BAB I

PENDAHULUAN

A. Latar Belakang

Nasionalisme adalah konsep mengenai jati diri yang berfungsi dalam penetapan identitas kebangsaan. Nasionalisme atau kebangsaan bukan sekadar instrument yang berfungsi sebagai perekat kemajemukan secara eksternal, namun juga merupakan wadah yang menegaskan identitas Indonesia yang bersifat plural dalam berbagai dimensi kulturnya. Nasionalisme menuntut adanya perwujudan nilai-nilai dasar yang berorientasi kepada kepentingan bersama dan menghindarkan segala legalisasi kepentingan pribadi yang merusak tatanan kehidupan bersama (Kusumawardani & Faturochman, 2004)

Tantangan bagi nasionalisme lahir seiring dengan semakin modernnya kehidupan manusia dimana jarak bukan lagi suatu halangan dan media telekomunikasi telah menyatukan semua lapisan masyarakat menjadi suatu *global village*. Dalam hal ini, globalisasi telah menjadi ujung tombak dalam mengikir paham nasionalisme (Hendrastomo, 2007). Sejalan dengan hal tersebut, Menurut Amrah (2016) mengemukakan bahwa masalah nasionalisme tidak lagi dapat dilihat sebagai masalah sederhana yang dapat dilihat dari satu perspektif saja. Saat ini masalah pembangunan nasionalisme di Indonesia tengah menghadapi tantangan yang berat. Di tengah situasi bangsa Indonesia yang seperti itu, nasionalisme sangat di butuhkan untuk menjaga Negara Kesatuan Republik Indonesia.

Menurut (Widjoyo, 2020) menyatakan bahwa nasionalisme adalah suatu sikap atau paham dari masyarakat suatu bangsa yang mempunyai kesamaan kebudayaan, dan wilayah serta kesamaan cita-cita dan tujuan, dengan demikian masyarakat suatu bangsa tersebut akan merasakan adanya kesetiaan yang mendalam terhadap bangsa itu sendiri.

Rasa nasionalisme perlu ditumbuhkan kepada anak sejak usia dini. Sejak dini generasi penerus bangsa harus paham bahwa kemerdekaan akan tanah air dan bumi pertiwi ini tidak diperoleh dengan mudah. Sosok yang telah berjuang dan berkorban nyawa mengusir penjajah demi kedaulatan Negara Kesatuan Republik Indonesia (NKRI) adalah atas perjuangan para pahlawan. Dengan memperkenalkan tokoh pahlawan diharapkan mampu menginspirasi maupun memotivasi anak bangsa untuk senantiasa meneruskan perjuangan mereka sesuai dengan tantangan dan dinamika kehidupan bangsa dan negara.

Pengenalan tokoh pahlawan bagi anak usia dini adalah salah satu alternatif untuk mengantisipasi dan mengatasi masalah nasionalisme yang merupakan salah satu dampak dari globalisasi. Kini, sosok pahlawan bagi anak-anak hanya sebatas fiksi, mereka lebih mengagumi setiap peran super hero yang ditayangkan di berbagai media seperti TV, bioskop dan gadget dibanding pahlawan di kehidupan nyata terkhusus bagi negaranya sendiri. Akibatnya adalah sejarah terlupakan, nasionalisme memudar dan kesadaran patriotisme akan hilang.

Media pembelajaran adalah media yang dapat menyampaikan pesan pembelajaran atau mengandung muatan untuk membelajarkan seseorang. Media pembelajaran adalah media kreatif yang digunakan dalam memberikan materi

kepada anak didik sehingga proses belajar mengajar lebih efektif, efesien, dan menyenangkan (Wibawanto dkk, 2017)

Lebih lanjut menurut (Anggraeny, 2020) salah satu metode pembelajaran yang menarik adalah dengan menggunakan metode *Puzzle*. Menurut (Fatimah dkk., 2017) dalam penelitiannya menjelaskan bahwa *Game Puzzle* adalah salah satu jenis permainan yang dapat meningkatkan daya ingat serta kreatifitas para pemain dalam menyelesaikan masalah dengan cepat dan tepat. Sejalan dengan hal tersebut menurut (Atthariq dkk., 2017) mengemukakan bahwa *Game Puzzle* merupakan media yang sangat efektif dalam meningkatkan keterampilan kognitif, motorik, sosial, melatih kordinasi mata dan tangan, logika, kesabaran dan memperluas pengetahuan.

Berdasarkan latarbelakang diatas, penulis merancang sebuah aplikasi *Game Puzzle* sebagai media pembelajaran untuk memperkenalkan tokoh-tokoh pahlawan Revolusi kepada anak usia dini. *Game* ini diharapkan mampu memacu keterampilan dan kecerdasan otak anak dengan cara yang lebih menarik dan menyenangkan. Adapun metode yang digunakan untuk menyelesaikan aplikasi *Game* ini adalah *Fisher Yates Shuffle*. *Fisher Yates Shuffle* merupakan salah satu metode pengacakan berdasarkan data input yang diberikan.

B. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka dapat dirumuskan beberapa masalah yaitu:

1. Bagaimana perancangan *Game Puzzle* pengenalan tokoh pahlawan Revolusi Indonesia?

2. Bagaimana penerapan *Fisher Yates Shuffle* dalam penyelesaian permainan *Game Puzzle*?

C. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Membuat perancangan media pembelajaran merangkai/mencocokkan gambar (*Puzzle*) yang menarik dan interaktif bagi pemain (anak usia dini).
2. Memperkenalkan kepada pemain (anak usia dini) mengenai tokoh-tokoh pahlawan revolusi Indonesia

D. Batasan Masalah

Dengan banyaknya aspek dalam membangun suatu aplikasi berbasis android,

maka dibutuhkan batasan masalah yang jelas untuk menghidari ketidakjelasan dalam

pembahasan. Adapun batasan masalah dalam penulisan ini agar ruang lingkup masalah tidak meluas yaitu :

1. Aplikasi yang dibangun lebih ditujukan untuk anak Sekolah Dasar.
2. Jumlah *pieces* atau jumlah pilihan potongan gambar sudah ditentukan.
3. Informasi yang diberikan pada keterangan gambar bersifat umum.
4. Aplikasi yang akan dibangun hanya bisa dimainkan pada mode 1 player saja.
5. Pembuatan aplikasi menggunakan metode *Fisher Yates Shuffle.*
6. Memperkenalkan kepada pemain (anak usia dini) mengenai tokoh – tokoh pahlawan revolusi indonesi

E. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Memperkaya pengetahuan pemain (anak usia dini) mengenai tokoh-tokoh pahlawan Revolusi Indonesia.
2. Menjadi sarana pembelajaran yang menarik dan edukatif bagi pemain (anak usia dini).

F. Sistematika Penulisan

Sistematika penulisan tugas akhir ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan tugas akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini menguraikan tentang latar belakang permasalahan, mencoba merumuskan inti permasalahan yang dihadapi, menentukan tujuan dan kegunaan penulisan, yang kemudian diikuti dengan pembatasan masalah, asumsi, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menguraikan tentang teori-teori yang relevan dengan kajian penelitian dan hal-hal lainnya yang mendukung penelitian serta berguna dalam pengembangan aplikasi *Game Puzzle* pengenalan tokoh pahlawan revolusi.

BAB III METODE PENELITIAN

Bab ini menguraikan tentang proses rekayasa balik dari aplikasi permainan yang dijadikan acuan kemudian menganalisis hasilnya untuk kebutuhan

pengembangan aplikasi.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menguraikan implementasi dari pembangunan aplikasi *Game Puzzle* berbasis android dan berisi tentang hasil dari perancangan dalam pembuatan sistem dan tahapan-tahapan yang dilakukan untuk menerapkan sistem yang telah dirancang.

BAB V PENUTUP

Pada bab ini berisi kesimpulan yang didapatkan dari analisis mengenai keterkaitan dengan penggunaan sistem dan pengembangan sistem yang akandatan

BAB II

TINJAUAN PUSTAKA

A. *Algoritma Fisher Yates Shuffle*

Algoritma *Fisher Yates* yang pertama diusulkan pada tahun 1938 dan dikaji pada tahun 1948 dengan versi modern yang disajikan dalam sebuah varian. Algoritma diterbitkan oleh Wilson pada tahun 2004 bernama "*Algoritma Santtolo*". Dalam memvalidasi *Fisher Yates Shuffle*, sebuah analisis statistik dari algoritma menggunakan analisis frekuensi yang disampaikan dalam sebuah kesimpulan-kesimpulan analisa yang menarik tentang kecepatan algoritma ini (Asih dkk., 2020). Algoritma merupakan bentuk dasar dari perintah-perintah yang akan dikodingkan ke dalam form yang telah dirancang pada tahap implementasi sistem (Yusfrizal, 2020).

Metode *Fisher Yates Shuffle* secara umum adalah sebagai berikut :

1. Inisialisasi jumlah elemen dalam array.
2. Ambil elemen secara acak dari elemen yang tersisa
3. Kemudian tukar dengan elemen saat ini.

Metode Pengacakan *Fisher Yates*

Metode Pengacakan *Fisher-Yates* adalah sebuah metode untuk menghasilkan suatu permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan tersebut. Misalkan dalam permainan kartu bridge ada 10 kartu yang diacak, maka *array*-nya adalah urutan Kartu = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]. *Array* tersebut dimasukkan kedalam prosedur pengacakan dimana

proses pengacakan terjadi, kemudian didapat panjang *array* yang kemudian dimasukkan ke dalam variabel m (m=10).

Langkah–langkah pengacakan *Puzzle* dengan Metode Pengacakan *Fisher-Yates* adalah sebagai berikut :

1. Ambil satu elemen secara acak dari elemen yang tersisa. Pengambilan elemen acak adalah berdasarkan elemen yang tersisa.Misalkan jika m =10, maka elemen acak yang boleh diambil adalah 10 (*array*[0…9]).
2. Tukar dengan elemen saat ini. Penukaran dilakukan dengan memasukkan elemen saat ini ke dalam suatu variabel baru bernama t. Elemen saat ini (*array*[m]) diisi nilai dari elemen acak tadi (*array*[i]) dan elemen acak (*array*[i]) diisi nilai dari variabel t.
3. Ulangi selama masih ada elemen yang tersisa (Edora, 2019).

Berikut adalah contoh pengerjan *algoritma fisher yates*, sebagai keterangan Range adalah jumlah angka yang belum terpilih, roll adalah angka acak yang terpilih, *scratch* adalah daftar angka yang belum terpilih dan result adalah hasil permutasi yang didapatkan.

Tabel 2. 1 Contoh pengerjaan *Algoritma Fisher Yates*

Range	Roll	Scratch	Result
		12345678	
1-8	5	1234867	5
1-7	4	123786	45
1-6	1	62378	145
1-5	6	8237	6145
1-4	3	827	36145
1-3	7	82	736145
1-2	8	2	8736145

Permutasi yang didapatkan pada table contoh pengerjaan *algoritma fisher yates* adalah 28736145.

B. *Sublime Text*

Sublime Text Editor adalah editor teks untuk berbagai bahasa pemograman termasuk pemograman PHP. *Sublime Text* Editor merupakan editor text lintas *platform* dengan *Python Application Programming Interface* (API). *Sublime Text* Editor juga mendukung banyak bahasa pemrograman dan bahasa markup, dan fungsinya dapat ditambah dengan *plugin*, dan *Sublime Text* Editor tanpa lisensi perangkat lunak. (Pasaribu.,dkk 2017)

Sublime Text 3 digunakan sebagai editor dari bahasa pemrograman PHP dalam melakukan pengelolaan konten di dalam aplikasi server, *Sublime Text* adalah text editor berbasis *Python*, sebuah text editor yang elegan, kaya fitur, *cross platform*, mudah dan simple yang cukup terkenal dikalangan *developer* (pengembang) dan desainer (Helfira, 2019)

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai *platform operating system* dengan menggunakan teknologi *Phyton* API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim. Aplikasi ini sangatlah fleksibel dan powerfull. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan sublime-packages. *Sublime Text* bukanlah aplikasi opensource dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (packages) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki linsensi aplikasi gratis (Siregar, 2019)

C. *Javascript*

Javascript adalah suatu bahasa kode atau pemrograman yang digunakan untuk menciptakan *web*site agar menjadi dinamis. Beberapa fitur yang dapat membuat situs menjadi dinamis (dan tentunya membutuhkan bahasa pemrograman) di antaranya adalah gambar animasi,*slideshow*, foto, saran pengisian teks otomatis, dan banyak lagi. Dibalik semua animasi nteraktif dan form otomatis pada di suatu situs, ada kumpulan rumus bahasa pemrograman seperti *Javascript* (Teknologi .,dkk 2021)

Menurut (Muarie, 2015) *Javascript* adalah bahasa skrip yang ditempatkan pada kode HTML dan diproses pada sisi klien. Dengan adanya bahasa ini maka kemampua dokumen HTML menjadi lebih luas.

Javascript menurut (Wasil dkk., 2020)adalah bahasa *scripting* yang popular di internet dan dapat bekerja di sebagian besar browser popoler *seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode Javascript* dapat disisipkan dalam halaman *web* menggunakan tag SCRIPT.

Adapun penerapan *Javascript* pada *game puzzle* adalah sebagai berikut :

```
<script>
let printArray = (arr, n)=>
{
      ans = '';
      for (let i = 0; i < n; i++)
      {
            ans += arr[i] + " ";
      }
      console.log(ans);
}
let randomize = (arr, n) =>
{
      for (let i = n - 1; i > 0; i--)
{
            let j = Math.floor(Math.random() * (i + 1));
            [arr[i], arr[j]] = [arr[j], arr[i]];
      }
```

```
}
let arr = [1, 2, 3, 4, 5, 6, 7, 8];
let n = arr.length;
randomize (arr, n);
printArray(arr, n);
</script>
```

D. *Cordova*

Cordova adalah sekumpulan API yang dapat digunakan oleh *developer* untuk mengakses fungsi *native* dari sebuah perangkat, sebagai contohnya adalah kamera dan *accelerometer* menggunakan *Javascript*. Dengan *Cordova*, *developer* tidak perlu mempelajari bahasa pemprograman baru yang berkaitan dengan *platform* tujuan. *Developer* cukup membuat aplikasi menggunakan *Javascript*, HTML, dan CSS selayaknya aplikasi *web* biasa kemudian cukup menggunakan *Cordova* sehingga aplikasi tersebut dapat berjalan selayaknya aplikasi *native* pada perangkat *mobile*. Aplikasi yang dihasilkan dari *Cordova* adalah sebuah aplikasi *hybrid*, karena proses *rendering* dilakukan pada *web view* namun juga memiliki akses pada API *native* perangkat (Ningsih dkk., 2020)

Apache Cordova adalah framework untuk membuat aplikasi pada berbagai macam platform seperti *Android, Blackberry, Iphone, atau Windows Phone* menggunakan HTML5, *Jquery, Jquery Mobile* dan CSS3. Membangun aplikasi untuk device yang berbeda seperti *Android, Iphone, Windows Mobile* dan lainnya dibutuhkan framework dan bahasa pemrograman yang berbeda, seperti pada Android menggunakan bahasa pemrograman *Java, blackberry dengan Java, Iphone* dengan Basic C dan *Windows Phone* dengan C#. (Sony Putra, 2017)

Adapun penerapan Cordova pada *game puzzle* adalah sebagai berikut :

```
<?xml version='1.0' encoding='utf-8'?>
xmlns="http://www.w3.org/ns/widgets"
```

```
xmlns:cdv="http://cordova.apache.org/ns/1.0">
    <name>puzzle</name>
    <description>
    </description>
    <author email="dev@cordova.apache.org"href="http://cordova.io">
        Apache Cordova Team
    </author>
    <content src="index.html" />
    <access origin="*" />
    <allow-navigation href="*"/>
    <allow-intent href="http://*/*" />
    <allow-intent href="https://*/*" />
    <allow-intent href="tel:*" />
    <allow-intent href="sms:*" />
    <allow-intent href="mailto:*" />
    <allow-intent href="geo:*" />
    <platform name="android">
        <allow-intent href="market:*" />
    </platform>
      <platform name="ios">
          <allow-intent href="itms:*" />
        <allow-intent href="itms-apps:*"/>
    </platform>
    <preference name="AndroidFileLocation" value="Compatibility"/>
</widget>
```

E. HTML

Menurut (Wasil dkk., 2020) HTML5 (*Hypertext Markup Language* version 5) adalah sebuah bahasa markah yang menstrukturkan isi dari *World Wide Web*, sebuah teknologi utama pada internet. Standar HTML5 menyempurnakan elemen-elemen lama yang terdapat pada standar sebelumnya, menambahkan lemen-elemen yang lebih semantik dan menambahkan fitur-fitur baru untuk mendukung pembuatan aplikasi *web* yang lebih kompleks.

Menurut (Ii, 2017)“HTML(*Hypertext Markup Language*) adalah sekumpulan dari berbagai symbol atau tag yang dituliskan dalam sebuah file yang digunakan untuk menampilkanhalaman pada browser". tag-tag HTML selalu dimulai dengan simbul <x> dan diakhiri </x> dimanaa x tag-tag HTML seperti u,, b, i dan lain-lainnya”.

Menurut (Muarie, 2015) HTML(*Hypertext Markup Language*) merupakan bahasa Yang digunakan untuk menulis halaman *web*. Biasanya mempunyai ektensi .htm, .html, atau shtml. HTML tersusun atas tag-tag, digunakan untuk menentukan tamoian dari dokumen HTML yang diterjemahkan oleh browser. Tag HTML tidak *case sensitive.* Jadi bisa menggunakan <HTML> atau <html>. Keduanya menghasilkan output yang sama.

Kerangka dasar dokumen HTML diatur oleh W3 Consurtium, yaitu :

```
<HTML>
      <HEAD>
            Universitas Muhammadiyah Parepare
      </HEAD>
      <BODY>
      Isi Dokumen
      </BODY>
</HTML>
```

Keterangan :

a) HTML : Setiap dokumen HTML biasanya diawali dan ditutup dengan tag HTML, yang memberi di dalam kedua tag tersebut adalah dokumen HTML.
b) HEAD : Bagian header dari dokumen HTML, berada di antara tag HEAD. Di dalam bagian ini biasanya dimuat tag TITLE yang menampilkan judul halaman pada bagian title milik *browser. Header* juga memuat tag META yang biasanya digunakan untuk menentukan informasi tertentu mengenai dokumen HTML.
c) BODY : Dokumen *body* digunakan untuk menampilkan pada halaman *web*.

Adapun penerapan script HTML pada game puzzle adalah sebagai berikut :

```
<html>
    <head>
        <meta charset="UTF-8" />
        <meta  name="viewport"  content="width=device-width,
initial-scale=1">
```

```
            <style>
                html, body {
                    height: 100%;
                    width: 100%;
                    margin: 0;
                    padding: 0;
                    overflow: hidden;
                }
                #boxhtml {
                    height: calc(100% - 20px);
                    width: calc(100% - 20px);
                    padding: 10px;
                }
            </style>
        </head>
        <body>
            <!-- <div id="boxhtml"></div> -->
            <iframe            src=""            frameborder="0"
    id="boxhtml"></iframe>
            <script        src="https://code.jquery.com/jquery-
    3.6.0.min.js"                              integrity="sha256-
    /xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4="
    crossorigin="anonymous"></script>
            <script>
                // $(document).ready(function() {
                //
    $('#boxhtml').attr('src','jigzaw/index.html')
                //
    $('#boxhtml').attr('src','sliding/index.html')
                // })
            </script>
        </body>
    </html>
```

F. PHP

Bahasa pemrograman *Open Source* yang digunakan untuk membuat sebuah aplikasi *web*. PHP *Hypertext Preprocessor* merupakan pemrograman *server side*, yaitu bahasa yang berjalan di sisi server. Kode program PHP akan dieksekusi oleh server dan hasil eksekusi tersebut akan di tampilakn kepada *client*. Keunggulan PHP adalah sangat sederhana dan mudah dimengerti sehingga cocok untuk pemula maupun tingkat lanjut. Selain itu, karena PHP merupakan *Open Source* maka PHP dapat digunakan di banyak sistem operasi (*Windows*, *Linux*, *Mac*, *Solaris*, dan lain-lain). Tujuan utama bahasa ini adalah untuk

memungkinkan perancang *web* untuk menulis halaman *web* dinamik dengan cepat. Struktur pada PHP : yang merupakan struktur pada PHP adalah struktur kendali atau statement yang menjadi bagian penting dalam suatu bahasa pemrograman, karena bagian ini mengatur jalurnya eksekusi suatu program. Struktur penulisan php, dapat dituliskan berdiri sendiri atau disisipkan pada script HTML. (Siregar, H. F.,dkk, 2018)

Adapun penerapan PHP pada *game puzzle* adalah sebagai berikut :

```
<?php
function randomize ($arr, $n)
{

      for($i = $n - 1; $i >= 0; $i--)
      {
            $j = rand(0, $i+1);
            $tmp = $arr[$i];
            $arr[$i] = $arr[$j];
            $arr[$j] = $tmp;
      }
      for($i = 0; $i < $n; $i++)
      echo $arr[$i]." ";
}

$arr = array(1, 2, 3, 4,
                  5, 6, 7, 8);
$n = count($arr);
randomize($arr, $n);
?>
```

G. CSS

Menurut (Muarie, 2015) CSS (*Cascading Style Sheets*) banyak digunakan untuk memperluas kemampuan HTML dalam memformat dokumen *web* atau untuk mempercanyik tampilan *web* , bahkan untuk pemosisikan dan *layouting* halaman *web* . dengan mendefinisikan suatu *style* itu akan dapat digunakan berulang kali. CSS dapat mengendalikan ukuran gambar, warnabagian tubuh teks,

warna tabel, ukuran border, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri,kanan,atas bawah, dan parameter lainnya.

Adapun penerapan CSS pada *game puzzle* adalah sebagai berikut :

```
.animasicontent-box {
  color: #fff;
  font-family: "Roboto", sans-serif;
  margin: 0 auto;
  text-align: center;
  padding-bottom: 1em;
}
.animasicontainer {
  height: 602px;
  width: 85%;
  max-width: 602px;
  display: inline-block;
  position: relative;
}
.animasicircle-clip {
}
.animasileft,
.animasiright {
  height: 100%;
  position: absolute;
  width: 100%;
  perspective: 150px;
  transform-style: preserve-3d;
  transform: scaleX(1.10) scaleY(1.10);
```

H. MYSQL

Menurut (Wasil dkk., 2020)“ MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengelolaan datanya”.

Jadi, MySQL adalah sebuah *database server* , dapat juga berperan sebagai *client* sehingga sering disebut *database client/server*, yang *Open Source* dengan kemampuan dapat berjalan baik di OS maupun, dengan *Platform Windows* dan *Linux*. Kelebihan MySQL ialah mampu menerima *query* yang bertumpuk dalam satu permintaan atau yang disebut *Multi-Threading*. MySQL juga sebagai

Relation Database Management System, sebagai halnya *ORACLE*, *postgresql*, MS SQL dan sebagainya.

Adapun penerapan MySQL pada *game puzzle* adalah sebagai berikut :

```
      CREATE TABLE `admin` (
        `id` int(11) NOT NULL,
        `username` varchar(50) NOT NULL,
        `password` varchar(500) NOT NULL,
        `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
        `updated_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP
ON
         UPDATE CURRENT_TIMESTAMP
      ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

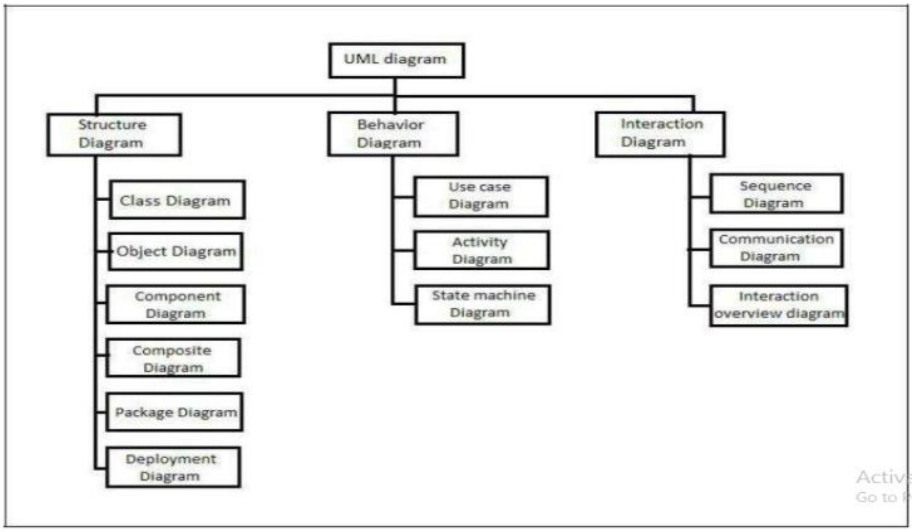
I. *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan kebutuhan, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman beorientasi objek. UML digunakan untuk kebutuhan pemodelan visual dalam menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak (Marwan dkk., 2017).

Unified Modeling Language (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Pongoh dkk., 2016)

UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. UML (*Unified Modeling Language*) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun system.

Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar di bawah ini.



Gambar 2. 1 Macam-macam Diagram UML

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1. *Structure Diagram*

   Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

2. *Behavior Diagram*

   Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

3. *Interaction Diagram*

   Yaitu kumpulan diagram yang digunakan untuk menggambarkan interkasi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. Dalam pengembangan aplikasi permainan Edunvi, peneliti hanya akan menggunakan 4 macam diagram UML yaitu *Use Case Diagram, Class Diagram, Sequence Diagram dan Activity Diagram*. Pemilihan diagram ini

didasarkan pada tingkat *Learning Priority* yang tinggi serta dirasa cukup mewakili dari seluruh segmentasi pada diagram UML.

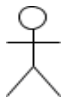
Notasi diagram-diagram tersebut adalah sebagai berikut :

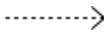
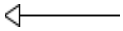
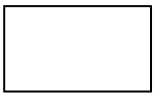
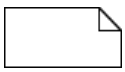
1. ***Use Case* Diagram**

   Use Case atau diagram *Use Case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Syarat penamaan pada *Use Case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *Use Case* yaitu pendefinisian apa yang disebut aktor dan *Use Case*. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. *Use Case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

   Berikut adalah simbol-simbol yang ada pada diagram *Use Case* (Edora, 2019).

Tabel 2. 2 simbol-simbol *diagram Use Case*

NO	GAMBAR	NAMA	KETERANGAN
1.		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *Use Case*.

NO	GAMBAR	NAMA	KETERANGAN
2.		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)*akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3.		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4.		*Extend*	Menspesifikasikan bahwa *Use Case* target memperluas perilaku dari *Use Case* sumber pada suatu titik yang diberikan.
5.		*Association*	Apa yang mengubungkan antara objek satu dengan objek lain
6.		*Include*	Menspesifikasikan bahwa *Use Case* sumber secara *eksplisit*.
7.		*Sistem*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9.		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan

2. *Activity Diagram*

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada

menggambarkan aktifitas sistem bukan apa yang dilakukan aktor, jadi aktifitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang ada pada diagram aktifitas (Edora, 2019).

Tabel 2. 3 simbol-simbol *diagram activity*

NO	GAMBAR	NAMA	KETERANGAN
1.		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2.		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3.		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4.		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5.		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

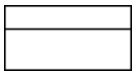
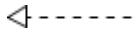
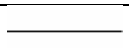
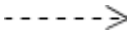
3. *Class Diagram*

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki apa yang disebut atribut dan metode atau operasi.

Berikut adalah simbol-simbol yang ada pada diagram kelas (Edora, 2019).

Tabel 2. 4 Simbol-Simbol *Diagram Class*

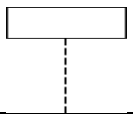
NO	GAMBAR	NAMA	KETERANGAN
1.		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).

NO	GAMBAR	NAMA	KETERANGAN
2.		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil terukur
5.		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya
7.		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

4. *Squence Diagram*

Diagram Squence menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Untuk menggambarkan *Diagram Squence* maka harus diketahui objek- objek yang terlibat dalam sebuah *Use Case* beserta metode-metode yang dimiliki kelas yang diinstanisasi oleh objek itu. Berikut adalah simbol-simbol yang ada pada *Diagram Squence* (Edora, 2019).

Tabel 2. 5 Simbol-Simbol *Diagram Squence*

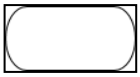
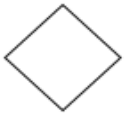
NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek entity, antarmuka yang saling berinteraksi.

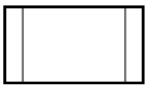
NO	GAMBAR	NAMA	KETERANGAN
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

J. *Flowchart*

Pengertian *Flowchart* (Diagram Alir) atau di sebut *Flowchart* merupakan bagan *(Chart)* yang mengarahkan alir *(flow)* di dalam prosedur atau program sistem secara logika. *Flowchart* adalah cara untuk menjelaskan tahap-tahap pemecahan masalah dengan merepresentasikan simbolsimbol tertentu yang mudah dipahamii, mudah digunakan dan standar. Tujuan penggunaan *Flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai dan rapi dengan menggunakan simbol-simbol yang standar yang dapat di mengerti oleh programer. Tahapan penyelesaian masalah yang disajikan harus tepat, sederhana, dan jelas (Syamsiah, 2019).

Tabel 2. 6 Simbol-Simbol *Flowchart*

Simbol	Nama dan Fungsi
	***Terminator*** (Tanda mulai atau selesai), menyatakan titik awal dan titik akhir diagram alir.
	**Proses** (Tanda komputasi), menyatakan sebarang proses.
	**Keputusan** (Pengambilan keputusan), digunakan untuk pengambilan keputusan. Dalam hal ini yang ada didalam symbol ini berupa suatu pertanyaan yang jawabannya berupa Ya atau Tidak.
	***Input-Output*** (Tanda masukan atau keluaran), menyatakan operasi pemasukan data atau penampilan data.

Simbol	Nama dan Fungsi
	**Proses Terdefinisi** (Tanda prosedur atau fungsi subalgoritma)
	**Konektor** (Tanda penghubung), digunakan untuk menghubungkan ke berbagai bagian dalam alir diagram.

K. *Blackbox Testing*

Blackbox Testing adalah tahap yang digunakan untuk menguji kelancaran program yang telah dibuat. Pengujian ini penting dilakukan agar tidak terjadi kesalahan alur program yang telah dibuat (Edora, 2019).

Blackbox Testing yaitu pengujian desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Nugroho dkk., 2018).

Blackbox Testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitamnya. Sama seperti pengujian *Black Box*, mengevaluasi hanya dari tampilan luar dan fungsionalitasnya. Tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detailnya (hanya mengetahui input dan output).

Black Box pengujian adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi yang bertentangan dengan struktur internal atau kerja. Pengetahuan khusus dari kode aplikasi/struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar

spesifikasi dan persyaratan, yaitu aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi 13 fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih input yang valid dan tidak valid dan menentukan output yang benar. Tidak ada pengetahuan tentang struktur internal benda uji itu.

Pengujian pada *BlackBox* berusaha menemukan kesalahan seperti:

a. Fungsi-fungsi yang tidak benar atau hilang
b. Kesalahan *interface*
c. Kesalahan dalam struktur data atau akses *database* eksternal
d. Kesalahan kinerja, inisialisasi dan kesalahan terminasi (Arfamaini, 2016)

L. *Whitebox Testing*

Whitebox Testing adalah salah satu cara untuk menguji suatu aplikasi atau *software* dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang di buat ada yang salah atau tidak. Kalau modul yang telah dan sudah di hasilkan berupa output yang tidak sesuai dengan yang di harapkan maka akan di compile ulang dan di cek kembali kode-kode tersebut hingga mencapai sesuai denganyang di harapkan (Nugroho dkk., 2018).

WhiteBox merupakan Proses pengujian perangkat lunak yang didasarkan dari segi desain dan kode program, berdasarkan spesifikasi fungsi-fungsi, masukan dan keluaran yang dihasilkan. Dalam pengujian ini akan dicek berdasarkan logic program

Ada beberapa Kelebihan pengujian *WhiteBox Testing* adalah:

1. Kesalahan logika.

   Digunakan pada sitanks "*if*" dan pengulangan. Dimana *White Box Testing* akan mendeteksi kondisi-kondisi yang tidak sesuai dan mendeteksi kapan proses pengulangan akan terhenti.

2. Ketidaksesuaian asumsi.

   Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk dianalisa dan diperbaiki.

3. Kesalahan ketik.

   Mendeteksi bahasa pemrograman yang bersifat *case sensitive*. (Nugroho & Komarudin, 2018)

M. Kajian Hasil Penelitian Terdahulu

Tinjauan Hasil Penelitian Sebelumnya Tentang *Game Puzzle*

Tabel 2. 7 Hasil Penelitian Terdahulu

NO	JUDUL	NAMA/ TAHUN	METODE
1.	*Game* Tebak Gambar Untk Melatih Daya Ingat Anak Berbasis Android	Ahmad Fitrah Ramadhan Paisal/2020	Metode pembuatannya menggunakan bahasa pemrograman PHP dan *Javascript*
2.	*Game* Online Duel Matematika Berbasis *Web*	Tri hariadi purnomo/2020	Metode pembuatannya menggunakan bahasa pemrograman HTML5, CSS3, *Javascript*, dan MySql
3.	Pembuatan *Game* Rey-Man Menggunakan Construct 2 Berbasis Android	Dedi sumardi/2020	Metode pembuatannya menggunakan bahasa pemrograman HTML5, CSS3, *Javascript*, dan MySql
4.	Game Edukasi Puzzle Untuk Pengenalan Alat Musik Tradisional Di	Hasma Sadda/2021	perancangan sistem informasi terstruktur diimplementasikan

	Indonesia Berbasis Web		menggunakan bahasa pemograman *PHP* dan *MySQL* sebagai databasenya.

Melihat dari sejumlah judul dan tema yang berkaitan dengan penelitian yang dilakukan, diperoleh persamaan dan perbedaan dari penelitian yang akan dilakukan. Tujuannya membuktikan bahwa penulisan tugas akhir ini asli dan bukan duplikasi dari tugas akhir penelitian lain, seperti berikut ini :

"*Game Puzzle* Pengenalan Tokoh Pahlawan Revolusi Indonesia". Aplikasi ini berbasis android menggunakan algoritma *Fisher Yates Shuffle*. Perancangan aplikasi *Game Puzzle* pengenalan tokoh pahlawan revolusi indonesia menggunakan bahasa pemrograman *Javascript*, permainan ini ditampilkan dalam bentuk visual 2D, dan aplikasi yang dipakai untuk membuat *Game Puzzle* pengenalan tokoh pahlawan revolusi indonesia adalah *texteditor* serta teknologi lain seperti CSS, HTML, MySQL dan *Apache Cordova* . Game puzzle ini mempunyai 3 model permainan yaitu, sledding puzzle, square puzzle dan jigsaw puzzle dan mempunyai 2 level yaitu mudah dan sulit dilengkapi dengan informasi dari setiap tokoh pahlawan dan mempunyai fitur membaca otomatis menggunakan suara. Aplikasi yang dihasilkan dapat memberikan informasi mengenai tokoh pahlawan Revolusi indonesia dengan cara yang berbeda dan dapat digunakan dalam proses pembelajaran.

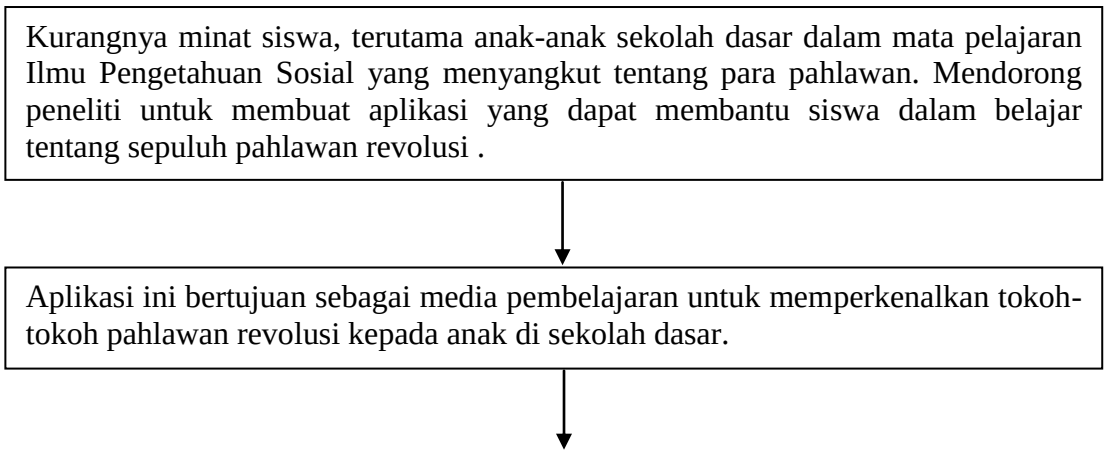
"*Game* Tebak Gambar Untuk Melatih Daya Ingat Anak Berbasis Android". Aplikasi ini bertujuan sebagai media untuk melatih kemampuan daya ingat dan konsentrasi anak. *Game* tebak gambar ini dibuat menggunakan bahasa pemrograman PHP dan *Javascript*, serta teknologi lain seperti CSS, HTML5,

MySQL dan *Apache Cordova*. Berdasarkan hal tersebut diatas maka perbedaan penelitian terdahulu dengan penelitian penulis adalah metode dan pembuatan serta fitur aplikasi. Game ini mempunyai 1 model permainan puzzle dengan model square puzzle

"Game Edukasi Puzzle Untuk Pengenalan Alat Musik Tradisional Di Indonesia Berbasis Web". aplikasi *Game* edukasi *Puzzle* untuk pengenalan alat musik tradisional di indonesia berbasis web. Metode penelitian yang digunakan meliputi pengumpulan data dengan interview, dokumentasi dan observasi, sedangkan metode pembangunan sistem menggunakan model pengujian *BlackBox* dan *WhiteBox*, alat bantu analisis dan perancangan sistem informasi terstruktur diimplementasikan menggunakan bahasa pemograman *PHP* dan *MySQL* sebagai databasenya.

N. Kerangka Pikir

Untuk lebih memperjelas kerangka pikir maka digambarkan dalam betuk diagram sebagai berikut :



Peneliti mengusulkan membuat sebuah aplikasi permainan *puzzle* tentang pahlawan revolusi sebagai media pembelajaran . Peneliti merancang aplikasi dengan metode kuantitatif, yaitu suatu metode yang bertujuan untuk mendapatkan gambaran yang jelas tentang hal-hal yang diperlukan tahapan-tahapannya serta membuatnya menggunakan teknologi *Texteditor* dan Algoritma pengacakan *Fisher-Yates Shuffle.*



Hasil dari aplikasi ini memiliki 2 level yaitu, mudah dan sulit, serta mempunyai 10 pilihan tokoh pahlawan revolusi Indonesia dan mempunyai 3 model bentuk yaitu sliding,square, dan jigsaw .

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Metodologi penelitian yang digunakan untuk membangun aplikasi *Game Puzzle* adalah menggunakan analisis kuantitatif, yaitu suatu metode yang bertujuan untuk mendapatkan gambaran yang jelas tentang hal-hal yang diperlukan tahapan-tahapannya. Kuantitatif adalah proses penelitian, hipotesis atau prediksi hasil, data empiris, analisis data, dan kesimpulan data sampai dengan hasil akhir penulisannya menggunakan aspek pengukuran, perhitungan, rumus dan data numerik, atau perhitungan statistik. (Rukminingsih.,dkk 2020).

B. Teknik Pengumpulan Data

Pengumpulan data Metode penelitian pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut :

a. Observasi

   Observasi adalah metode pengumpulan data dengan mengadakan penelitian dan peninjauan langsung terhadap permasalahan yang diambil, agar penyusun dapat mengetahui apa saja yang terdapat pada *Game* bertipe *Puzzle*.

b. Studi literatur

   Studi literatur adalah metode pengumpulan data dengan cara mengumpulkan literatur, jurnal, paper dan bacaan - bacaan yang diperlukan untuk membangun

   aplikasi *Game Puzzle*.

c. Studi literatur

Studi literatur adalah metode pengumpulan data dengan cara mengumpulkan literatur, jurnal, paper dan bacaan-bacaan yang diperlukan untuk membangun aplikasi *Game Puzzle*.

d. Studi Pustaka

Pengumpulan data dengan menggunakan atau mengumpulkan sumber-sumber tertulis, dengan cara membaca, mempelajari dan mencatat hal-hal penting yang berhubungan dengan masalah yang sedang dibahas guna memperoleh gambaran secara teoritis.

C. Lokasi dan Waktu

Lokasi penelitian dilaksanakan di Desa Kariango,Pananrang, Kec. Mattiro Bulu, Kabupaten Pinrang, Sulawesi Selatan 91271. Waktu yang dipergunakan untuk pelaksanaan penelitian ini berlangsung selama -+ 2 bulan.

D. Metode Pengolahan Data

Metode pengujian aplikasi ini menggunakan *System Usability Testing* (SUS). Pengujian ini dilakukan untuk mengetahui kepuasan peserta dalam menggunakan aplikasi, dapat mengakses aplikasi secara efektit, dapat menggunakan aplikasi secara efisien.

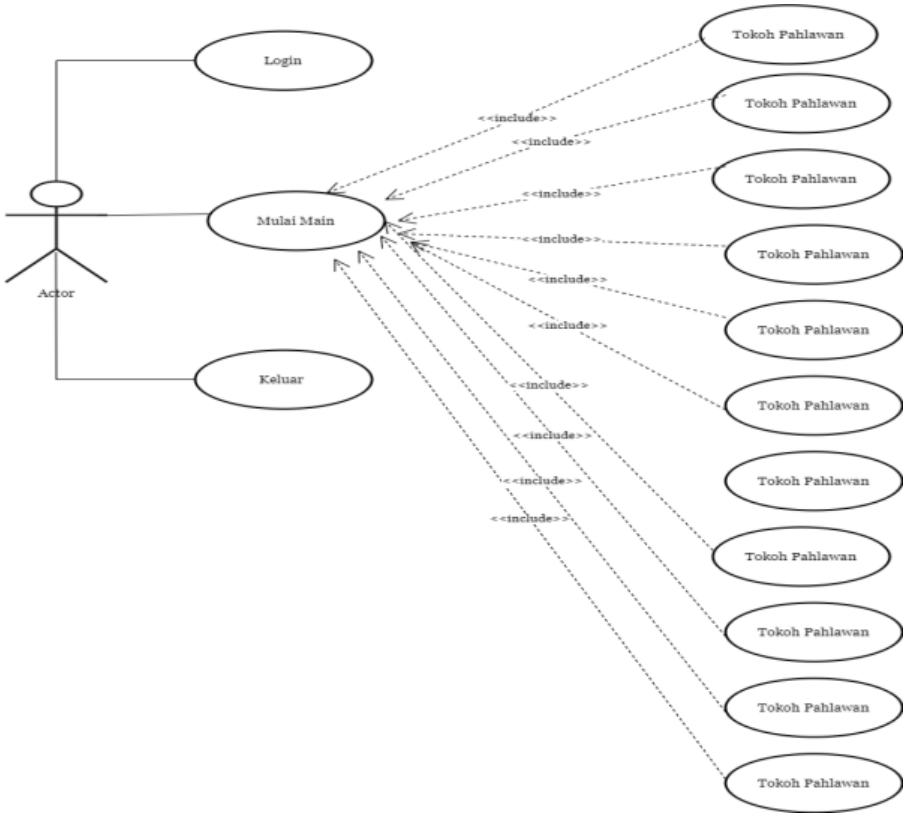
E. Rancangan Sistem

Rancangan *Game Puzzle* disajikan menggunakan *Unified Modelling Language* (UML). *Use Case* diagram ini digunakan untuk menggambaran hubungan sejumlah external aktor dengan *Use Case* yang terdapat dalam sistem.

Use Case ini hanya menggambarkan keadaan lingkungan sistem yang dapat dilihat dari luar aktor.

Dalam Penelitian ini, Penulis menyajikan rancangan sistem menggunakan 3 diagram, yaitu Diagram *Use Case*, Diagram Activity dan Diagram Sequence.

1. *Use Case* Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Menggambarkan kebutuhan sistem dari sudut pandang user/Pemain.



Gambar 3. 1 *Use Case Diagram Game Puzzle*

BAB IV

HASIL DAN PEMBAHASAN

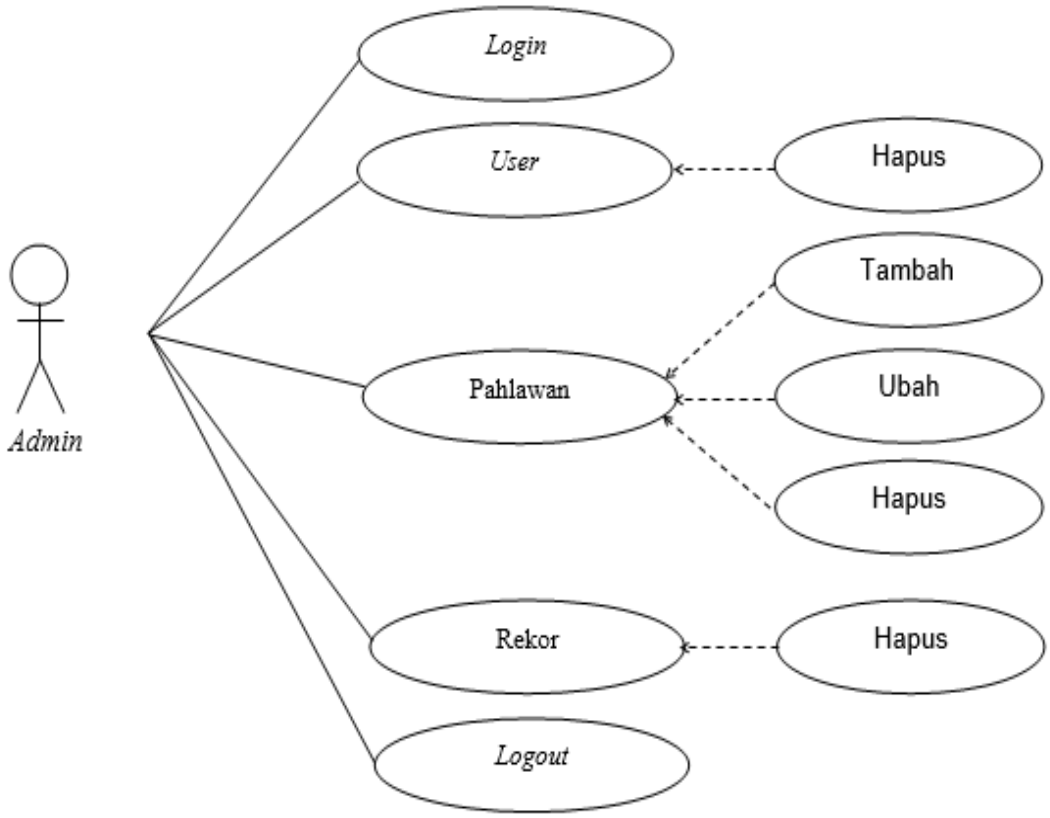
A. Rancangan Aplikasi

Dalam analisis sistem ini, penulis menggunakan *Use Case* Diagram, *Activity* Diagram, *Sequence* Diagram dan Class Diagram.

1. *Use Case Diagram*

   Use Case Diagram berfungsi untuk menjalankan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor*).

   a. Aktor *Admin*

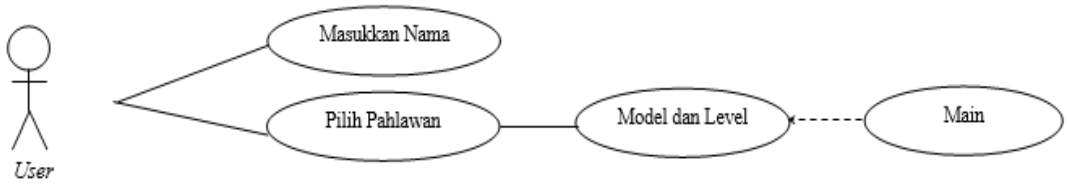


Gambar 4. 1 *Use Case* Diagram *Admin*

Tabel 4. 1 Penjelasan *Use Case* Diagram Aktor *Admin*

Nama *Use Case*	**Deskripsi *Use Case***
Login	Merupakan proses *admin* untuk masuk halaman utama.
User	Merupakan halaman pengelolaan data *user*.
Pahlawan	Merupakan halaman pengelolaan data pahlwan.
Rekor	Merupakan halaman pengelolaan data rekor.
Pelacakan	Merupakan *form* pengelolaan lokasi mobil.
Tambah	Merupakan proses *admin* menyimpan data baru.
Ubah	Merupakan proses *admin* mengubah data.
Hapus	Merupakan proses *admin* menghapus data.
Logout	Merupakan proses *admin* keluar dari *form* utama aplikasi.

b. Aktor *User*



Gambar 4. 2 *Use Case* Diagram *User*

Tabel 4. 2 Penjelasan *Use Case* Diagram Aktor *User*

Nama *Use Case*	**Deskripsi *Use Case***
Masukkan Nama	Merupakan proses *user* memasukkan nama.
Pilih Pahlawan	Merupakan proses *user* memilih pahlawan pada *form utama*.
Model dan *Level*	Merupakan proses *user* memilih model dan level permainan pada *popup show.*
Main	Merupakan *form* memulai permainan.

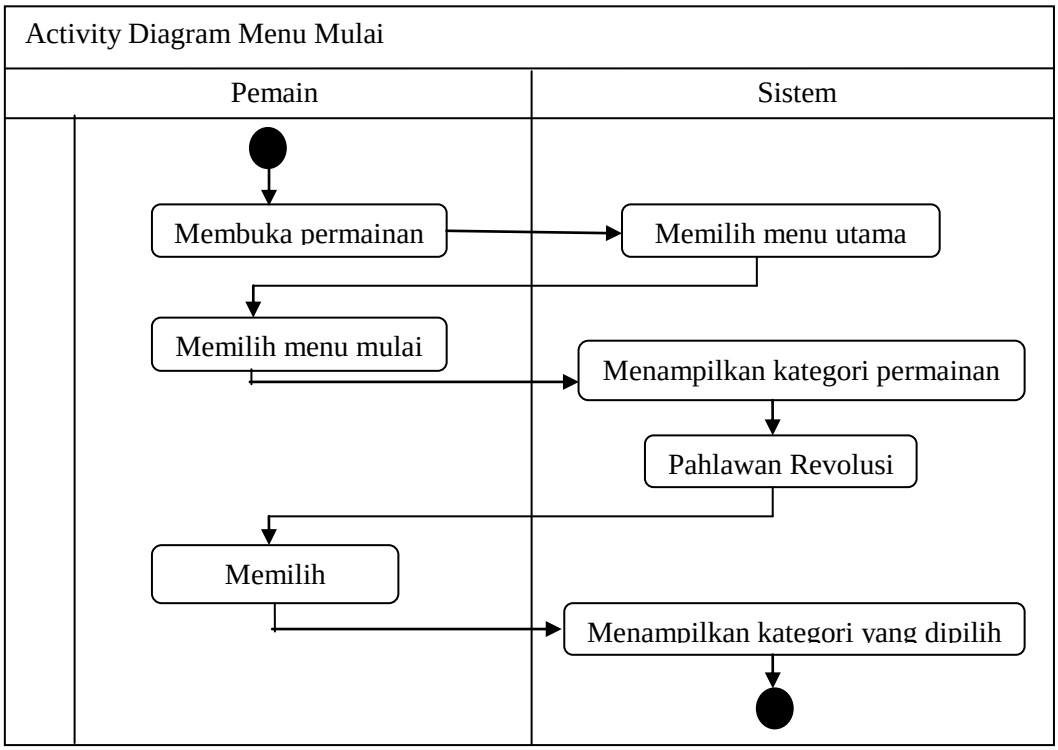
2. Activity Diagram

Activity Diagram ini menjelaskan tentang aktivitas-aktivitas yang terjadi dalam sebuah aliran proses pada sistem.

a. *Activity* Diagram Menu Kategori

Gambar 4.3 menunjukkan bahwa aktivitas Pemain terhadap sistem dalam Menu Mulai. Sistem menanpilkan kategori permainan *Puzzle* yaitu Pahlawan Revolusi

Berikut *Activity* Diagram Menu Mulai:



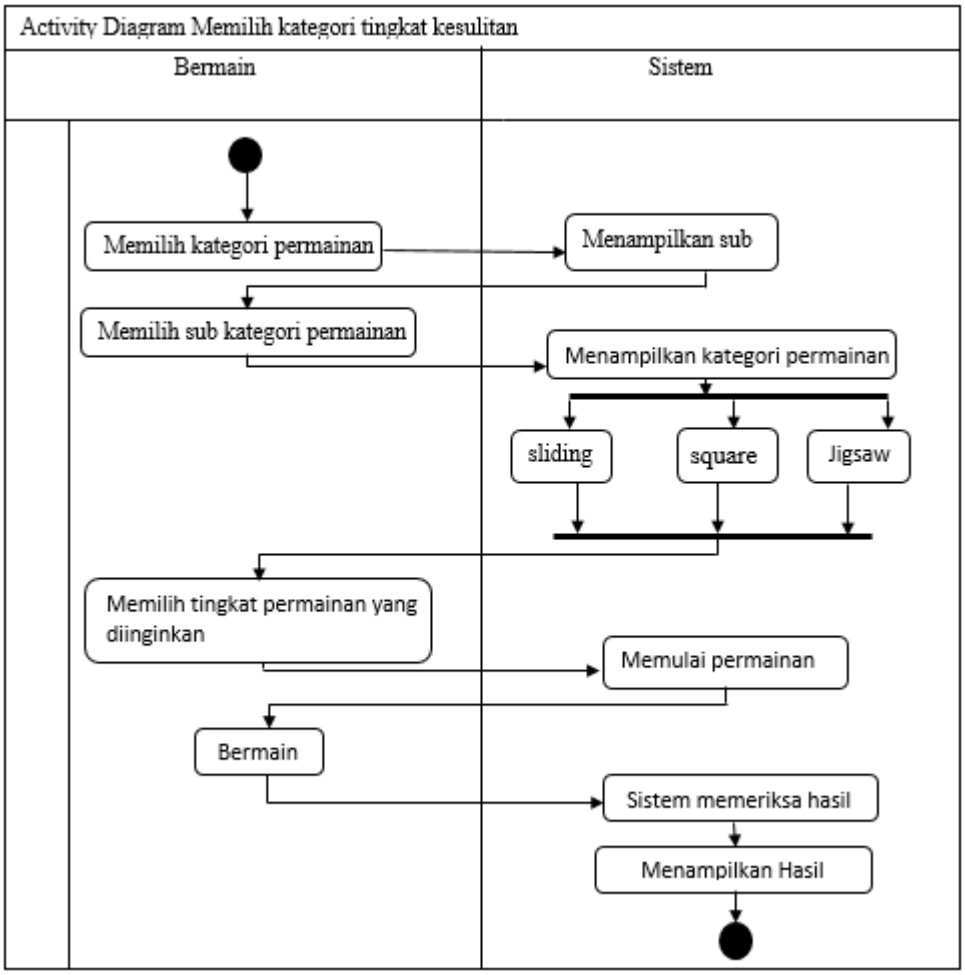
Gambar 4. 3 *Activity* Diagram Menu Kategori

b. *Activity Diagram Kategori Model Puzzle*

Gambar 4.4 menggambarkan ketika Pemain melakukan aktivitas terhadap

sistem saat memilih kategori, maka sistem akan menanpilkan semua 3 kategori bentukpermainan yaitu: *sliding, square, dan jigsaw puzzle.*

Berikut *Acrivity* Diagram Memilih Menu Kategori model *Puzzle*:



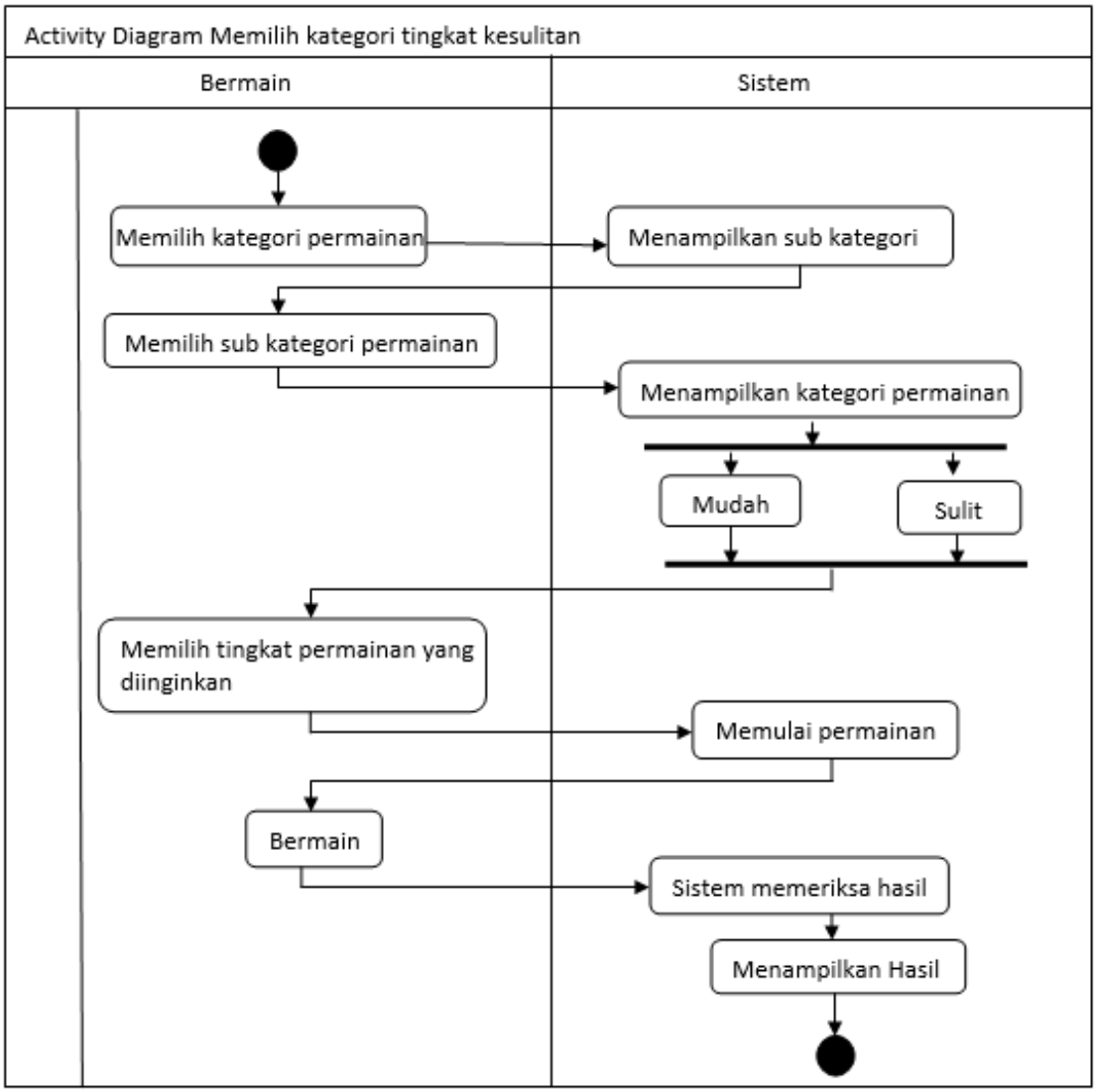
Gambar 4. 4 *Activity* Diagram Menu Kategori

c. *Activity* Diagram Kategori Tingkat Kesulitan

Gambar 4.5 menggambarkan ketika Pemain melakukan aktivitas terhadap

sistem saat memilih kategori, maka sistem akan menanpilkan semua 2 kategori kesulitan permainan yaitu: *medium, hard*.

Berikut *Acrivity* Diagram Memilih Menu Kategori Tingkat Kesulitan



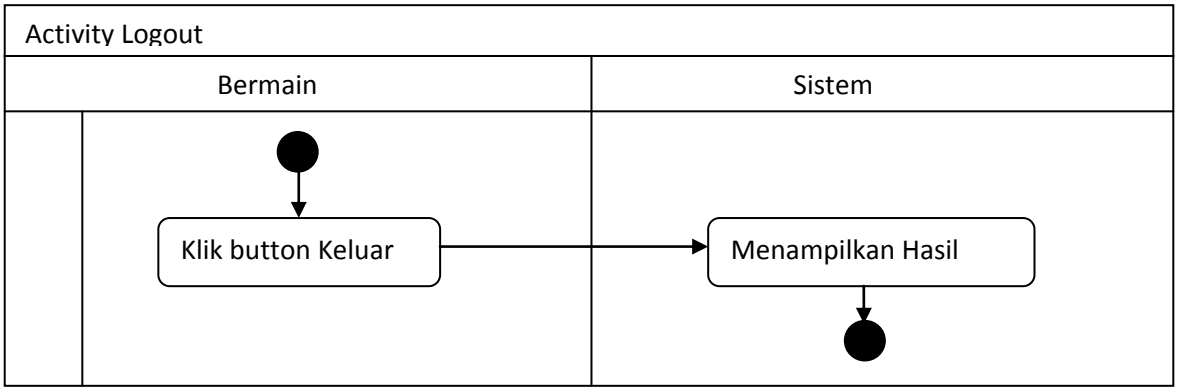
Gambar 4. 5 *Activity* Diagram Kategori Tingkat Kesulitan

d. *Activity* Diagram Keluar

Gambar 4.6 menggambarkan ketika Pemain melakukan aktivitas terhadap

sistem saat memilih menu keluar, maka sistem akan menanpilkan pilihan apakah anda ingin keluar (Ya/Tidak).

Berikut *Acrivity* Diagram Memilih Menu Keluar:



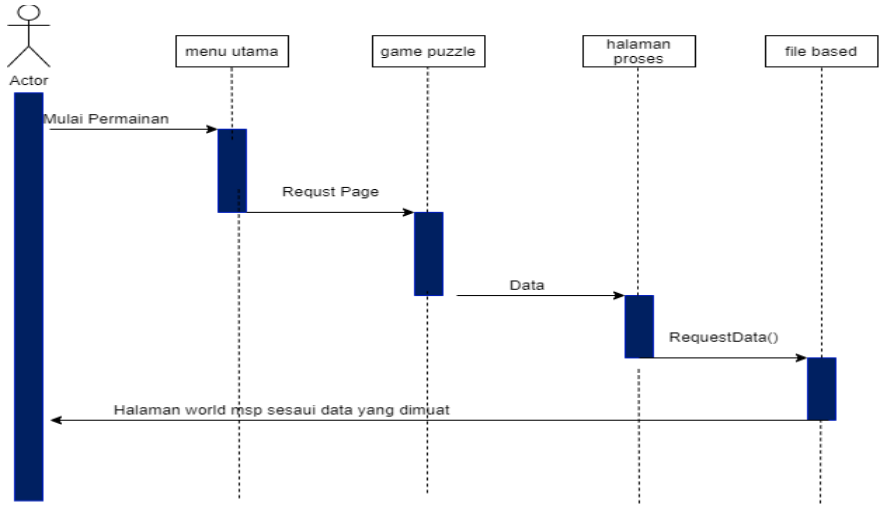
Gambar 4. 6 *Activity* Diagram Keluar Gambar

3. Diagram *Squence*

Sequence Diagram merupakan aliran antara objek yang membentuk proses, berikut adalah diagram *Sequencenya.*

a. *Diagram Squence* Menu Utama

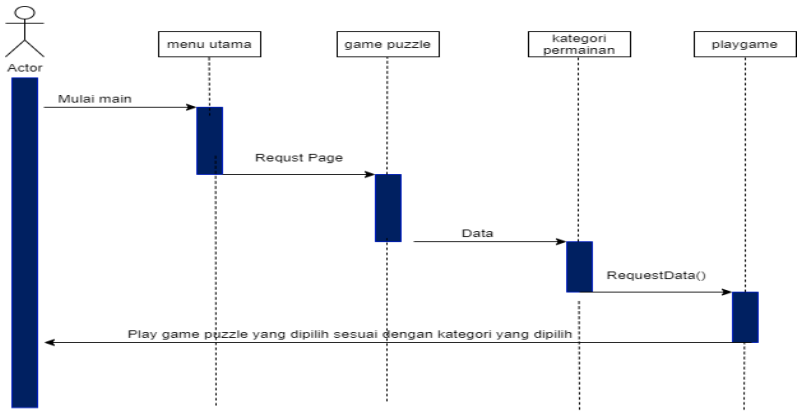
Sequence diagram menu utama untuk menggambarkan keseluruhan objek yang terdapat pada *Game Puzzle*. Keseluruhan *sequence diagram* menu utama dapat dilihat pada Gambar 4.7



Gambar 4. 7 *Diagram Squence* Menu Utama

b. *Diagram Squence* Menu Kategori

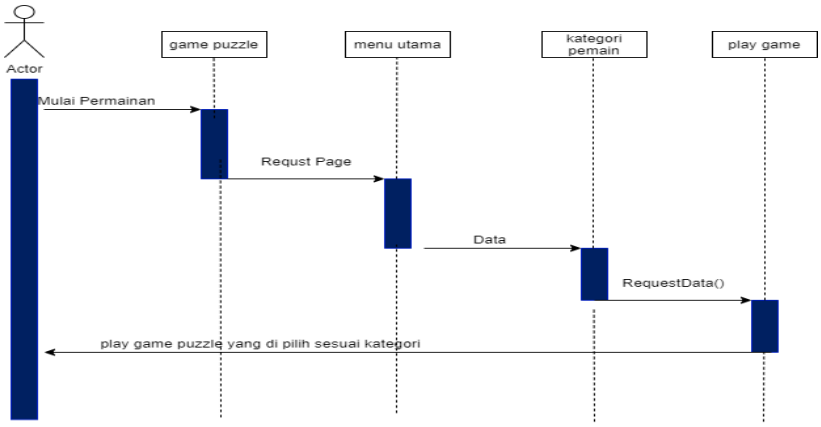
Squence diagram menu kategori untuk menentukan jenis permainan *Game* yang akan dimainkan pada *Game Puzzle*. Keseluruhan *sequence diagram* menu utama dapat dilihat pada Gambar 4.8



Gambar 4. 8 *Diagram Squence* Menu Kategori

c. *Diagram Squence* Menu Mulai

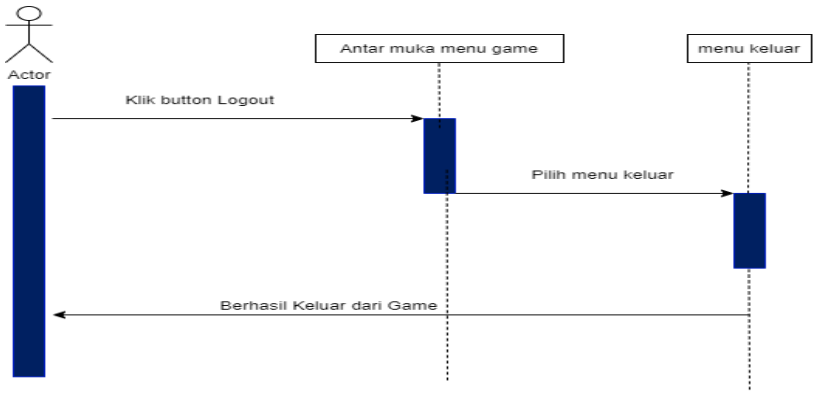
Sequence diagram menu mulai untuk memulai permainan *Game* yang akan dimainkan pada *Game Puzzle*. Keseluruhan *sequence diagram* menu mulai dapat dilihat pada Gambar 4.9



Gambar 4. 9 *Diagram Squence* Menu Mulai

d. *Diagram Squence* Menu Keluar

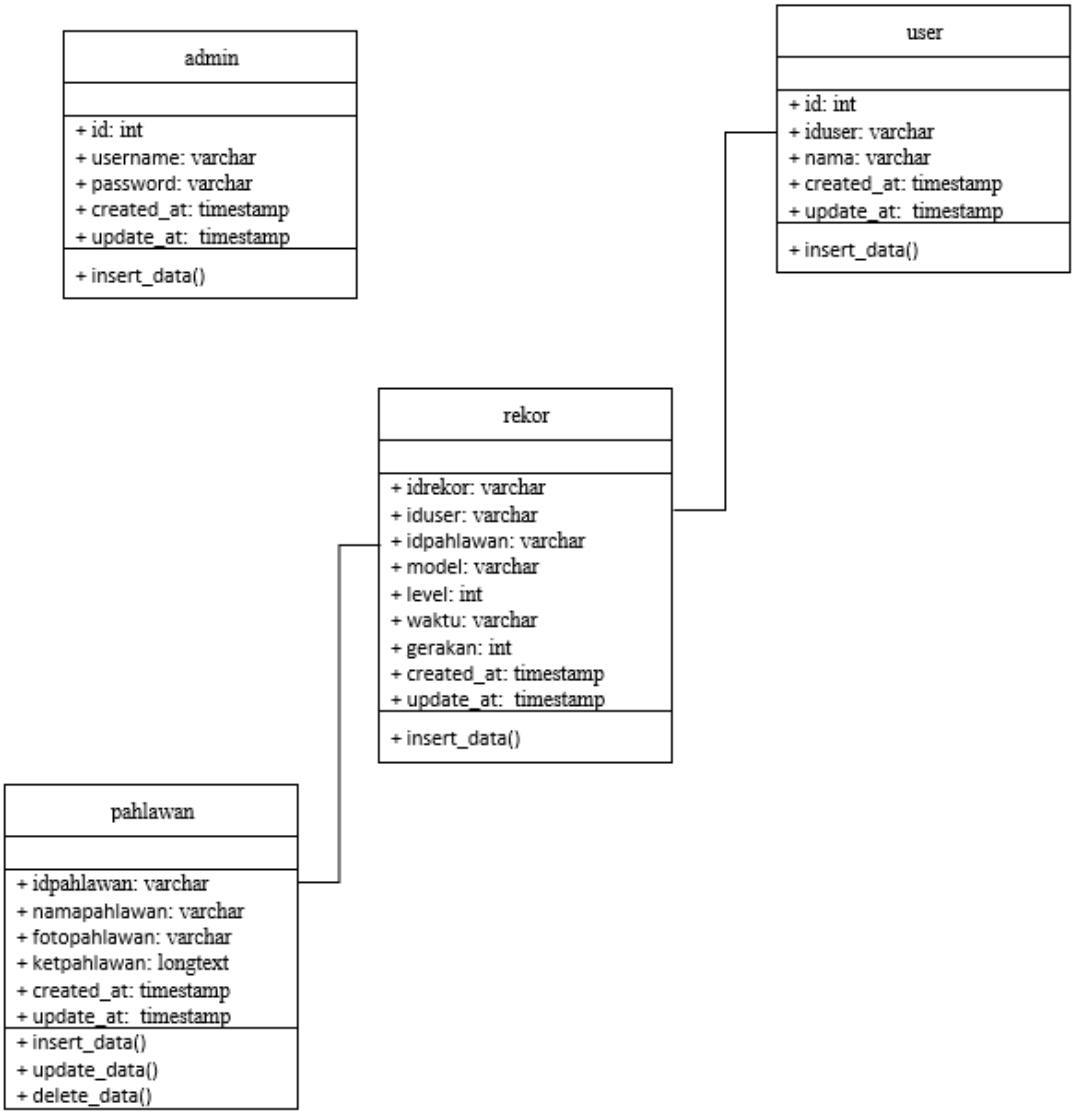
Sequence diagram menu keluar untuk keluar dalam permainan *Game Puzzle*. Keseluruhan *sequence diagram* menu keluar dapat dilihat pada Gambar 4.10



Gambar 4. 10 *Diagram Squence* Menu Keluar

4. *Class* Diagram

Diagram kelas atau *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. *Class* diagram menunjukkan hubungan antara *Class* dalam sistem bagaimana mereka saling berkolaborasi dalam mencapai tujuan. Pada aplikasi ini struktur *Class* membantu dalam visualisasi dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap *Class* yang didalamnya terdapat obyek dan atribut. Berikut ini adalah perancangan struktur sistem yang terdapat pada *Game Puzzle.*



Gambar 4. 11 Class Diagram

B. Implementasi Sistem

Implementasi *Game Puzzle* dilakukan untuk menguji apakah sistem dapat berjalan seperti yang didefinisikan pada perancangan system. Dalam Penelitian ini, penulis melakukan implementasi pada perangkat keras yang memiliki spesifikasi yang sama dengan perangkat keras untuk membangun sistem. Implementasi yang dilakukan merupakan implementasi antar muka sistem.

C. Tampilan Aplikasi

Tampilan aplikasi akan menjelaskan secara *visual* bagaimana tampilan dari aplikasi *Game Puzzle* pengenalan tokoh pahlawan revolusi indonesia

1. Tampilan Halaman Utama

   Antarmuka Halaman Utama merupakan tampilan utama dari aplikasi *Game Puzzle*. Pada Halaman Utama terdapat fungsionalitas yang dapat diakses oleh Pemain yaitu menu main, dan keluar. Tampilan Menu Halaman Utama pada *Game Puzzle* adalah sebagai berikut:



Gambar 4. 12 Tampilan Halaman Utama

2. Tampilan Menu Kategori

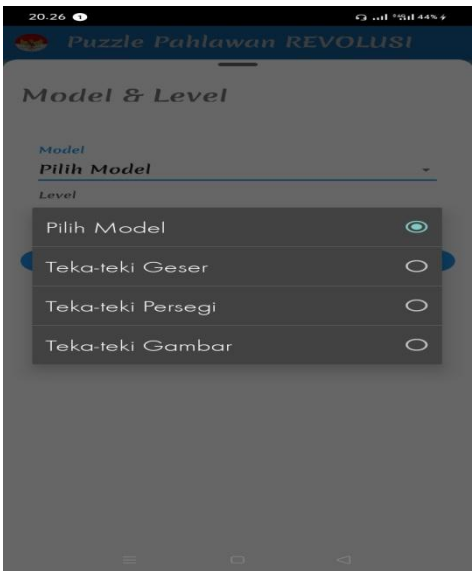
   Gambar 4.13 Menu Kategori menunjukkan bahwa ketika pemain memilih menu mulai maka akan tampil kategori-kategori *Game Puzzle* yang dapat dipilih sesuai dengan keinginan pemain. Tampilan Menu Kategori pada *Game Puzzle* adalah sebagai berikut:



Gambar 4. 13 Tampilan Menu Kategori

3. Tampilan Kategori Model

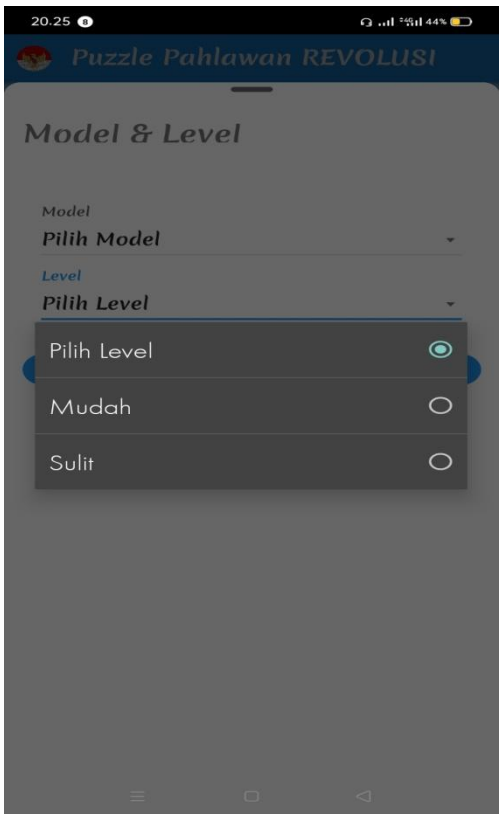
Gambar 4.14 menjelaskan ketika Pemain memilih salah satu kategori. Disini pemain dapat memilih kategori model bentuk *Puzzle* yang ingin pemain mainkan. Pada saat pemain memilih model square *Puzzle* , maka *Game Puzzle* akan langsung dimulai sesuai pilihan pemain. Tampilan menu Model pada *Game Puzzle* adalah sebagai berikut:



Gambar 4. 14 Tampilan Kategori Model

4. Tampilan Kategori Tingkat Kesulitan

Gambar 4.15 menjelasakan ketika Pemain memilih salah satu kategori. Disini pemain dapat memilih kategori tingkat kesulitan yang ingin pemain untuk mainkan *Game*. Pada saat pemain memilih tingkat kesulitan, maka *Game Puzzle* akan langsung dimulai sesuai pilihan pemain. Tampilan menu Tingkat Kesulitan pada *Game Puzzle* adalah sebagai berikut:



Gambar 4. 15 Tampilan Kategori Tingkat Kesulitan

5. Tampilan Kategori tokoh pahlawan

Gambar 4.16 menjelasakan ketika Pemain memilih kategori tokoh pahlawan yang diinginkan dengan tingkat kesulitan mudah. Disini pemain memiliki waktu untuk menyelesaikan permainan *Game Puzzle* yang dipilih dengan cara menggeser *Puzzle*, sehingga *Puzzle* tersusun. Menu *reset* pada

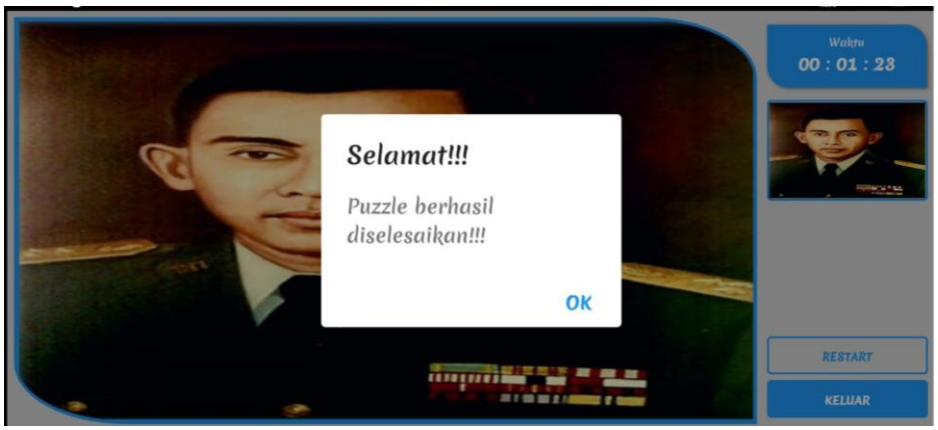
Game berfungsi untuk mengembalikan *Puzzle* yang diacak atau untuk membantu pemain menyelesaikan permainan *Puzzle*. Tampilan menu Kategori tokoh pahlawan pada *Game Puzzle* adalah sebagai berikut:



Gambar 4. 16 Permainan Game Puzzle

6. Tampilan Kategori Tokoh Pahlawan Berhasil

Gambar 4.17 menjelasakan ketika Pemain berhasil menyusun *Puzzle* sesuai dengan waktu yang telah di tentukan atau menyelesaikan *Game* sebelum waktu berakhir. Tampilan *Game* Berhasil berisi tentang penjelasan gambar *Puzzle* yang telah disusun pemain. Tampilan *Game* Berhasil pada *Game Puzzle* adalah sebagai berikut:



Gambar 4. 17 Tampilan Kategori Tokoh Pahlawan Berhasil

7. Tampilan *Game* Gagal

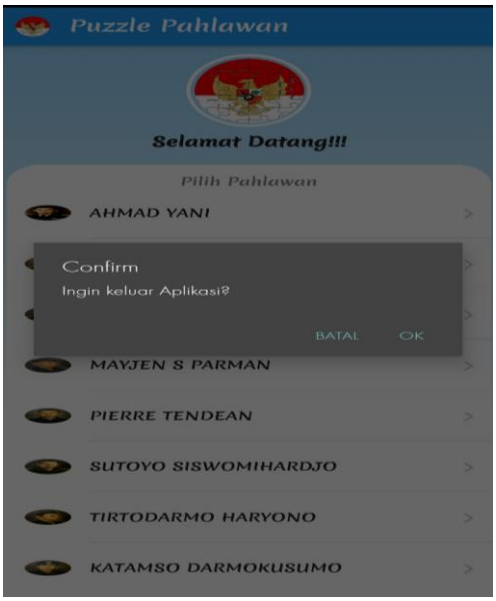
Gambar 4.18 menunjukkan bahwa antar muka *Game* Gagal merupakan tampilan *Game* Gagal ketika Pemain tidak berhasil menyelesaikan *Game Puzzle* sesuai dengan waktu yang sudah ditentukan pada *Game*. Tampilan *Game* Gagal pada *Game Puzzle* adalah sebagai berikut:



Gambar 4. 18 Tampilan Game Gagal

8. Tampilan *Game* Menu Keluar

Gambar 4.19 menunjukkan bahwa antar muka *Game* Menu Keluar merupakan tampilan atau pesan yang tampil ketika pemain memilih Menu Keluar. Tampilan *Game* Gagal pada *Game Puzzle* adalah sebagai berikut:



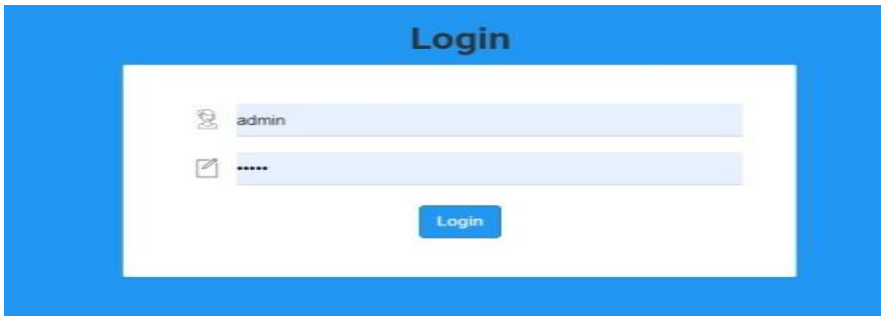
Gambar 4. 19 Tampilan Keluar

D. Tampilan Aplikasi Pada Admin

Tampilan aplikasi pada admin akan menjelaskan secara visual bagaimana tampilan dari aplikasi *Game Puzzle* pengenalan tokoh pahlawan revolusi indonesia pada admin.

1. Tampilan Halaman *Login* Pada Admin

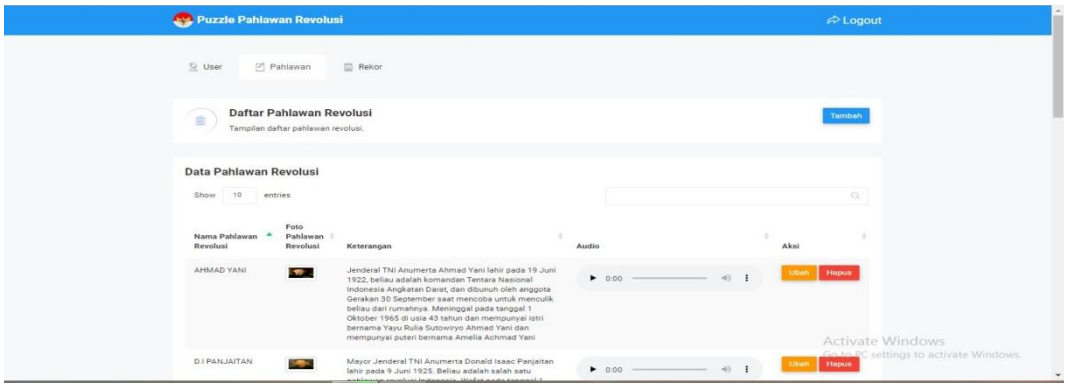
   Rancangan Halaman *Login* Admin merupakan rancangan yang menampilkan halaman yang berisikan *form* untuk *login* admin.



Gambar 4. 20 Tampilan Halaman Login Pada Admin

2. Tampilan Halaman *Dashboard* Pada Admin

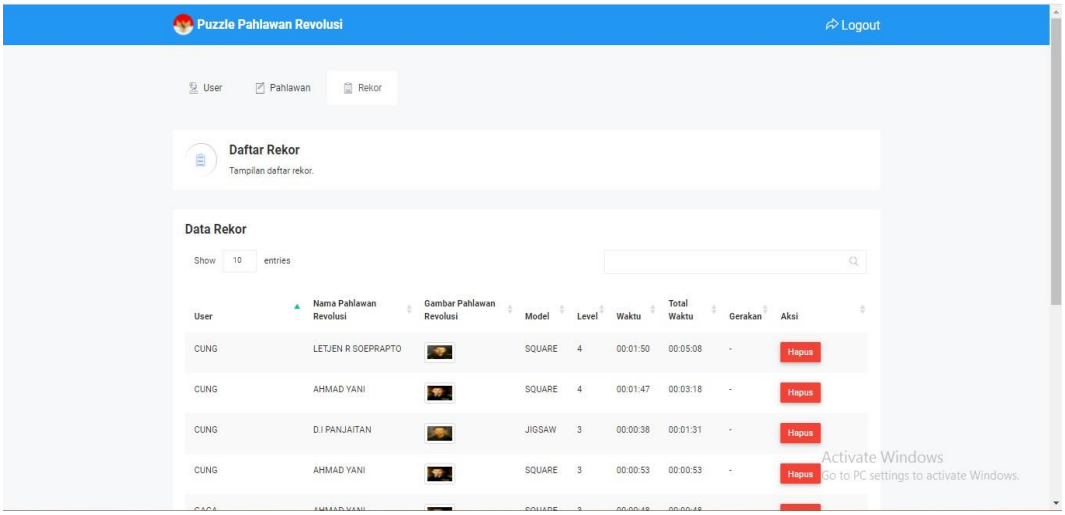
   Tampilan Halaman *Dashboard* pada admin merupakan rancangan yang menampilkan halaman yang berisi menu ‘Data *User*’, ‘Data Pahlawan’, ‘Data Rekor’ dengan fasilitas tambah penilaian untuk menambah kategori dari setiap kriteria, *edit* untuk mengubah data, dan *delete* untuk menghapus data.



Gambar 4. 21 Tampilan Halaman Dashboard Pada Admin

3. Rancangan Halaman Data Rekor Pada Admin

Rancangan Halaman Data *Rekor* merupakan rancangan yang menampilkan halaman yang berisikan tabel data *rekor* yaitu informasi user, nama pahlawan,gambar pahlawan,model,level, waktu, dan aksi. Dengan fasilitas hapus dan *button 'Logout'* untuk keluar.



Gambar 4. 22 Tampilan Halaman Data Rekor Pada Admin

E. Pengujian Sistem

Tahap ini adalah tahap dimana *Game Puzzle* pengenalan tokoh pahlawan revolusi indonesia ini di periksa. Tahap ini lebih ke pemeriksaan tombol navigasi apakah ada kesalahan pemberian perintah, navigasi tidak bisa diakses, pengecekan kesalahan pada materi, kesalahan penulisan. Tahap ini diperiksa oleh pembuat untuk memastikan desain tampilan, pemograman, dan materi pembelajaran sudah sesuai dan layak digunakan sebagai media pembelajaran.

1. Pengujian Fungsi Umum Aplikasi

Berikut tabel pengujian pada fungsi umum aplikasi *Game Puzzle*, ditunjukkan pada tabel berikut:

Tabel 4. 3 Pengujian Fungsi Umum

Nama Pengujian	Bentuk Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
Pengujian Menu Main	Menyentuh Tombol Main	Tampil halaman kategori *Game*	Berhasil
Pengujian Menu Keluar	Menyentuh tombol menu keluar	Tampilan halaman keluar	Berhasil

2. Pengujian Halaman Mulai Main

Berikut tabel pengujian halaman Mulai Main pada *Game Puzzle*. Ditunjukkan pada tabel 4.4

Tabel 4. 4 Pengujian Halaman Mulai Main

Nama Pengujian	Bentuk pengujian	Hasil yang diharapkan	Hasil pengujian
Pengujian tombol kategori tokoh pahlawan revolusi jendral Ahmad Yani	Menyentuh tombol	Tampil halaman *Game* mulai main	Berhasil
Pengujian tombol kategori *restart*	Menyentuh tombol *restart* pada *Game*	Mengacak game puzzle	Berhasil
Pengujian tombol kategori keluar	Menyentuh tombol keluar pada *Game*	Tampil halaman awal *Game Puzzle*	Berhasil

3. Pengujian Halaman Menu Keluar

Berikut tabel pengujian halaman menu keluar pada *Game Puzzle*. Ditunjukkan pada tabel 4.5

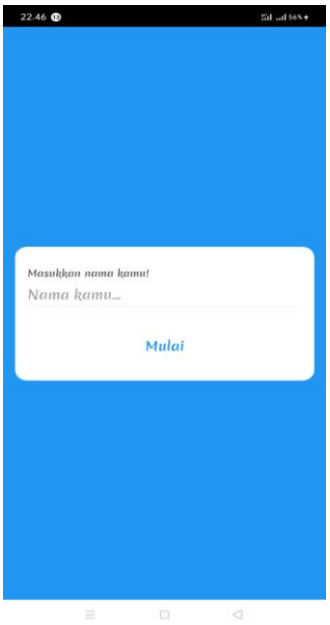
Tabel 4. 5 Pengujian Halaman Menu Keluar

Nama Pengujian	Bentuk pengujian	Hasil yang diharapkan	Hasil pengujian
Pengujian cara menu keluar	Menyentuh tombol *Keluar*	Tampil halaman pilihan menu keluar keluar (Ya/Tidak)	Berhasil

1. **Pengujian *BlackBox***

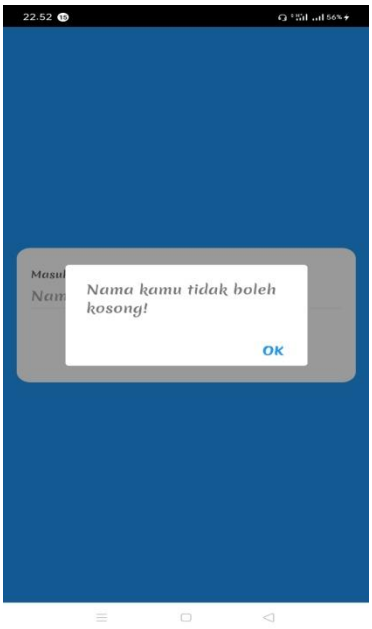
Pengujian sistem dilakukan dengan cara pengujian *Black Box*

Tabel 4. 6 BlackBox Halaman Login

Test Faktor	Hasil	Kesimpulan
Halaman *Login* aplikasi, apabila pertama kali membuka aplikasi.	✓	Informasi, tampil halaman *Login*.
Screen Shot		
		

Tabel 4. 7 BlackBox Kesalahan Username

Test Faktor	Hasil	Kesimpulan
Jika salah memasukkan *Username*	✓	Informasi, tampil *Popup* pesan gagal *Login* dan tidak ada terjadi proses.
Screen Shot		

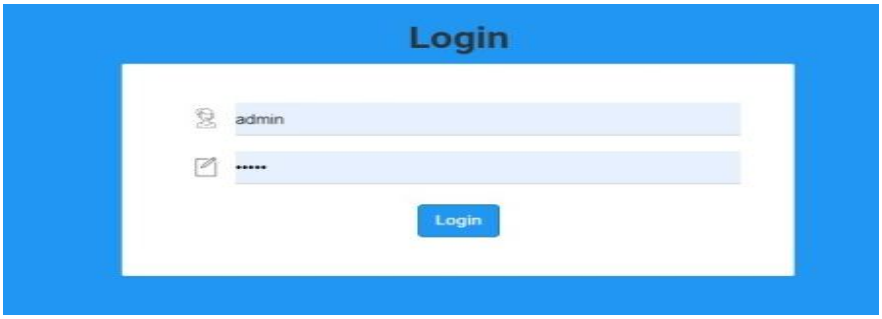


Tabel 4. 8 BlackBox Halaman Daftar

Test Faktor	Hasil	Kesimpulan
Jika *User* menekan daftar disini.	✓	Informasi, tampil halaman daftar.
Screen Shot		



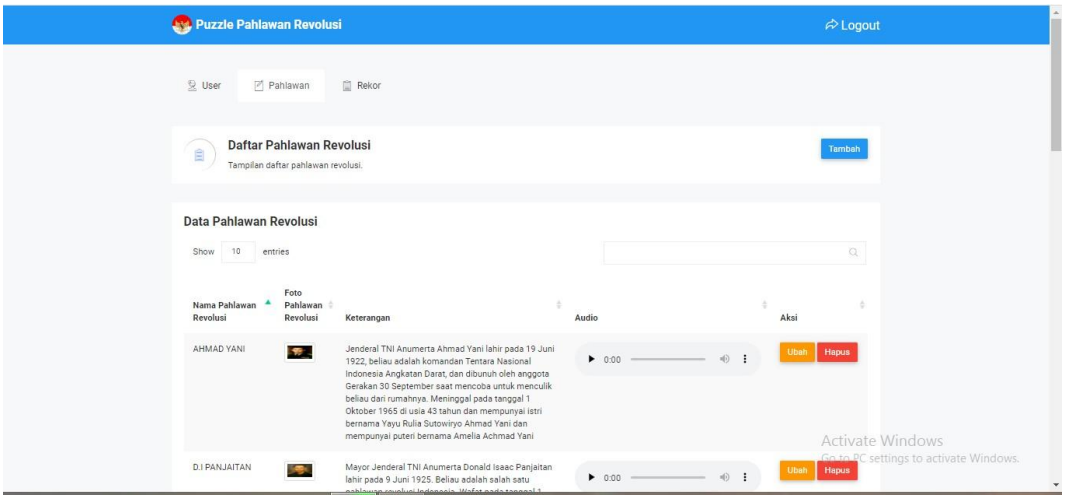
Tabel 4. 9 BlackBox Login Admin

Test Faktor	Hasil	Kesimpulan
Jika memasukkan *Username* dan *Password* dengan *Level Admin*.	✓	Informasi, tampil halaman utama *Admin*.
Screen Shot		
		

Tabel 4. 10 BlackBox Login Pengguna

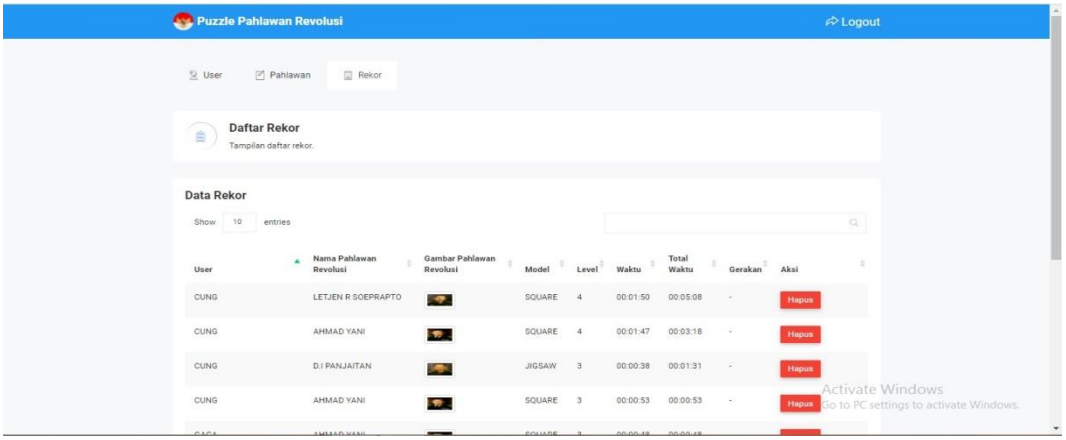
Test Faktor	Hasil	Kesimpulan
Jika memasukkan *Username* pengguna.	✓	Informasi, tampil halaman utama pengguna.
Screen Shot		
		

Tabel 4. 11 BlackBox Tampilan Halaman *Dashboard* Pada Admin

Test Faktor	Hasil	Kesimpulan
Jika *Admin* menekan Halaman *Dashboard*	✓	Informasi, tampil halaman *Dashboard* Pada Admin
Screen Shot		
		

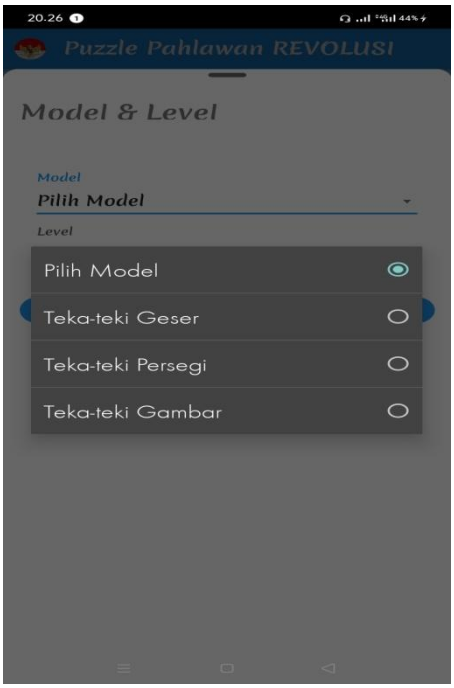
Tabel 4. 12. BlackBox Halaman Data Rekor Pada Admin

Test Faktor	Hasil	Kesimpulan
Jika *Admin* menekan tombol Halaman Data Rekor Pada Admin	✓	Informasi, tampil *Popup* Halaman Data Rekor Pada Admin
Screen Shot		



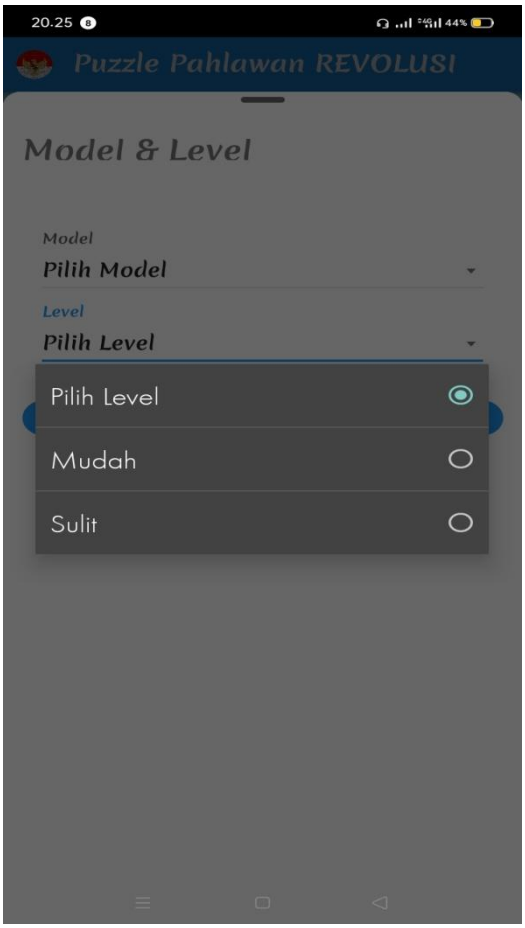
Tabel 4. 13 BlackBox Popup Pilih Model

Test Faktor	Hasil	Kesimpulan
Jika pengguna menekan tombol memainkan pada halaman *Puzzle*.	✓	Informasi, tampil *Popup* pilih *Level*.
Screen Shot		



Tabel 4. 14 BlackBox Popup Pilih Level

Test Faktor	Hasil	Kesimpulan

Jika pengguna menekan tombol memainkan pada halaman *Puzzle*.	✓	Informasi, tampil *Popup* pilih *Level*.
Screen Shot		
		

Tabel 4. 15 BlackBox Halaman Game Puzzle Sesuai Level Yang Telah Di Pilih

Test Faktor	Hasil	Kesimpulan
Jika pengguna telah memilih *Level* dan menekan mainkan pada *Popup* pilih *Level*.	✓	Informasi, tampil halaman *Game Puzzle* sesuai *Level* yang telah di pilih.
Screen Shot		



Tabel 4. 16 BlackBox Hasil Dan Keterangan Tantang Pahlawan.

Test Faktor	Hasil	Kesimpulan
Jika pengguna telah berhasil menyelasikan *Game Puzzle*-nya.	✓	Informasi, tampil hasil dan keterangan tantang alat musik.
Screen Shot		



2. Pengujian *WhiteBox*

Algoritma yang digunakan pada aplikasi Game Edukasi Puzzle ini adalah *Algoritma Fisher-Yates Shuffle. Algoritma Fisher-Yates Shuffle* digunakan untuk pengacakan potongan gambar di tiap-tiap tingkat kesulitan yang berbeda. Pengacakan potongan gambar dalam game dibuat secara teratur agar potongan gambar tidak terjadinya perulangan. Misalkan dalam tingkat kesulitan Mudah ada 9 potongan gambar yang akan diacak dala hal ini matriks 3 x 3, maka array-nya adalahurutan potongan gambar = [0, 1, 2, 3, 4, 5, 6, 7, 8]. Array tersebut dimasukkan ke dalam prosedur shuffle dimana proses pengacakan terjadi. Dari array tersebut didapat panjang array yang kemudian dimasukkan ke dalam variabel (= 9).

Langkah-langkah pengacakan potongan gambar dengan *Fisher-Yates Shuffle* adalah sebagai berikut :

1. Mengambil satu elemen secara acak dari elemen yang tersisa. Pengambilan elemen acak adalah berdasarkan elemen yang tersisa. Misalkan jika = 8, maka elemen acak yang boleh diambil adalah 8 (array[0…7]).
2. Menukar dengan elemen saat ini. Penukaran dilakukan dengan memasukkan elemen saat ini (array[i]) ke dalam suatu variabel baru bernama temp. Kemudian elemen saat ini (array[i]) diisi nilai dari elemen acak tadi (array[ran]). Dan elemen acak (array[ran]) diisi nilai dari variabel temp.
3. Mengulangi selama masih ada elemen yang tersisa. Selanjutnya 9 kali untuk mengacak urutan potongan gambar tersebut. Iterasinya dilihat pada table dibawah.

Tabel 4. 17 Iterasi Pengacakan Fisher Yates Shuffle

I		Array [ ]	t/ array[I]	Array yang Sudah fix	Isi array setelah Di swap
1	2	2	0	1	2,1,0,3,4,5,6,7,8
2	5	5	1	5,2	2,5,0,3,4,1,6,7,8
3	6	6	0	6,5,2	2,5,6,3,4,1,0,7,8
4	4	4	3	4,6,5,2	2,5,6,4,3,1,0,7,8
5	6	0	3	0,4,6,5,2	2,5,6,4,0,1,3,7,8
6	1	5	1	5,0,4,6,5,2	2,1,6,4,3,5,0,7,8
7	1	1	3	1,5,0,4,6,5,1	2,3,6,4,1,5,0,7,8
8	8	8	7	8,1,5,0,4,6,5,2	2,3,6,4,1,5,0,8,7
9	3	4	7	4.8.1.5.0.4.6.5.2	2,3,6,7,1,5,0,8,4

Kolom i menunjukkan index elemen saat ini. Kolom menunjukkan index dari elemen yang akan ditukar, nilai ran tersebut diambil secara acak dari range yang diperbolehkan (i). Kolom array[ ] menunjukkan nilai yang terdapat pada array ke-ran. Kolom t/array[i] menujukkan nilai yang terdapat pada array ke-i (elemen saat ini). Array yang sudah fix menunjukkan array yang dapat ditukar kembali, jumlahnya bertambah seiring bertambahnya iterasi. Kemudian kolom array yang sudah di-swap menunjukkan isi array setelah elemennya ditukar pada setiap iterasi.

Implemtasi Javascript menggunakan algoritma Fisher-Yates Shuffle model Jigzaw:

```
// https://zimjs.com/cat/scrambler.html - less jig and more
saw.
var filename = "";
function acak(rc) {
  let array = [];
  for (let i = 0; i < rc; i++) {
```

```
      array[i] = i;
    }
    let oldElement;
    for (let i = array.length - 1; i > 0; i--) {
      let rand = Math.floor(Math.random() * (i + 1));
      oldElement = array[i];
      array[i] = array[rand];
      array[rand] = oldElement;
    }
    return array;
  }
  document.addEventListener("DOMContentLoaded", function () {
    window.addEventListener("message", function (e) {
      const data = JSON.parse(e.data);
      if (data.status == "load") {
        filename = data.gambar.replace(/^.*[\\\/]/, "");
        model = data.model;
        // alert(data.gambar.replace(filename, ""));
        const frame = new Frame(
          "fit",
          980,
          550,
          "transparent",
          "transparent",
          filename,
          data.gambar.replace(filename, "")
        );
        frame.on("ready", () => {
          // ES6 Arrow Function - similar to function(){}
          zog("ready from ZIM Frame"); // logs in console
  (F12 - choose console)

          // often need below - so consider it part of the
  template
          let stage = frame.stage;
          let stageW = frame.width;
          let stageH = frame.height;

          // REFERENCES for ZIM at http://zimjs.com
          // see http://zimjs.com/learn.html for video and
  code tutorials
          // see http://zimjs.com/docs.html for documentation
          // see https://www.youtube.com/watch?v=pUjHFptXspM
  for INTRO to ZIM
          // see https://www.youtube.com/watch?v=v7OT0YrDWiY
  for INTRO to CODE

          // CODE HERE
          const mob = mobile();

          // ~~~~~~~~~~~~~~~~~~~~
          // CLASS to make JigSaw piece with bumps out or in
  or no bump on any four edges
          class Piece extends Shape {
            // format is 1 for bump out, -1 for bump in and 0
  for no bump
```

```
            constructor(
              w = 100,
              h = 100,
              format = [1, 1, 1, 1],
              s = black,
              ss = 4,
              f = white
            ) {
              super(w, h);
              const p = Piece.part; // static property -
defined below class
              const g = Piece.gap;
              this.s(s).ss(ss).f(f).mt(0, 0);
              if (format[0] == 0) this.lt(w, 0);
              // top left-right
              else {
                if (data.model == "jigsaw") {
                  this.lt(w * p, 0);
                  let s = format[0] == 1 ? -1 : 1; // sign
                  this.ct(w * (p - g / 2), s * w * g, w / 2,
s * w * g); // curve left to middle
                  this.ct(w * (p + g + g / 2), s * w * g, w *
(1 - p), 0); // curve middle to right
                } else if (data.model == "oval") {
                  this.lt(0, 0);
                  let s = format[0] == 1 ? -1 : 1; // sign
                  this.ct(w * (p - g / 2), s * w * g, w / 2,
s * w * g); // curve left to middle
                  this.ct(w * (p + g + g / 2), s * w * g, w,
0); // curve middle to right
                }
                this.lt(w, 0);
              }
              if (format[1] == 0) this.lt(w, h);
              // right top-bottom
              else {
                if (data.model == "jigsaw") {
                  this.lt(w, h * p);
                  let s = format[1] == 1 ? 1 : -1;
                  this.ct(w + s * w * g, h * (p - g / 2), w +
s * w * g, h / 2);
                  this.ct(w + s * w * g, h * (p + g + g / 2),
w, h * (1 - p));
                } else if (data.model == "oval") {
                  this.lt(w, 0);
                  let s = format[1] == 1 ? 1 : -1;
                  this.ct(w + s * w * g, h * (p - g / 2), w +
s * w * g, h / 2);
                  this.ct(w + s * w * g, h * (p + g + g / 2),
w, h);
                }
                this.lt(w, h);
              }
              if (format[2] == 0) this.lt(0, h);
              // bottom right-left
              else {
```

```
                if (data.model == "jigsaw") {
                  this.lt(w * (1 - p), h);
                  let s = format[2] == 1 ? 1 : -1;
                  this.ct(
                    w * (p + g + g / 2),
                    h + s * w * g,
                    w / 2,
                    h + s * w * g
                  );
                  this.ct(w * (p - g / 2), h + s * w * g, w *
p, h + 0);
                } else if (data.model == "oval") {
                  this.lt(w, h);
                  let s = format[2] == 1 ? 1 : -1;
                  this.ct(
                    w * (p + g + g / 2),
                    h + s * w * g,
                    w / 2,
                    h + s * w * g
                  );
                  this.ct(w * (p - g / 2), h + s * w * g, 0,
h + 0);
                }
                this.lt(0, h);
              }
              if (format[3] == 0) this.lt(0, 0);
              // left bottom-top
              else {
                if (data.model == "jigsaw") {
                  this.lt(0, h * (1 - p));
                  let s = format[3] == 1 ? -1 : 1;
                  this.ct(s * w * g, h * (p + g + g / 2), s *
w * g, h / 2);
                  this.ct(s * w * g, h * (p - g / 2), 0, h *
p);
                } else if (data.model == "oval") {
                  this.lt(0, h);
                  let s = format[3] == 1 ? -1 : 1;
                  this.ct(s * w * g, h * (p + g + g / 2), s *
w * g, h / 2);
                  this.ct(s * w * g, h * (p - g / 2), 0, 0);
                }
                this.lt(0, 0);
              }
              this.cp(); // close path
            }
          }
Piece.part = 0.4; // part of the edge with no gap ratio
          Piece.gap = 1 - Piece.part * 2; // gap ratio of
edge

          // ~~~~~~~~~~~~~~~~~~~~
          // PUZZLE SIZE

          console.log(data);
          let numX = parseInt(data.level);
```

```
        let numY = parseInt(data.level);
        const obj = getQueryString(); // note,
getQueryString returns {} now if no query string
        if (obj.col) numX = Math.min(14, Number(obj.col));
// or we would have to start chaching things...
        if (obj.row) numY = Math.min(10, Number(obj.row));

        // ~~~~~~~~~~~~~~~~~~2~~~
        // PICTURE

        const pic =
asset(filename).clone().center().alp(0.7).vis(false); //
checkbox later to turn on
        const w = pic.width / numX;
        const h = pic.height / numY;

        // chop the picture into bitmaps
        // false is for an array rather than the default
Tile output
        // extra is because bumps go outside the width and
height
        const extra = Math.max(w, h) * Piece.gap;
        const pics = chop(asset(filename), numX, numY,
false, extra);

        // ~~~~~~~~~~~~~~~~~~~~
        // PIECES

        // makePieces gets called from Tile - for each
piece
        let count = 0;
        let lastX = rand() > 0.5 ? 1 : -1; // 1 or -1 for
out or in horizontally
        let lastYs = []; // 1 or -1 vertically - remember
with array and modulus
        loop(numX, (i) => {
          lastYs.push(rand() > 0.5 ? 1 : -1);
        });
        function makePiece() {
          // prepare format for jigsaw piece [1,0,-1,0]
          // 1 bump out, 0 no bump, -1 bump in, etc.
          let currentX = lastX * -1; // opposite of last x
          let currentY = lastYs[count % numX] * -1; //
opposite of last y
          let nextX = rand() > 0.5 ? 1 : -1; // randomize
the next 1 or -1 for out or in horizontally
          let nextY = rand() > 0.5 ? 1 : -1; // and
vertically
          // top, right, bottom, left
          let format = [currentY, nextX, nextY, currentX];
          lastX = nextX;
          lastYs[count % numX] = nextY;

          // override edges to 0
          if (count < numX) format[0] = 0;
          else if (count >= numX * numY - numX) format[2] =
```

```
0;
          if (count % numX == 0) format[3] = 0;
          else if ((count - numX + 1) % numX == 0)
format[1] = 0;

          // make a container to hold jigsaw shape and
later picture part

          let piece = new Container(w, h).centerReg({ add:
false });
          piece.puzzle = new Piece(w, h,
format).addTo(piece);
          piece.mouseChildren = false;
          console.log("count");
          console.log(count);
          console.log("count");
          count++;
          return piece;
        }

        const pieces = new Tile({
          obj: makePiece,
          cols: numX,
          rows: numY,
          clone: false, // otherwise makes clone of piece
        })
          .center()
          .drag(stage)
          .animate({
            props: { alpha: 1 },
            time: 0.1,
            sequence: 0.05,
          });

        // ~~~~~~~~~~~~~~~~~~~~
        // HINT AND SNAP HIT BOX

        // tidy up alpha setting on hint around border
        // const outline = new Rectangle(pic.width,
pic.height, clear, mist, 4)
        //   .center()
        //   .ord(-1); // under pieces
        const hint = pieces
          .clone(true) // exact
          .center()
          .ord(-1) // under pieces
          .cache(-5, -5, pic.width + 10, pic.height + 10)
// cache by default does not include outside border
          .alp(0.2)
          .vis(0); // checkbox below to show

        // make a little box to do hit test to see if in
right place
        const snap = 50; // pixel distance considered
correct
        loop(hint, (h) => {
```

```
          h.box = new Rectangle(snap,
snap).centerReg(h).vis(0); // do not use alpha=0 as that
will make it not hittable
        });

        // ~~~~~~~~~~~~~~~~~~~~
        // ADD PICTURE TO PIECES, ADD EVENTS, ROTATE AND
SCRAMBLE
        true;
        const padding = 50;
        const rotate = false;

        const emitter = new Emitter({
          obj: new Poly({ min: 40, max: 70 }, [5, 6], 0.5,
[
            orange,
            blue,
            green,
          ]),
          num: 0,
          force: 6,
          startPaused: true,
        });

        const arrpic = acak(pieces.items.length);
        console.log(arrpic);
        // loop(pieces, (piece, i) => {
        //   console.log(i);
        //   console.log(piece);
        //   piece.alp(0); // sequence animation above will
animate in alpha
        //   pics[i].addTo(piece).setMask(piece.puzzle);
        //   // test on mobile and see if you need to
cache...
        //   // usually this is just cache() but the bumps
are outside the piece
        //   // and the cache size really does not make a
difference if rest is background transparent
        //   if (mob)
        //     piece.cache(-100, -100, piece.width + 200,
piece.width + 200);
        //   if (rotate) {
        //     piece.rotation = shuffle([0, 90, 180,
270])[0];
        //     piece.tap({
        //       time: 0.5, // within .5 seconds
        //       call: () => {
        //         pieces.noMouse(); // do not let anything
happen while animating until done
        //         piece.animate({
        //           props: { rotation:
String(frame.shiftKey ? -90 : 90) }, // string makes
relative
        //           time: 0.2,
        //           call: () => {
        //             pieces.mouse();
```

```
          //             test(piece);
          //           },
          //         });

          //         stage.update();
          //       },
          //       call2: () => {
          //         // if no tap
          //         test(piece);
          //       },
          //     });
          //   } else {
          //     piece.on("pressup", () => {
          //       test(piece);
          //     });
          //   }
          //   piece.on("pressdown", () => {
          //     // shadows are expensive on mobile
          //     // could add it to container so shadow
inside container
          //     // then cache the container but might not be
worth it
          //     if (!mob) piece.sha("rgba(0,0,0,.4)", 5, 5,
5);
          //   });

          //   // scramble location
          //   console.log(piece.x);
          //   // piece.loc(
          //   //   piece.x,
          //   //   piece.y
          //   // );
          //   // piece.loc(
          //   //   padding + w / 2 + rand(stageW - w -
padding * 2) - pieces.x,
          //   //   padding + h / 2 + rand(stageH - h -
padding * 2) - pieces.y
          //   // );
          // });

          // ~~~~~~~~~~~~~~~~~~~~
          // EMITTER

          // ~~~~~~~~~~~~~~~~~~~~
          // MESSAGE LABEL

          const num = numX * numY;
          let placed = 0;
          STYLE = { color: blue.darken(0.3), size: 24 };
          const stats = new Label({
            text: ``,
            // text: `Tersusun ${placed} bagian dari ${num}
bagian puzzle`,
            italic: true,
            align: CENTER,
          })
```

```
            .centerReg()
            .pos(0, 30, CENTER, BOTTOM);

          // ~~~~~~~~~~~~~~~~~~~~
          // TEST FOR PIECE IN RIGHT PLACE AND END

          function test(piece) {
            piece.sha(-1);
            let box = hint.items[piece.tileNum].box;
            if (piece.rotation % 360 == 0 &&
box.hitTestReg(piece)) {
              piece.loc(box).bot().noMouse();
              emitter.loc(box).spurt(30);
              placed++;
              if (placed == num) {
                // stats.text = `Congratulations all ${num}
pieces placed!`;
                stats.text = ``;
                timeout(1, function () {
                  // emitter.emitterForce = 8;
                  // emitter.center().mov(0, -
170).spurt(100);
                });
                window.parent.postMessage(
                  JSON.stringify({
                    status: "selesai",
                    model: model,
                    date: Date.now(),
                  }),
                  "*"
                );
                // timeout(1, function () {
                hintCheck.removeFrom();
                picCheck.removeFrom();
                picCheck.checked = true;
                pieces.animate({ alpha: 0 }, 0.7);
                // outline.animate({ alpha: 0 }, 0.7);
                hint.animate({ alpha: 0 }, 0.7);
                pic.alp(0).animate({ alpha: 1 }, 0.7);
                // new Button({
                //   label: "AGAIN",
                //   color: white,
                //   corner: [60, 0, 60, 0],
                //   backgroundColor: blue.darken(0.3),
                //   rollBackgroundColor: blue,
                // })
                //   .sca(0.5)
                //   .pos(150, 30, LEFT, BOTTOM)
                //   .alp(0)
                //   .animate({ alpha: 1 })
                //   // normally just zgo("index.html") to
reload
                //   // but it is different in CodePen and
they have disabled document.location.reload()
                //   // so not sure what to do... here is a
the puzzle on ZIM
```

```
                //   .tap(() => {
                //     zgo("https://zimjs.com/puzzle",
"_blank");
                //   });
                // });
              } else stats.text = ``;
              // stats.text = `Placed ${placed} piece${
              //   placed == 1 ? "" : "s"
              // } of ${num}`;
            } else stage.update();
          }

          // ~~~~~~~~~~~~~~~~~~~~
          // CHECKBOXES AND FINISHING TOUCHES

          Style.addType("CheckBox", { borderColor:
blue.darken(0.3) });
          const hintCheck = new CheckBox(30, "hint")
            .alp(0)
            .pos(50, 23, LEFT, BOTTOM)
            .wire({ target: hint, prop: "visible", input:
"checked" });
          const picCheck = new CheckBox(30, "picture")
            .alp(0)
            .pos(150, 23, LEFT, BOTTOM)
            .wire({ target: pic, prop: "visible", input:
"checked" });

          // new Label("Tap atau Shift Tap untuk
Rotasi").pos(0, 30, CENTER);
          // const pixar = new Label("Image from Brave ©
Pixar")
          //   .sca(0.5)
          //   .pos(77, 32, RIGHT, BOTTOM)
          //   .hov(purple)
          //   .tap(() => {
          //     pixar.color = blue.darken(0.3);
          //     zgo("https://pixar.com", "_blank");
          //   });
          // frame.madeWith().sca(0.8).pos(20, 12, RIGHT);

          pieces.top(); // add pieces above everything

          stage.update(); // needed to view changes

          window.addEventListener("message", function (e1) {
            if (JSON.parse(e1.data).status == "preview") {
              // console.log(picCheck.checked);
              picCheck.checked = JSON.parse(e1.data).nilai;
            }
          });

          var arrpos = [];
          for (let i = 0; i < pieces.items.length; i++) {
            let jx = arrpic[i];
            let piece = pieces.items[jx];
```

```
            arrpos[i] = {};
            arrpos[i]["x"] = piece.x;
            arrpos[i]["y"] = piece.y;
          }
          for (let i = 0; i < pieces.items.length; i++) {
            let jx = arrpic[i];
            let piece = pieces.items[jx];
            console.log(jx);
            console.log(piece);
            piece.alp(0); // sequence animation above will
animate in alpha
            pics[jx].addTo(piece).setMask(piece.puzzle);
            // test on mobile and see if you need to cache...
            // usually this is just cache() but the bumps are
outside the piece
            // and the cache size really does not make a
difference if rest is background transparent
            if (mob)
              piece.cache(-100, -100, piece.width + 200,
piece.width + 200);
            if (rotate) {
              piece.rotation = shuffle([0, 90, 180, 270])[0];
              piece.tap({
                time: 0.5, // within .5 seconds
                call: () => {
                  pieces.noMouse(); // do not let anything
happen while animating until done
                  piece.animate({
                    props: { rotation: String(frame.shiftKey
? -90 : 90) }, // string makes relative
                    time: 0.2,
                    call: () => {
                      pieces.mouse();
                      test(piece);
                    },
                  });

                  stage.update();
                },
                call2: () => {
                  // if no tap
                  test(piece);
                },
              });
            } else {
              piece.on("pressup", () => {
                test(piece);
              });
            }
            piece.on("pressdown", () => {
              // shadows are expensive on mobile
              // could add it to container so shadow inside
container
              // then cache the container but might not be
worth it
              if (!mob) piece.sha("rgba(0,0,0,.4)", 5, 5, 5);
```

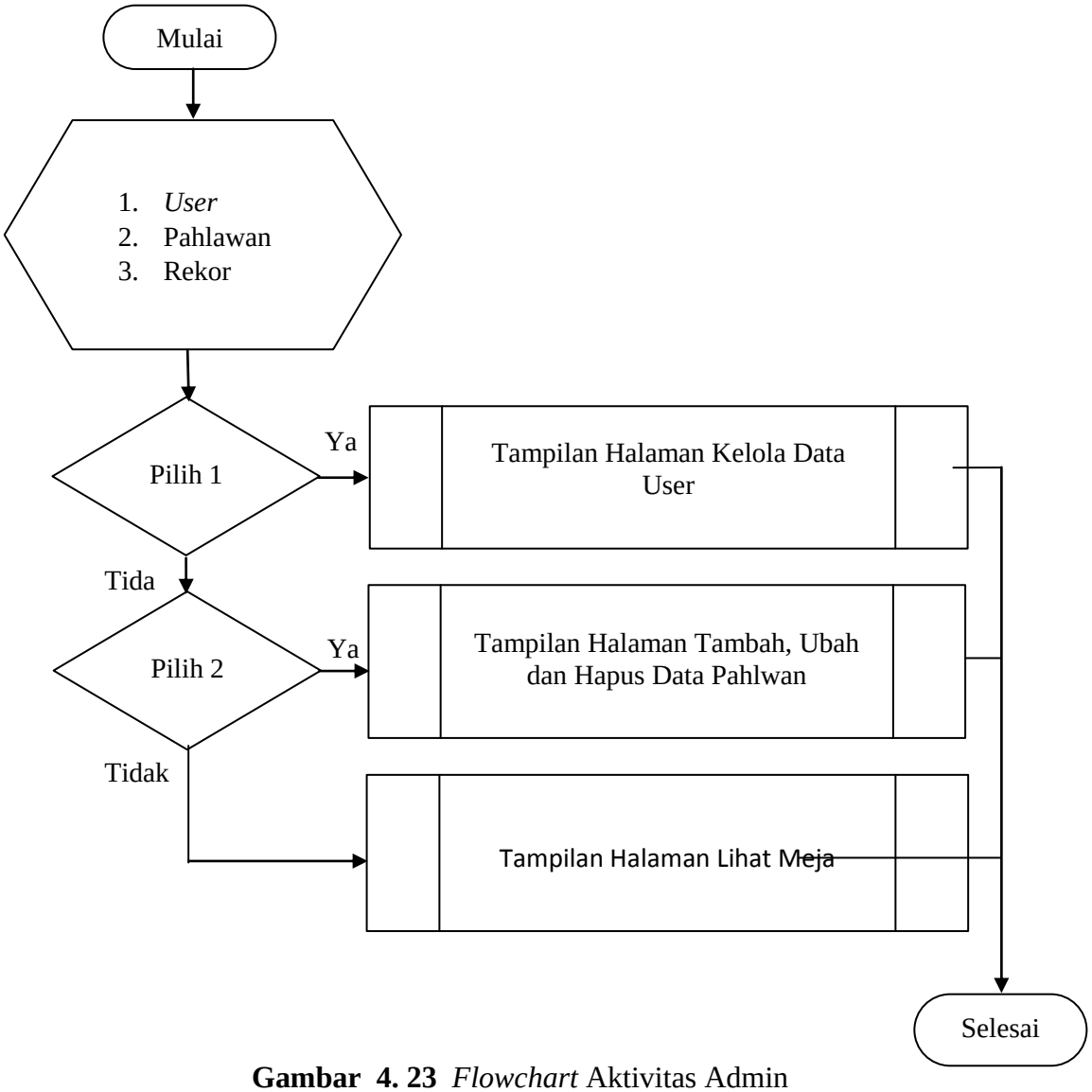
```
            });

            // scramble location
            console.log(arrpos[jx]["x"]);
            piece.loc(arrpos[jx]["x"], arrpos[jx]["y"]);
            // piece.loc(
            //   padding + w / 2 + rand(stageW - w - padding
* 2) - pieces.x,
            //   padding + h / 2 + rand(stageH - h - padding
* 2) - pieces.y
            // );
            test(piece);
          }

        }); // end of ready
      }
    });
});
```

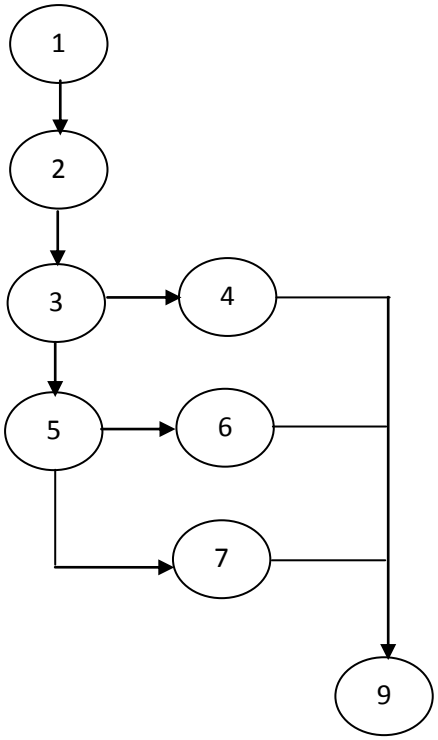
Pengujian aplikasi dilakukan dengan cara pengujian *WhiteBox:*

a. *Flowchart* dan *Flowgraph* Aktivitas *Admin*



Gambar 4. 23 *Flowchart* Aktivitas Admin

Dari *Flowchart* yang digunakan untuk pengujian perangkat lunak, maka ditentukan *Flowgraph* sebagai berikut:



Gambar 4. 24 *Flowgraph* Aktivitas *Admin*

Dari *Flowgraph* aktivitas *Admin* diatas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic Complexcity* V(G) dari *Egde* dan *Node*:

   Dengan rumus : V(G) = E – N + 2

   E (*edge*) = 9

   N (*Node*) = 8

   P (Predikat *Node*) = 2

   Penyelesaian : V(G) = E – N + 2

   = 9 – 8 + 2

   = 3

Predikat (P) = P + 1

= 2 + 1

= 3

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 3
3. *Independent path* pada *Flowgraph* diatas adalah:

   Path 1 = 1 – 2 – 3 – 4 – 8

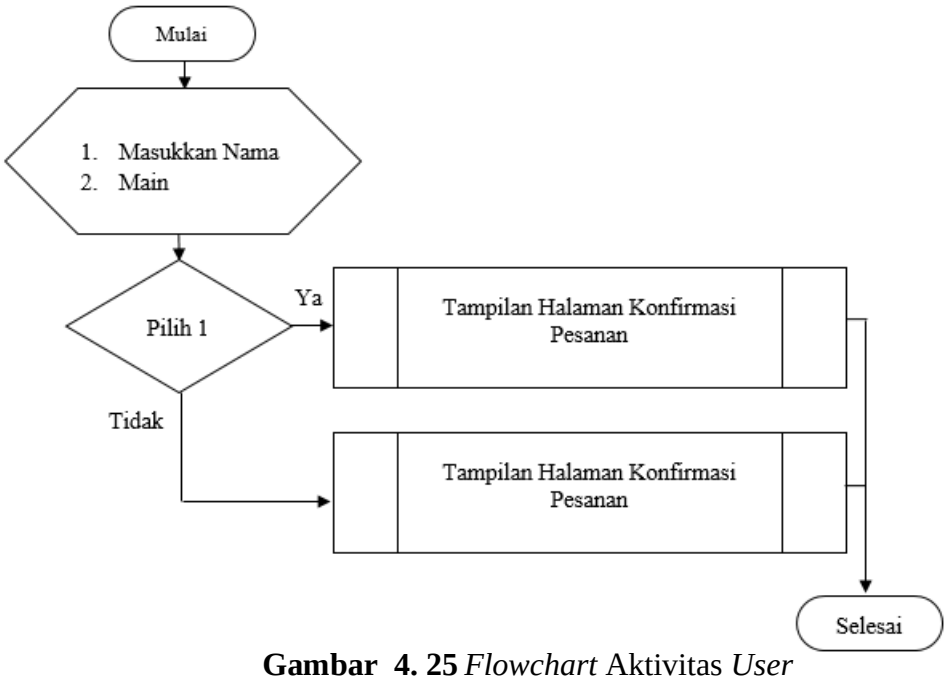
   Path 2 = 1 – 2 – 3 – 5 – 6 – 8

   Path 3 = 1 – 2 – 3 – 5 – 7 – 8
4. Grafik Matriks Aktivitas *Admin*

Tabel 4. 18 Grafik Matriks Aktivitas Admin

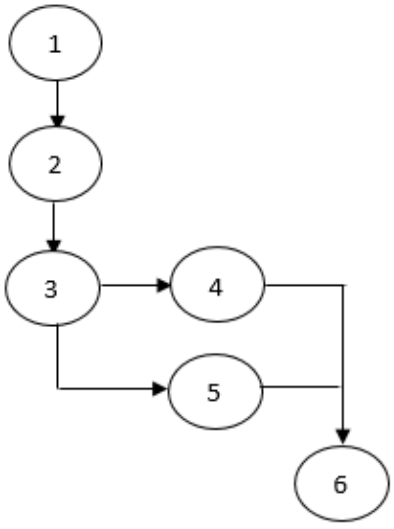
	1	2	3	4	5	6	7	8	E – 1
1		1							1 – 1 = 0
2			1						1 – 1 = 0
3				1	1				2 – 1 = 1
4								1	1 – 1 = 0
5						1	1		2 – 1 = 1
6								1	1 – 1 = 0
7								1	1 – 1 = 0
8									0
SUM (E + 1)									2 + 1 = 3

b. *Flowchart* dan *Flowgraph* Aktivitas *User*



Gambar 4. 25 *Flowchart* Aktivitas *User*

Dari *Flowchart* yang digunakan untuk pengujian perangkat lunak, maka ditentukan *Flowgraph* sebagai berikut:



Gambar 4. 26 *Flowgraph* Aktivitas *User*

Dari *Flowgraph* aktivitas bar diatas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic Complexcity* V(G) dari *Egde* dan *Node*:

   Dengan rumus : V(G) = E – N + 2

   E (*edge*) = 6

   N (*Node*) = 6

   P (Predikat *Node*) = 1

   Penyelesaian : V(G) = E – N + 2

   = 6 – 6 + 2

   = 2

   Predikat (P) = P + 1

   = 1 + 1

   = 2

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2
3. *Independent path* pada *Flowgraph* diatas adalah:

   Path 1 = 1 – 2 – 3 – 4 – 6

   Path 2 = 1 – 2 – 3 – 5 - 6

4. Grafik Matriks Aktivitas Bar

Tabel 4. 19 Grafik Matriks Aktivitas Bar

	1	2	3	4	5	E – 1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3				1	1	2 – 1 = 1
4					1	1 – 1 = 0
5					1	1 - 1 = 0
6						0
SUM (E + 1)						1 + 1 = 2

c. Hasil Pengujian

Tabel 4. 20 Hasil Pengujian

No	*Flowchart*	*Independent Path*	*Region*	*Cyclomatic Complexity*
1	Aktivitas *Admin*	3	3	3
2	Aktivitas *User*	2	2	2

F. Pengujian Penerapan Metode SUS

Metode penelitian ini menggunakan SUS (*System Usability Scale*), pengambilan data menggunakan kuisioner yang terdiri dari 11 responden kelas 1,2,3 dan 4) dan memiliki 10 pertanyaan dengan 5 pilihan jawaban dari 1 sampai 5 dengan pertanyaan sebagai berikut:

Q1 Apakah kamu senang belajar menggunakan aplikasi ini?

Q2 Apakah kamu merasa aplikasi ini rumit untuk digunakan?

Q3 Apakah suara di dalam aplikasi ini terdengar dengan jelas?

Q4 Apakah kamu membutuhkan bantuan dari orang lain dalam menyelesaikan permainan ini?

Q5 Menurut adik-adik apakah gambarnya sudah bagus/jelas?

Q6 Apakah kamu merasa ada banyak hal yang tidak konsisten/tidak serasi pada aplikasi ini?

Q7 Apakah huruf/tulisan pada aplikasi terlihat dengan jelas?

Q8 Apakah kamu merasa aplikasi itu membingungkan?

Q9 Apakah aplikasi ini membantu kamu untuk leboh mudah belajar mengenal pahlawan revolusi di Indonesia?

Q10 Apakah aplikasi ini tidak menarik?

Tabel 4. 21 Skala Penilaian Skor

Jawaban	Skor
Sangat Tidak Setuju	1
Tidak Setuju	2
Ragu-ragu	3
Setuju	4
Sangat Setuju	5

Data-data kuisioner tersebut diberikan kepada responden, selanjutnya melakukan konvensi tanggapan responden dengan cara sebagai berikut (Saputra, 2019):

a) Pertanyaan ganjil, yaitu 1,3,5,7, dan 9 skor yang diberikan oleh responden dikurangi 1

b) Pertanyaan genap, yaitu 2,4,6,8, dan 10 skor yang diberikan oleh responden digunakan untuk mengurangi 5

c) Hasil dari konversi tersebut selanjutnya dijumlahkan untuk setiap responden kemudian dikalikan dengan 2,5 agar mendapatkan rentang antara 0-100

d) Setelah skor dari masing-masing responden yang telah diketahui langkah selanjutnya adalah mencari skor rata-rata dengan cara menjumlahkan semua hasil skor dan dibagi dengan jumlah responden yang ada.

Berdasarkan penelitian yang dilakukan diperoleh data tabel sebagai berikut:

Tabel 4. 22 Skor konsioner sebelum di kelola

Nama	Skor Asli									
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Responden 1	5	1	4	2	5	2	4	2	5	1
Responden 2	5	1	4	2	5	2	5	2	5	1
Responden 3	5	1	4	2	5	2	5	2	5	1
Responden 4	5	1	4	2	5	2	5	2	5	1
Responden 5	5	1	4	2	5	2	5	2	5	1
Responden 6	5	1	4	2	5	2	5	2	5	1
Responden 7	5	2	4	2	5	2	5	2	5	1
Responden 8	5	2	3	4	5	2	5	2	5	1
Responden 9	5	2	5	2	5	1	5	2	5	1
Responden 10	5	2	5	2	5	1	5	2	5	1
Responden 11	5	2	5	2	5	1	5	2	5	1

(Sumber: Dikelola oleh peneliti)

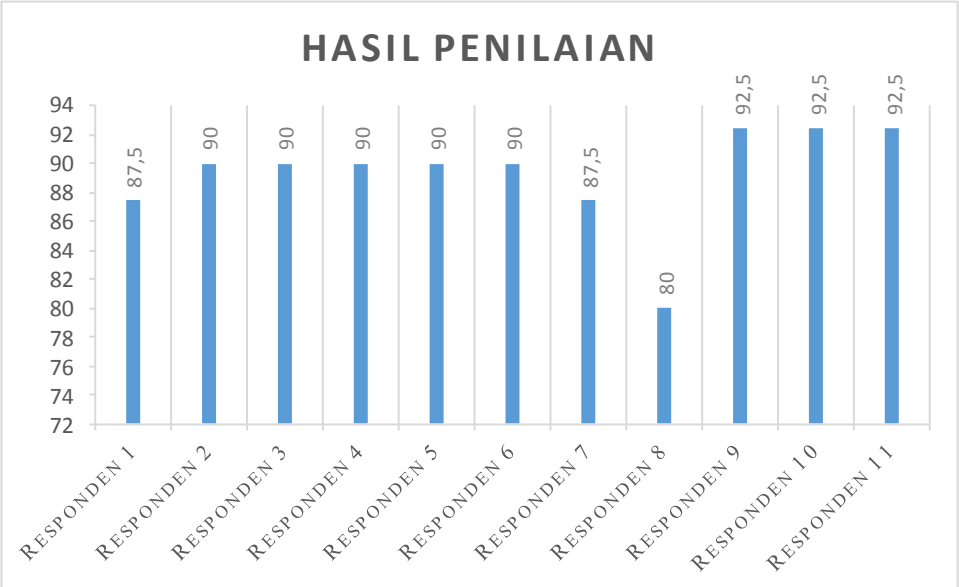
Setelah mendapatkan skor disetiap pertanyaan berdasarkan peraturan metode SUS maka diperoleh data sebagai berikut:

Tabel 4. 23 Skor konsioner sesudah di kelola

No	Nama	Skor Hasil Hitung										Jumlah	Nilai (Jumlah x 2,5)
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10		
1	Responden 1	4	4	3	3	4	3	3	3	4	4	35	87,5
2	Responden 2	4	4	3	3	4	3	4	3	4	4	36	90
3	Responden 3	4	4	3	3	4	3	4	3	4	4	36	90
4	Responden 4	4	4	3	3	4	3	4	3	4	4	36	90
5	Responden 5	4	4	3	3	4	3	4	3	4	4	36	90
6	Responden 6	4	4	3	3	4	3	4	3	4	4	36	90
7	Responden 7	4	3	3	3	4	3	4	3	4	4	35	87,5
8	Responden 8	4	3	2	1	4	3	4	3	4	4	32	80
9	Responden 9	4	3	4	3	4	4	4	3	4	4	37	92,5
10	Responden 10	4	3	4	3	4	4	4	3	4	4	37	92,5
11	Responden 11	4	3	4	3	4	4	4	3	4	4	37	92,5
Skor Rata-rata (Hasil Akhir)													**89.31**

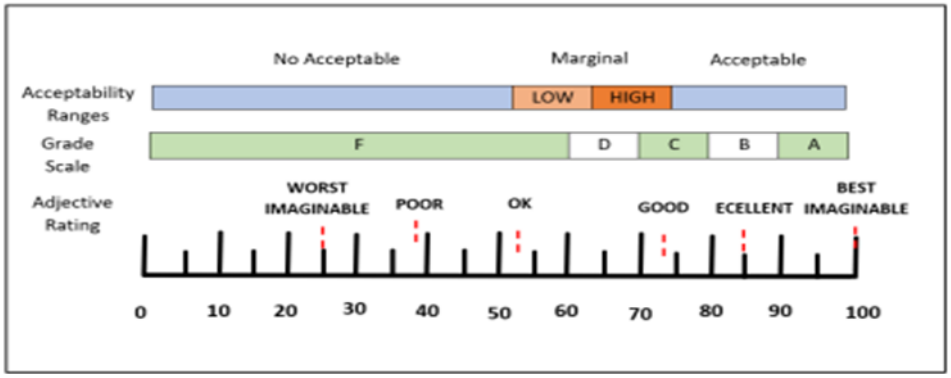
(Sumber: Dikelola oleh peneliti)

Berdasarkan pengolahan data yang dilakukan diperoleh skor rata-rata hasil akhir SUS adalah 89,31. Adapun grafik hasil penilaiannya dapat dilihat sebagai berikut:



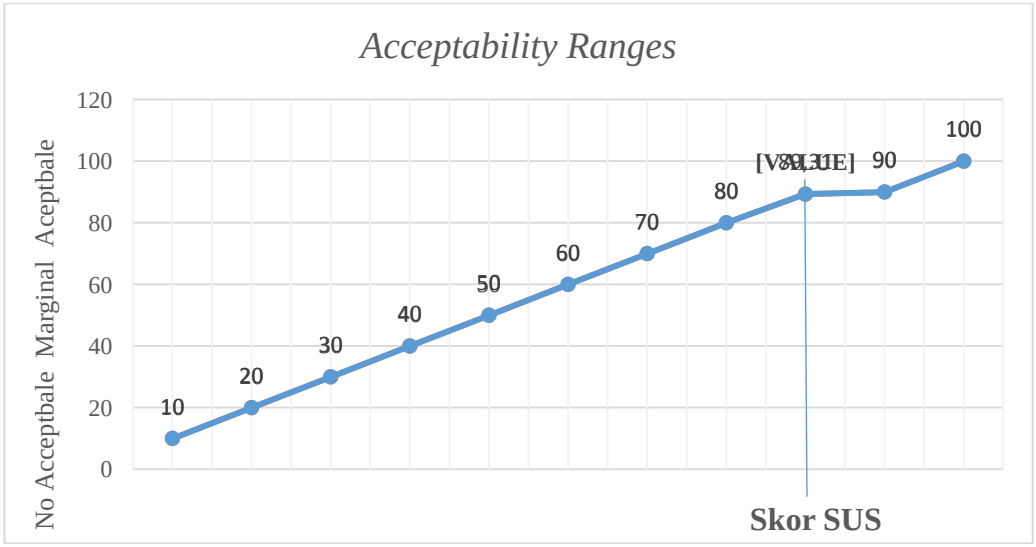
Gambar 4. 27 Hasil Penilaian Kuisioner

Skor SUS yang telah diperoleh kemudian dibandingkan dengan skala dibawah ini. Skor SUS 89,31 jika diinterpretasikan sesuai skala dibawah menunjukkan bahwa Aplikasi *Game Puzzle* Pengenalan Tokoh Pahlawan Revolusi Indonesia berada pada kategori *acceptable*, *grade* B dan mendapatkan predikat *excellent* pada kategori *adjective rating*.



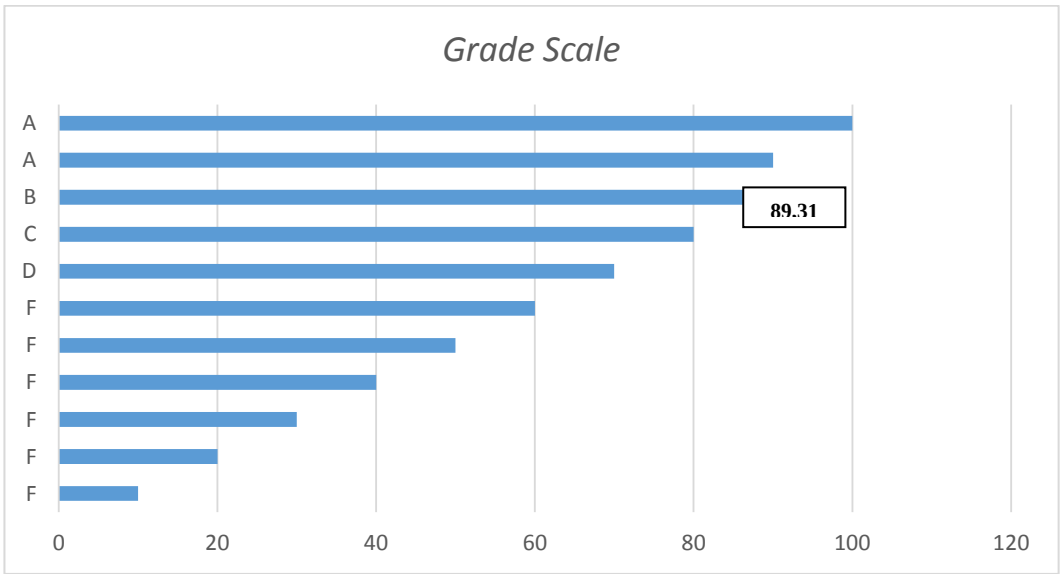
Gambar 4. 28 Penentuan Hasil Penelitian Skor SUS

Adapun grafik *Acceptability Ranges* dari data yang telah diolah adalah sebagai berikut: Skor SUS **89.31** memenuhi kriteria ***Acceptable***



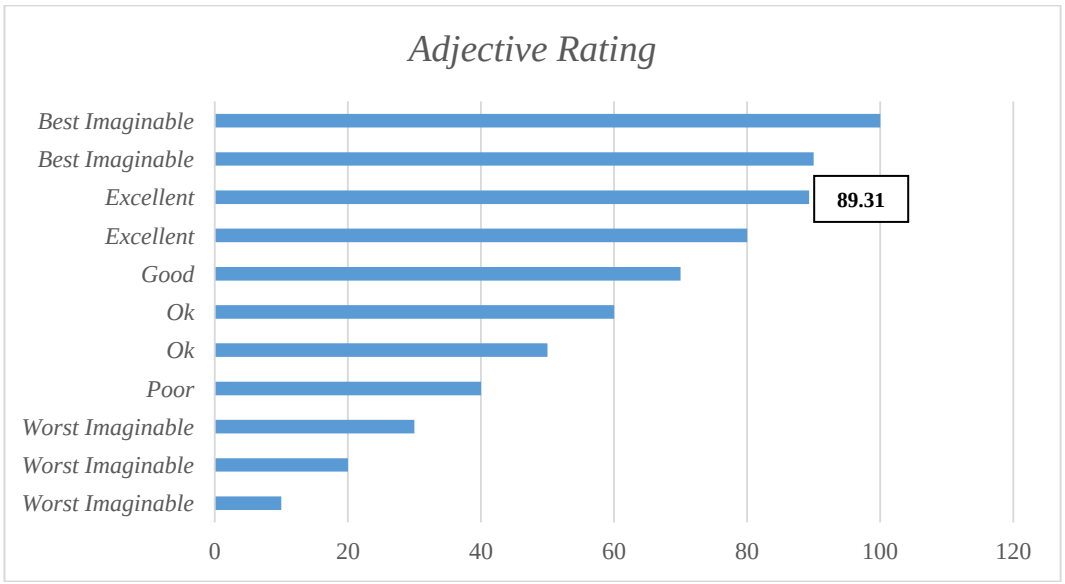
Gambar 4. 29 Grafik *Acceptability Ranges*

Adapun grafik *Grade Scale* dari data yang telah diolah adalah sebagai berikut: Skor SUS **89.31** memenuhi kriteria skala **B**



Gambar 4. 30 Grafik *Grade Scale*

Sedangkan, grafik *Adjective Rating* dari data yang telah diolah adalah sebagai berikut: Skor SUS **89.31** memenuhi kriteria ***Excellent***



Gambar 4. 31 Grafik *Adjective Rating*

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan makan dapat disimpulkan bahwa :

1. Perancangan media pembelajaran *game puzzle* pengenalan tokoh pahlawan revolusi Indonesia bagi anak usia dini dikategorikan sangat baik karena mendapatkan skor 89,31.
2. Aplikasi yang dibuat sebagai pengenalan tokoh-tokoh pahlawan bagi anak usia dini dikategorikan menarik dan edukatif.

B. Saran

Adapun yang dapat menjadi saran Penulis bagi pengembang selanjutnya, diantaranya sebagai berikut :

1. Pada aplikasi *Game* terdapat beberapa gambar yang kualitasnya masih kurang baik, sehingga pada proses pengembangan berikutnya diharapkan dapat menampilkan gambar dengan kualitas lebih baik.
2. Untuk pengembangan selanjutnya diharapkan *Game* Edukasi *Puzzle* ini dapat dibuat berbasis Android 3D.

DAFTAR PUSTAKA

Arfamaini, R. (2016). Aplikasi Penjualan Outlet Smartphone Berbasis Android Pada Gx Shop Surakarta. Applied Microbiology And Biotechnology, 85(1), 2071–2079.

Asih, V., Saputra, A., & Subagio, R. T. (2020). Penerapan Algoritma *Fisher Yates Shuffle* Untuk Aplikasi Ujian Berbasis Android. Jurnal Digit, 10(1), 59. Https://Doi.Org/10.51920/Jd.V10i1.156

Doni Koesoema, A., & Anggraeny, E. (2020). Inspirasi Praktik Baik Pendidikan Karakter Berbasis Kultur Sekolah: Membentuk Budaya Berkarakter Setiap Hari. Pt Kanisius.

Edora, S. (2019). Rancang Bangun Game Edukasi Puzlle Sebagai Media Pembelajaran Berbasis Android Menggunakan Algoritma Fisher-Yates Shuffle (Studi Kasus Di Sdn 2 Ngumbul Todanan–Blora). Pelita Teknologi, 13(2), 32–38.

Fatimah, D. D. S., Tresnawati, D., & Ma'rup, C. S. (2017). Perancangan Game Puzzle Untuk Pembelajaran Menggunakan Metodologi Multimedia. Jurnal Algoritma, 14(2), 281–287.

Helfira, D. (2019). Aplikasi Rekam Medis Pasien Pada Rsud Ogan Ilir. Politeknik Negeri Sriwijaya.

Kusumawardani, A., & Faturochman, M. A. (2004). Nasionalisme. Buletin Psikologi, 12(2).

Marwan, A., Sutardi, S., & Ramadhan, R. (2017). Penerapan Metode Linear Congruent Method (Lcm) Dalam Perancangan Dan Pembuatan Game Monopoli Edukasi Untuk Tokoh Pahlawan Nasional. Semantik, 3(1).

Muarie, M. Son. (2015). Rancang Bangun Sistem Ujian Online Pada Smp Negeri 8 Sekayu Muhamad Son Muarie Nidn : - Teknik Informatika Politeknik Sekayu Email : Meki.Koe@Gmail.Com Diintegrasikan Dengan Kehadiran Teknologi Informasi . Dampak Dari Perkembangan Itulah Yang Manusia Yang. Ii(1), 28–40.

Ningsih, E. Y., Rosyadi, I., & ... (2020). Sistem Informasi Pengaduan Online Pada Masyarakat Kecamatan Kajen Kabupaten Pekalongan Berbasis Web Dan Android. … : Membangun Informasi …, 1.

Nugroho, A. H., & Komarudin, R. (2018). Pengembangan Game Puzzle Berbasis Android Sebagai Media Edukasi Pengenalan Pahlawan Nasional. Informatics For Educators And Professionals, 2(2), 149–158.

Pasaribu, J. S., Informatika, T., Ganesha, P. P., Peserta, P., Baru, D., Web, B., & Based, W. (2017). Penerapan Framework Yii Pada Pembangunan. Iii(2), 154–163.

Pongoh, F. M., Sinsuw, A. A. E., & Tulenan, V. (2016). Alat Peraga Digital Pengenalan Pahlawan Sulawesi Utara. 8(1).

Rukminingsih, Adnan, G., & Latief, M. A. (2020). Metode Penelitian Pendidikan. Penelitian Kuantitatif, Penelitian Kualitatif, Penelitian Tindakan Kelas. In Journal Of Chemical Information And Modeling (Vol. 53, Issue 9).

Siregar, H. F., Siregar, Y. H., & Melani, M. (2018). (2018). Perancangan Aplikasi Komik Hadist Berbasis Multimedia. Jurti (Jurnal Teknologi Informasi), 2(2), 113-121. Jurti (Jurnal Teknologi Informasi), 2(2), 113–121. Http://Www.Jurnal.Una.Ac.Id/Index.Php/Jurti/Article/View/425

Siregar, N. S. (2019). Perancangan Aplikasi Penentuan Air Mineral Berdasarkan Kandungan Terbaik Menggunakan Perbandingan Metode Weighted Product Dan Weighted Sum Model (Pp. 1–62). Https://Jurnal.Pancabudi.Ac.Id/Index.Php/Fastek/Article/View/1410/1277

Sony Putra, G. S. (2017). Program Studi Sistem Informasi, Manajemen Informatika, Dan Teknik Komputer Fakultas Teknik Universitas Batam. 7.

Syamsiah, S. (2019). Perancangan Flowchart Dan Pseudocode Pembelajaran Mengenal Angka Dengan Animasi Untuk Anak Paud Rambutan. String (Satuan Tulisan Riset Dan Inovasi Teknologi), 4(1), 86. Https://Doi.Org/10.30998/String.V4i1.3623

Wasil, M., Samsu, L., & Putra, Y. K. (2020). Sistem Informasi Geografis Untuk Pemetaan Homestay Di Lombok Timur Berbasis Android. 29(9), 1890–1896.

Wibawanto, W., & Ds, S. S. M. (2017). Desain Dan Pemrograman Multimedia Pembelajaran Interaktif. Cerdas Ulet Kreatif Publisher.

Widjoyo, L. H. (2020). Ta: Perancangan Buku Ilustrasi Pengenalan Pahlawan Nasional Sebagai Upaya Meningkatkan Rasa Nasionalis Pada Anak Usia 9-11 Tahun. Universitas Dinamika.

Yusfrizal. (2020). Penerapan Algoritma Fisher-Yates Shuffle Pada Game Mencocokkan Gambar Monumen Dunia. Jurnal Teknik Informatika Kaputama (Jtik), 4(2), 162–170.