#include <LCD.h>

#include <LiquidCrystal\_SR.h>

#include <LiquidCrystal\_SR\_LCD3.h>

const int

LCD\_DATA = 4,

LCD\_CLOCK = 3,

LCD\_STB = 2;

const int

pingPin = 13,

echoPin = 12,

buzzer = A5;

int

a = 11, b = 10, c = 9, d = 8, e = 7, f = 6, g = 5;

LiquidCrystal\_SR\_LCD3 lcd(LCD\_DATA, LCD\_CLOCK, LCD\_STB);

byte counter, tombol;

boolean statusRunGame;

String gameComponent;

void setup() {

// Inisiasi untuk pin seven segment

for(int i = 5; i <= 11; i++)

pinMode(i, OUTPUT);

counter = 3;

displayToSeventSegment(counter);

// Inisiasi untuk pin buzzer

pinMode(buzzer, OUTPUT);

// Initiasi untuk LCD Control

lcd.begin(16, 2);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Andy Maulana Y");

lcd.setCursor(0, 1);

lcd.print("6706164071");

// Inisiasi untuk Serial Control

Serial.begin(9600);

Serial.println("Tekan spasi untuk memulai permainan");

// Inisiasi Game Engine

statusRunGame = false;

gameComponent = "0X0X00X0XX0X0X00X00X";

}

void loop() {

// Mulai game

startToGame();

}

void startToGame() {

// Check jika ada inputan dan status game belum mulai

if (Serial.available() > 0 && !statusRunGame) {

tombol = Serial.read();

if (tombol == 32) {

Serial.println("Game Dimulai !!!");

lcd.clear();

statusRunGame = true;

}

}

// Check status game untuk memulai harus bernilai true

if (statusRunGame) {

runGameComponent();

}

}

/\*\*

\* Play Game

\*/

void runGameComponent() {

int

indexUp = 0,

indexDown = 19,

finalPoint = 0;

String

contentUp = String(),

contentDown = String(),

lastCharacterUp = String(),

lastCharacterDown = String();

boolean show = false, loop = true, action = false, i = 0;

while(loop) {

lcd.clear();

int buffers = 0;

if(Serial.available() > 0) {

tombol = Serial.read();

if (tombol == 119) {

lcd.setCursor(15, 0);

lcd.print("<");

lcd.setCursor(15, 1);

lcd.print(" ");

action = true;

buffers = 119;

}

else if (tombol == 115) {

lcd.setCursor(15, 1);

lcd.print("<");

lcd.setCursor(15, 0);

lcd.print(" ");

action = true;

buffers = 115;

}

}

if (action) {

soundEffect();

if (!show) {

if (i == 0) contentUp = contentUp;

else contentUp = " " + contentUp;

if (i >= 20) {

contentDown = " " + contentDown;

}

else contentDown = gameComponent[indexDown] + contentDown;

}

else {

if (i == 0) contentDown = contentDown;

else contentDown = " " + contentDown;

if (i >= 20) {

contentUp = " " + contentUp;

}

else contentUp = gameComponent[indexUp] + contentUp;

}

delay(100);

lcd.setCursor(0, 0);

lcd.print(contentUp);

lcd.setCursor(0, 1);

lcd.print(contentDown);

lcd.println();

if (contentUp.length() < 15 || contentDown.length() < 15) {

Serial.println("Isi bagian atas : " + contentUp);

Serial.println("Isi bagian bawah : " + contentDown);

}

else {

lastCharacterDown = contentDown[15];

lastCharacterUp = contentUp[15];

Serial.println("Isi bagian atas : " + lastCharacterUp);

Serial.println("Isi bagian bawah : " + lastCharacterDown);

if (buffers == 119) {

if (lastCharacterUp == "0") displayToSeventSegment(++counter);

else if (lastCharacterUp == "X") displayToSeventSegment(--counter);

}

else if (buffers == 115) {

if (lastCharacterDown == "0") displayToSeventSegment(++counter);

else if (lastCharacterDown == "X") displayToSeventSegment(--counter);

}

if (counter == 0 || (contentUp.length() >= 36 && contentDown.length() >= 36)) {

loop = false;

statusRunGame = false;

}

}

delay(100);

if (!show) {

contentUp = "" + contentUp;

show = true;

}

else {

contentDown = "" + contentDown;

show = false;

}

indexUp++;

indexDown--;

i++;

finalPoint = counter;

action = false;

}

}

if (counter == 0) {

Serial.println("Game Over");

Serial.print("Your Point : ");

Serial.print(finalPoint);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Game Over");

lcd.setCursor(0,1);

lcd.print("Your Point : ");

lcd.print(finalPoint);

}

else {

Serial.println("Time's Up");

Serial.print("Your Point : ");

Serial.print(finalPoint);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Time's Up");

lcd.setCursor(0,1);

lcd.print("Your Point : ");

lcd.print(finalPoint);

}

}

void soundEffect() {

pinMode(pingPin, OUTPUT);

digitalWrite(pingPin, LOW);

delayMicroseconds(2);

digitalWrite(pingPin, HIGH);

delayMicroseconds(10);

digitalWrite(pingPin, LOW);

pinMode(echoPin, INPUT);

long duration = pulseIn(echoPin, HIGH);

long cm = duration / 29 / 2;

if (cm < 300) digitalWrite(buzzer, HIGH);

else digitalWrite(buzzer, LOW);

}

/\*\*

\* Fungsi ini digunakan untuk update display pada

\* sevent segment, dengan memasukkan parameter angka yang

\* ingin ditampilkan dari angka 0 - 9.

\*/

void displayToSeventSegment(int number) {

switch(number) {

case 1:

digitalWrite(a, LOW);

digitalWrite(b, HIGH);

digitalWrite(c, HIGH);

digitalWrite(d, LOW);

digitalWrite(e, LOW);

digitalWrite(f, LOW);

digitalWrite(g, LOW);

break;

case 2:

digitalWrite(a, HIGH);

digitalWrite(b, HIGH);

digitalWrite(c, LOW);

digitalWrite(d, HIGH);

digitalWrite(e, HIGH);

digitalWrite(f, LOW);

digitalWrite(g, HIGH);

break;

case 3:

digitalWrite(a, HIGH);

digitalWrite(b, HIGH);

digitalWrite(c, HIGH);

digitalWrite(d, HIGH);

digitalWrite(e, LOW);

digitalWrite(f, LOW);

digitalWrite(g, HIGH);

break;

case 4:

digitalWrite(a, LOW);

digitalWrite(b, HIGH);

digitalWrite(c, HIGH);

digitalWrite(d, LOW);

digitalWrite(e, LOW);

digitalWrite(f, HIGH);

digitalWrite(g, HIGH);

break;

case 5:

digitalWrite(a, HIGH);

digitalWrite(b, LOW);

digitalWrite(c, HIGH);

digitalWrite(d, HIGH);

digitalWrite(e, LOW);

digitalWrite(f, HIGH);

digitalWrite(g, HIGH);

break;

case 6:

digitalWrite(a, HIGH);

digitalWrite(b, LOW);

digitalWrite(c, HIGH);

digitalWrite(d, HIGH);

digitalWrite(e, HIGH);

digitalWrite(f, HIGH);

digitalWrite(g, HIGH);

break;

case 7:

digitalWrite(a, HIGH);

digitalWrite(b, HIGH);

digitalWrite(c, HIGH);

digitalWrite(d, LOW);

digitalWrite(e, LOW);

digitalWrite(f, LOW);

digitalWrite(g, LOW);

break;

case 8:

digitalWrite(a, HIGH);

digitalWrite(b, HIGH);

digitalWrite(c, HIGH);

digitalWrite(d, HIGH);

digitalWrite(e, HIGH);

digitalWrite(f, HIGH);

digitalWrite(g, HIGH);

break;

case 9:

digitalWrite(a, HIGH);

digitalWrite(b, HIGH);

digitalWrite(c, HIGH);

digitalWrite(d, HIGH);

digitalWrite(e, LOW);

digitalWrite(f, HIGH);

digitalWrite(g, HIGH);

break;

default:

digitalWrite(a, HIGH);

digitalWrite(b, HIGH);

digitalWrite(c, HIGH);

digitalWrite(d, HIGH);

digitalWrite(e, HIGH);

digitalWrite(f, HIGH);

digitalWrite(g, LOW);

break;

}

}