0] == false || response_results[1] == false
|| response_results[2] == false) {router.result = "W";}}

// Inisialisasi objek dan parameter

Response = new Object();

```

```
Response.Response1 = Response1;
Response.Response2 = Response2;
Response.Response3 = Response3;
Response_Value = new Object();
Response_Value.response_value1 = toBoolean(Response_Value1);
Response_Value.response_value2 = toBoolean(Response_Value2);
Response_Value.response_value3 = toBoolean(Response_Value3);
Response_Value.other_responses = toBoolean(Other_Responses);
Feedback_List = new Array();
Feedback_List[0] = Initial_Feedback;
Feedback_List[1] = Correct_Feedback;
Feedback_List[2] = Incorrect_Feedback;
Feedback_List[3] = Evaluate_Feedback;
Feedback_List[4] = Tries_Feedback;
Assets = new Object();
Assets.QuestionField = Assets1;
Assets.FeedbackField = Assets2;
Assets.UserEntry = Assets3;
Assets.ControlButton = Assets4;
ButtonLabels = new Array();
ButtonLabels[0] = undefined;
ButtonLabels[1] = ButtonLabels1;
ButtonLabels[2] = ButtonLabels2;
ButtonLabels[3] = ButtonLabels3;
ButtonLabels[4] = ButtonLabels4;
```

```

function toBoolean( str ){if(typeof(str) == "boolean"){return
str;}else if(str.toLowerCase() == "true"){return true;}else{return
false;}}

if(toBoolean( Feedback ) == false || typeof(Number(Num_Of_Tries)) !=
"number"){

Num_Of_Tries = 1;}

// Inisialisasi index

*/if(!_parent.SessionArray){_parent.session = 0;

_parent.SessionArray = new Array();

} else {_parent.session++;}

_parent.SessionArray[_parent.session] = new LToolBox();

var router = _parent.SessionArray[_parent.session];

// Initialize objektive ID

router.assets_path = eval(this._parent);

router.Assets = Assets;

router.interaction_id = Interaction_ID;

router.interaction_type = "F";

router.question = Question;

router.response = Response;

router.response_value = Response_Value;

router.case_sensitive = toBoolean(Case_Sensitive);

router.exact_match = toBoolean(Exact_Match);

router.tracking = toBoolean(Tracking);

router.weighting = Number(Weighting);

router.objective_id = Objective_ID;

router.navigation = Number(Navigation);

router.navLabel = Go_To_Label;

router.navAction = toBoolean(Go_To_Action);

```

```

router.feedback = toBoolean(Feedback);
router.numOfTries = Number(Num_Of_Tries);
router.feedback_list = Feedback_List;
router.buttonLabels = ButtonLabels;
router.buttonFlag = false;
router.resetFlag = false;
router.scoreFlag = false;
router.assets_path[router.Assets.UserEntry].changeHandler =
function(){
                                if (router.scoreFlag == false){
                                router.setComponentState(router.Assets.Contr
olButton, true);
                                router.setFeedback(1);}}
// inisialisasi feedback
router.setTextField(Assets.QuestionField, Question);
if(toBoolean( Feedback ) == false){
router.setTextField(Assets.FeedbackField, "");}
router.initControlButton(onFBBUTTON);
router.setFeedback(0);
router.sessionStart();
onFBInitText();

```

Pada kuis terdapat juga action scrip untuk mengacak dari setiap pertanyaan yang ditampilkan. Hal ini dilakukan untuk membuat tantangan bagi pengguna aplikasi ini. Action script yang digunakan adalah :

```

this._visible = false;
if(getTimer() > 1000){if(!startFlag){

```



```

// inisialisasi kuis

        _parent.QuizTrack = new Quiz();

// Inisialisasi object

        _parent.QuizTrack.randomize = Randomize;
        _parent.QuizTrack.quest_to_ask =
QuestionsToAsk;

        _parent.QuizTrack.activity_ID =
Activity_ID;

        _parent.QuizTrack.activity_name =
Activity_Name;

        _parent.QuizTrack.results_page = Results;
        _parent.QuizTrack.start_time =
Math.round(getTimer()/1000);;

        _parent.QuizTrack.level = _parent;
        _parent.QuizTrack.setQuestArray();

        _parent.QuizTrack.initStartQuiz();
startFlag = true;

stop();}

} else{

        play();}

```

LAMPIRAN C

HASIL SURVEY

LAMPIRAN B

NAMA KARAKTER TOKOH KARTUN



Wikipetan (ウィキペたん *Wikipetan*?) adalah salah satu maskot untuk Wikipedia Indonesia. Kata "-tan" adalah pergeseran pelafalan (*hypochorisma*) dari salah satu kata dalam bahasa Jepang. Sama seperti *-tan*, ia juga merupakan hasil dari *moe*.

Wikipetan adalah tokoh maskot untuk Wikipedia Indonesia, Anti Vandal, dan Pengurus; ia juga berjuang dalam pemilihan maskot Wikipedia. Selain itu, beberapa komunitas Kong menggunakan gambar dan Wikipetan sebagai maskot mereka, dalam masa kerja dan promosi mereka.



Nama tokoh dari karakter kartun diatas adalah Corgiwings

Judul gambar dari karakter diatas adalah Anime_South



Haruhi Suzumiya (涼宮ハルヒ *Suzumiya Haruhi?*), adalah nama untuk serial novel ringan yang ditulis oleh Tanigawa dan diilustrasikan oleh Ito. Novel pertama kali dipublikasikan di Jepang pada tanggal 6 Juni 2003 oleh Shoten. Novel ini diadaptasi menjadi tiga serial , tiga permainan video dan .

LAMPIRAN D

HIRAGANA

huruf hidup				
あ <i>a</i>	い <i>i</i>	う <i>u</i>	え <i>e</i>	お <i>o</i>
か <i>ka</i>	き <i>ki</i>	く <i>ku</i>	け <i>ke</i>	こ <i>ko</i>
さ <i>sa</i>	し <i>shi</i>	す <i>su</i>	せ <i>se</i>	そ <i>so</i>
た <i>ta</i>	ち <i>chi</i>	つ <i>tsu</i>	て <i>te</i>	と <i>to</i>
な <i>na</i>	に <i>ni</i>	ぬ <i>nu</i>	ね <i>ne</i>	の <i>no</i>
は <i>ha</i>	ひ <i>hi</i>	ふ <i>fu</i>	へ <i>he</i>	ほ <i>ho</i>
ま <i>ma</i>	み <i>mi</i>	む <i>mu</i>	め <i>me</i>	も <i>mo</i>
や <i>ya</i>		ゆ <i>yu</i>		よ <i>yo</i>
ら <i>ra</i>	り <i>ri</i>	る <i>ru</i>	れ <i>re</i>	ろ <i>ro</i>
わ <i>wa</i>	ゐ <i>wi</i>		ゑ <i>we</i>	を <i>wo</i>
				ん <i>n</i>
が <i>ga</i>	ぎ <i>gi</i>	ぐ <i>gu</i>	げ <i>ge</i>	ご <i>go</i>
ざ <i>za</i>	じ <i>ji</i>	ず <i>zu</i>	ぜ <i>ze</i>	ぞ <i>zo</i>
だ <i>da</i>	ぢ (<i>ji</i>)	づ (<i>zu</i>)	で <i>de</i>	ど <i>do</i>
ば <i>ba</i>	び <i>bi</i>	ぶ <i>bu</i>	べ <i>be</i>	ぼ <i>bo</i>
ぱ <i>pa</i>	ぴ <i>pi</i>	ぷ <i>pu</i>	ぺ <i>pe</i>	ぽ <i>po</i>

yōon		
(ya)	(yu)	(yo)
きゃ <i>kya</i>	きゅ <i>kyu</i>	きょ <i>kyo</i>
しゃ <i>sha</i>	しゅ <i>shu</i>	しょ <i>sho</i>
ちゃ <i>cha</i>	ちゅ <i>chu</i>	ちょ <i>cho</i>
にゃ <i>nya</i>	にゅ <i>nyu</i>	にょ <i>nyo</i>
ひゃ <i>hya</i>	ひゅ <i>hyu</i>	ひょ <i>hyo</i>
みゃ <i>mya</i>	みゅ <i>myu</i>	みょ <i>myo</i>
りゃ <i>rya</i>	りゅ <i>ryu</i>	りょ <i>ryo</i>
ぎゃ <i>gya</i>	ぎゅ <i>gyu</i>	ぎょ <i>gyo</i>
じゃ <i>ja</i>	じゅ <i>ju</i>	じょ <i>jo</i>
ぢゃ (<i>ja</i>)	ぢゅ (<i>ju</i>)	ぢょ (<i>jo</i>)
びゃ <i>bya</i>	びゅ <i>byu</i>	びょ <i>byo</i>
ぴゃ <i>pya</i>	ぴゅ <i>pyu</i>	ぴょ <i>pyo</i>

KATAKANA

ア <i>a</i>	イ <i>i</i>	ウ <i>u</i>	エ <i>e</i>	オ <i>o</i>
カ <i>ka</i>	キ <i>ki</i>	ク <i>ku</i>	ケ <i>ke</i>	コ <i>ko</i>
サ <i>sa</i>	シ <i>shi</i>	ス <i>su</i>	セ <i>se</i>	ソ <i>so</i>
タ <i>ta</i>	チ <i>chi</i>	ツ <i>tsu</i>	テ <i>te</i>	ト <i>to</i>
ナ <i>na</i>	ニ <i>ni</i>	ヌ <i>nu</i>	ネ <i>ne</i>	ノ <i>no</i>
ハ <i>ha</i>	ヒ <i>hi</i>	フ <i>fu</i>	ヘ <i>he</i>	ホ <i>ho</i>
マ <i>ma</i>	ミ <i>mi</i>	ム <i>mu</i>	メ <i>me</i>	モ <i>mo</i>
ヤ <i>ya</i>		ユ <i>yu</i>		ヨ <i>yo</i>
ラ <i>ra</i>	リ <i>ri</i>	ル <i>ru</i>	レ <i>re</i>	ロ <i>ro</i>
ワ <i>wa</i>	ヰ <i>wi</i>		ヱ <i>we</i>	ヲ <i>wo</i>
				ン <i>n</i>
ガ <i>ga</i>	ギ <i>gi</i>	グ <i>gu</i>	ゲ <i>ge</i>	ゴ <i>go</i>
ザ <i>za</i>	ジ <i>ji</i>	ズ <i>zu</i>	ゼ <i>ze</i>	ゾ <i>zo</i>
ダ <i>da</i>	ヂ <i>ji</i>	ヅ <i>zu</i>	デ <i>de</i>	ド <i>do</i>
バ <i>ba</i>	ビ <i>bi</i>	ブ <i>bu</i>	ベ <i>be</i>	ボ <i>bo</i>
パ <i>pa</i>	ピ <i>pi</i>	プ <i>pu</i>	ペ <i>pe</i>	ポ <i>po</i>

キヤ <i>kya</i>	キユ <i>kyu</i>	キヨ <i>kyo</i>
シャ <i>sha</i>	シュ <i>shu</i>	ショ <i>sho</i>
チャ <i>cha</i>	チュ <i>chu</i>	チョ <i>cho</i>
ニヤ <i>nya</i>	ニユ <i>nyu</i>	ニヨ <i>nyo</i>
ヒヤ <i>hya</i>	ヒユ <i>hyu</i>	ヒヨ <i>hyo</i>
ミヤ <i>mya</i>	ミユ <i>myu</i>	ミヨ <i>myo</i>
リヤ <i>rya</i>	リュ <i>ryu</i>	リヨ <i>ryo</i>
ギヤ <i>gya</i>	ギユ <i>gyu</i>	ギヨ <i>gyo</i>
ジャ <i>ja</i>	ジュ <i>ju</i>	ジョ <i>jo</i>
ビヤ <i>bya</i>	ビユ <i>byu</i>	ビヨ <i>byo</i>
ピヤ <i>pya</i>	ピユ <i>pyu</i>	ピヨ <i>pyo</i>

KANJI

日本	東京	大阪	北海道			
Japan	Tokyo	Osaka	Hokkaido			
山	川	日	雨	水	火	田
mountain	river	sun	rain	water	fire	rice field
米	魚	寿司	肉	酒	茶	
rice	fish	sushi	meat	alcohol	tea	
車	電気	自転車	飛行機			
car	electricity	bicycle	airplane			
一	二	三	四	五	六	七
one	two	three	four	five	six	seven
男	女	松井秀喜	黒沢明			
man	woman	Matsui Hideki	Kurozawa Akira			
食べる	行く	小	大	多	少	
to eat	to go	small	big	many	few	