

## **BAB IV**

### **DESKRIPSI HASIL PENELITIAN DAN PEMBAHASAN**

#### **4.1 Deskripsi Hasil Penelitian**

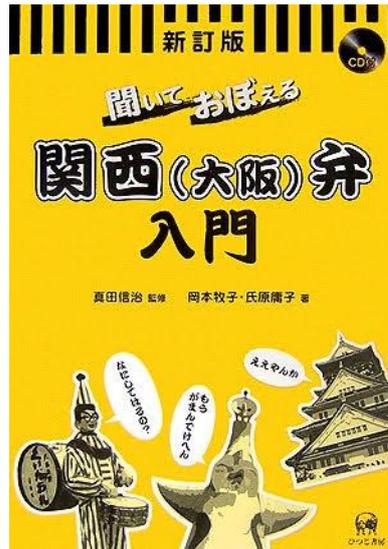
##### **4.1.1 Perancangan dan Proses Pembuatan Aplikasi *Kansaimo***

Pada proses pembuatan aplikasi *Kansaimo*, *software* yang penulis gunakan merupakan *software* desain dari Adobe, yaitu Adobe Animate CC 2015 dan Adobe Photoshop CC 2015. Adobe Animate CC 2015 digunakan dalam pembuatan tampilan, pembuatan berbagai animasi dan menjadikan semua hasil pengolahan menjadi sebuah aplikasi yang dapat dijalankan pada *smartphone* berbasis *android*. Sedangkan untuk Adobe Photoshop CC 2015 digunakan dalam proses pengeditan gambar untuk keperluan bahan yang dibutuhkan saat proses pembuatan aplikasi dengan Adobe Animate CC 2015, gambar yang sudah melewati proses pengeditan maka akan dimasukkan kedalam aplikasi Adobe Animate CC 2015 untuk diolah menjadi bagian-bagian yang terdapat pada aplikasi. Dalam bab ini, menjelaskan tentang bagaimana proses pembuatan aplikasi *Kansaimo*.

##### **4.1.1.1 Perancangan Aplikasi**

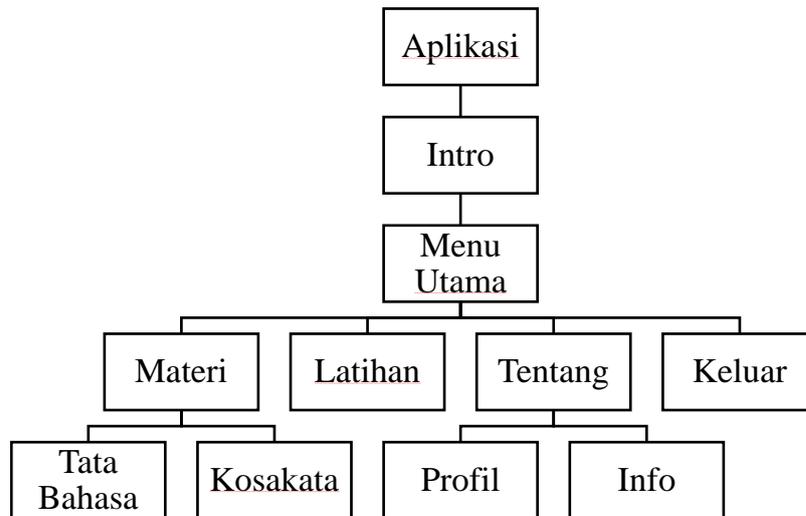
Merancang aplikasi merupakan tahap awal dalam pembuatan aplikasi. Dalam tahap awal ini penulis mengumpulkan semua data yang diperlukan seperti yang sudah ditentukan pada bab sebelumnya. Data yang penulis kumpulkan berupa daftar pola kalimat atau tata bahasa lengkap dengan penjelasannya dalam buku *Kiiteoboeru Kansai(Osaka)ben Nyuumon* sebanyak 21 contoh tata bahasa dialek

Kansai, daftar kosakata yang sering digunakan penutur dialek Kansai berserta arti dan contoh pengucapannya dalam bentuk suara, dan beberapa pertanyaan yang penulis buat untuk menu latihan sebagai evaluasi untuk pengguna aplikasi ini.



**Gambar 4.1**  
*Kiiteoboeru Kansai(Osaka)ben Nyuumon*

Pembuatan rancangan awal ini juga bertujuan untuk mempermudah penulis dalam proses pengembangan aplikasi ke tahap lebih lanjut sehingga kekeliruannya dalam membuat aplikasi dapat berkurang karena rancangan awal yang sudah jelas. Berikut adalah rancangan aplikasi *Kansaimo*.

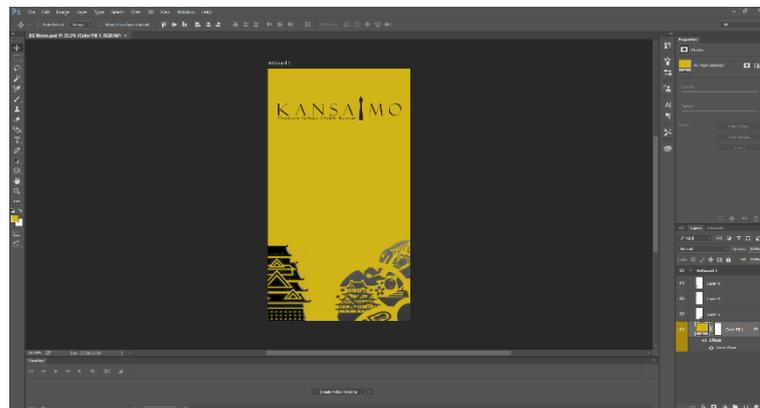


**Bagan 4.1**  
*Blueprint media Kansaimo*

1. *Aplikasi*: merupakan aplikasi Kansaimo itu sendiri.
2. *Intro*: meliputi logo unikom sebagai pembuka dan cover aplikasi beserta *button* mulai untuk memulai aplikasi.
3. *Menu Utama*: menyediakan empat pilihan menu, yaitu *Materi*, *Latihan*, *Tentang*, dan tombol keluar berupa *icon* dua pedang yang berbentuk X.
4. *Materi*: berisikan pilihan materi tata bahasa dan kosakata.
5. *Latihan*: berisikan 20 soal latihan untuk menguji kemampuan.
6. *Tentang*: terdapat dua pilihan di dalamnya, yaitu *profil* yang memuat data diri penulis, dan *info* yang memuat informasi aplikasi Kansaimo.
7. *Keluar*: tombol keluar berupa *icon* atau simbol dua pedang yang berbentuk X yang terdapat di pojok kiri layar *smartphone*.

#### 4.1.1.2 Pembuatan Desain Aplikasi

Setelah konsep rancangan aplikasi dibuat, tahap selanjutnya adalah membuat desain untuk aplikasi yang akan dibuat. Tahap ini sangat penting karena disini penulis mendesain berbagai unsur aplikasi yang akan digunakan mulai dari background hingga button sedemikian rupa sehingga dari segi tampilan terlihat menarik. Untuk mendesain semua bahan yang diperlukan, penulis menggunakan *software* desain dari *Adobe Photoshop CC 2015*.



**Gambar 4.2**  
Desain *Background* Menu Utama

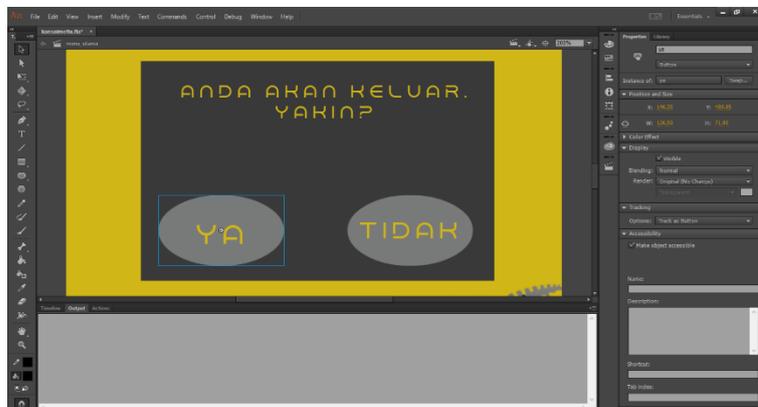


**Gambar 4.3**  
Desain *Button*

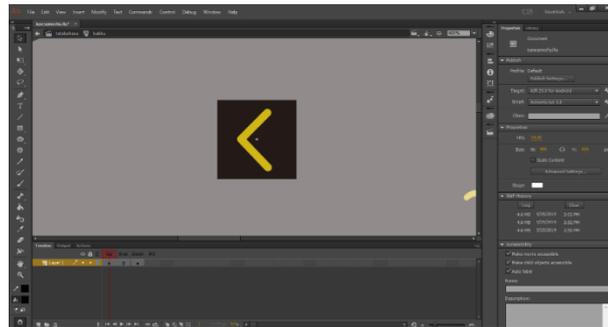


**Gambar 4.4**  
Desain Berbagai *Button* Navigasi

Selain mendesain menggunakan *software* desain *Adobe Photoshop CC 2015* ini, penulis juga mendesain beberapa navigasi untuk aplikasi langsung di dalam *software* pembuat aplikasi *Kansaimo* ini, yaitu *Adobe Animate CC 2015*.



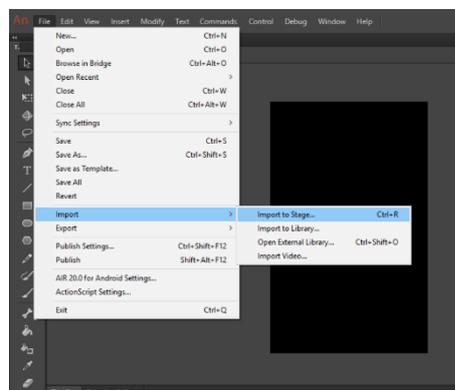
**Gambar 4.5**  
Desain *Button* di *Adobe Animate CC 2015*



**Gambar 4.6**  
 Desain *Button Back* di *Adobe Animate CC 2015*

#### 4.1.1.3 *Import Desain Aplikasi*

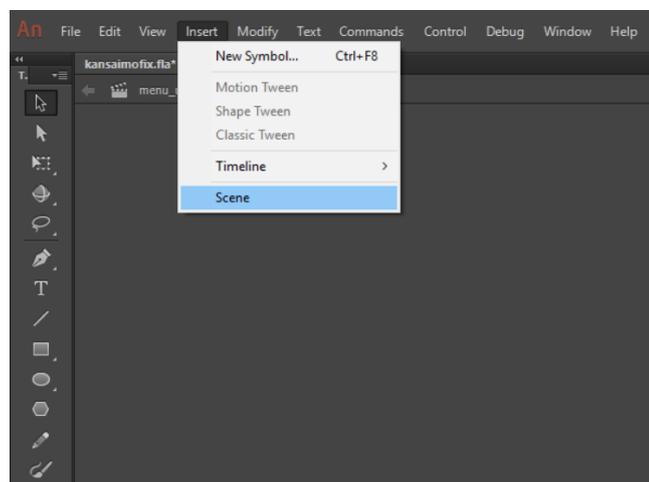
Langkah selanjutnya adalah *import data* atau memasukan semua objek yang sudah didesain menggunakan *Adobe Photoshop CC 2015* ke dalam *Adobe Animate CC 2015* sebagai software pembuat aplikasi *Kansaimo*. Untuk *import* objek yang akan langsung dibuat kedalam aplikasi, dapat memilih pilihan *Import to Stage*, sedangkan objek yang akan disimpan dalam *software* dan akan digunakan sewaktu waktu, dapat memilih *Import to Library* dalam pilihan menu *File* lalu ke *Import pada Adobe Animate CC 2015*.



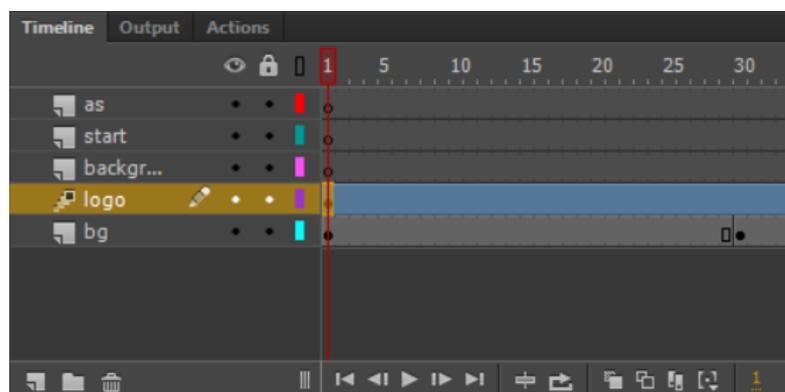
**Gambar 4.7**  
*Import* Objek Desain

#### 4.1.1.4 Pembuatan Animasi

Tahap selanjutnya setelah memasukan objek yang diperlukan ke dalam *library* maupun *stage* adalah membuat *scene* yang terdapat pada menu *Insert* lalu ke *Scene* untuk membuat *scene* sebagai wadah untuk objek yang akan diolah, dilanjutkan membuat layer pada setiap *scene*.

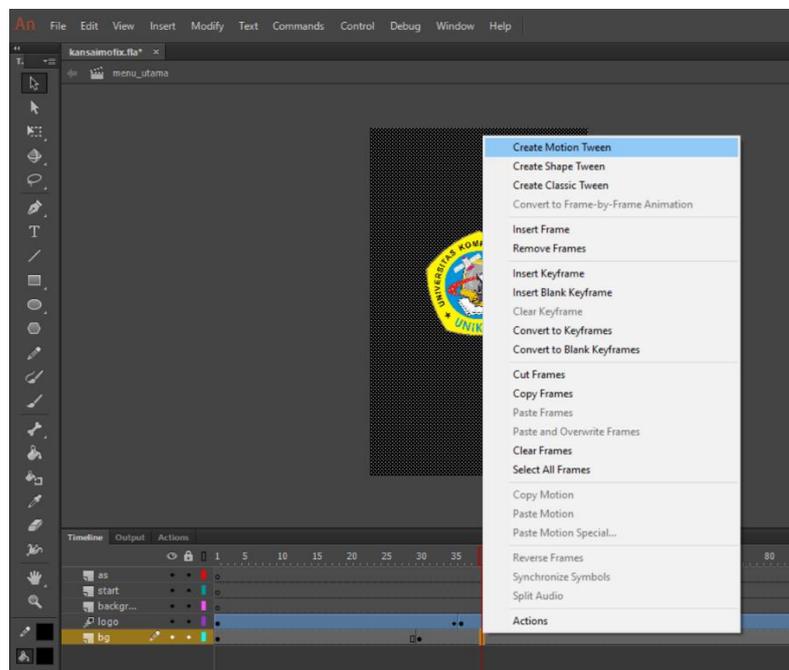


**Gambar 4.8**  
Menambahkan *Scene*



**Gambar 4.9**  
Menambahkan *Layer*

Setelah itu *import* masukan *file* atau objek yang dibutuhkan ke dalam *layer*, lalu langkah selanjutnya adalah membuat animasi pada objek yang sudah penulis *import* untuk aplikasi yang akan dibuat. Contoh dari sekian banyak pilihan animasi yang tersedia di *Adobe Animate CC 2015* ini penulis membuat animasi yang memberikan efek muncul dan menghilang suatu objek. Salah satunya penulis terapkan pada bagian pembuka aplikasi dengan menggunakan *Motion Tween* ini.



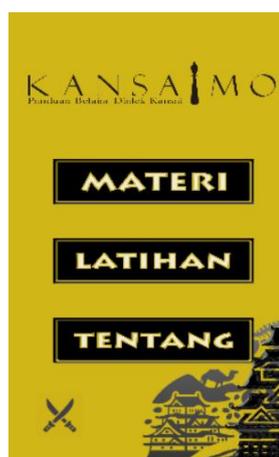
**Gambar 4.10**  
Membuat Animasi *Motion Tween*

Pada tahap berikutnya penulis membuat background dan menyusun beberapa *icon* pendukung semenarik mungkin dengan menggunakan beberapa objek yang telah di *import* serta membuat *button* 'mulai' yang berfungsi untuk memulai pembelajaran atau menuju menu utama.



**Gambar 4.11**  
*Button Mulai pada Intro*

Jika *button* ‘mulai’ disentuh, pengguna akan langsung menuju *scene* menu utama. Dalam *scene* menu utama terdapat 3 pilihan *button* utama, yaitu Materi, Latihan, dan Tentang yang masing-masing diantaranya jika disentuh akan secara otomatis mengarah kepada *scene* yang isinya sesuai dengan nama yang terdapat pada *button* tersebut. Berikut adalah tampilan dari *scene* menu utama pada media *Kansaimo*.



**Gambar 4.12**  
*Menu Utama Kansaimo*

#### 4.1.1.5 Pemberian *ActionScript*

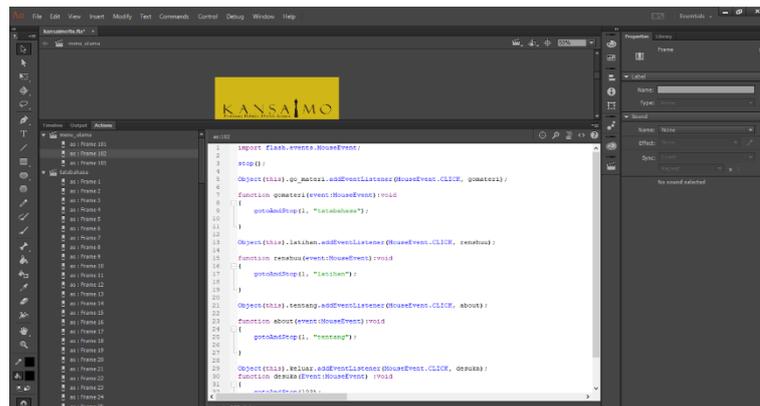
Langkah selanjutnya setelah pembuatan animasi dan penataan objek untuk aplikasi adalah pemberian *actionsript*. Pemberian *action script* ini juga tidak kalah penting, seperti memberikan ruh pada aplikasi yang akan dibuat, tanpa *actionsript* aplikasi tidak akan berfungsi sebagaimana mestinya. *Actionsript* ini berfungsi memberikan perintah atau sebagai jembatan penghubung antara pengguna dan aplikasi. Contohnya jika pengguna memilih menu materi, maka setelah *button* ‘materi’ disentuh, sistem akan mengarahkan pengguna pada tampilan materi. Lebih mudahnya adalah untuk menghubungkan *frame* satu dengan *frame* lainnya atau *scene* satu dengan *scene* lainnya. Pada pembuatan aplikasi ini, penulis menggunakan *actionsript 3.0* atau biasa disebut dengan AS3. Berikut adalah urutan pemberian *actionsript* pada tiap *frame* dan *scene*.

##### 1. Pemberian *ActionScript* di Menu Utama

Urutan pemberian *actionsript* pada AS3 yaitu pertama membuat layer khusus *actionsript* dan simpan di paling atas diantara setiap layer yang ada, dengan jumlah *frame* kosong disesuaikan dengan jumlah *frame* animasi yang telah dibuat. Setelah itu pada *frame* pertama tekan tombol F9 dan munculah *sheet* untuk pemberian *actionsript*. Urutan ini dilakukan pada setiap *frame* atau *scene* yang terdapat animasinya hingga akhir. Lalu pada tahap ini penulis menambahkan action script pada setiap *button* yang ada di menu utama.



**Gambar 4.13**  
Animasi pada Menu Utama



**Gambar 4.14**  
Sheet ActionScript 3.0

Berikut adalah *actionscript* yang penulis tambahkan di setiap *button* pada *scene* menu utama.

```

stop();
Object(this).caw_mulai.addEventListener(MouseEvent.CLICK, cawmulai);
function cawmulai(e:MouseEvent):void{
    nextFrame();
}
import flash.events.MouseEvent;
stop();
Object(this).go_materi.addEventListener(MouseEvent.CLICK, gomateri);
function gomateri(event:MouseEvent):void
{
  
```

```

        gotoAndStop(1, "tatabahasa");
    }
    Object(this).latihan.addEventListener(MouseEvent.CLICK, renshuu);
    function renshuu(event:MouseEvent):void
    {
        gotoAndStop(1, "latihan");
    }
    Object(this).tentang.addEventListener(MouseEvent.CLICK, about);
    function about(event:MouseEvent):void
    {
        gotoAndStop(1, "tentang");
    }
    Object(this).keluar.addEventListener(MouseEvent.CLICK, desuka);
    function desuka(Event:MouseEvent) :void
    {
        gotoAndStop(103);
    }

    Object(this).ya.addEventListener(MouseEvent.CLICK, exitHandler);
    import flash.desktop.NativeApplication;
    import flash.ui.Mouse;
    function exitHandler (event:MouseEvent):void
    {
        NativeApplication.nativeApplication.exit();
    }

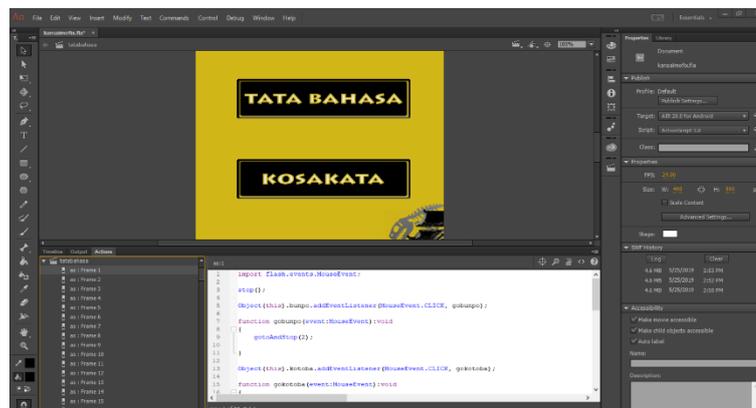
    Object(this).no.addEventListener(MouseEvent.CLICK, menu1);
    function menu1(Event:MouseEvent) :void
    {
        gotoAndStop(102);
    }

```

Pada *actionscript* diatas, terdapat sebuah *script* `addEventListener(MouseEvent.CLICK, "nama perintah")` yang fungsinya untuk memberi perintah pada suatu objek yang telah di *convert* menjadi *button* untuk menghubungkan *frame* yang sedang dibuka menuju *frame* yang akan dituju dengan menyentuh *button* terkait.

## 2. Pemberian *ActionScript* di Menu Materi

Pada tahap ini penulis akan menambahkan *actionscript* pada *scene* ‘tata bahasa’ dan ‘kosakata’. Dalam *scene* tata bahasa terdapat total 38 *frame* dan 24 *layer* dengan 2 *button* penghubung ke tampilan tata bahasa dan kosakata pada *frame* pertama, 21 *button* yang telah dihubungkan dengan *frame* materi tata bahasa, dan sisanya *button* sebagai navigasi untuk *next* dan *back* pada tiap *frame* yang telah dibuat, serta *home* pada tampilan materi untuk kembali ke menu utama. Dengan menyentuh *button* terkait pada *smartphone*, maka pengguna akan diarahkan menuju *frame* dengan judul yang sesuai pada apa yang tercantum pada *button* yang dipilih.



**Gambar 4.15**  
*ActionScript* di *Frame 1 Scene* Tata Bahasa

*Frame* tata bahasa berisikan materi tata bahasa itu sendiri. Sedangkan untuk menghubungkan *button* kosakata dengan *frame* yang akan dituju, maka akan ditambahkan *actionscript* untuk mengarahkan tampilan menuju *scene* kosakata. Berikut adalah *actionscript* yang ditambahkan pada *scene* tata bahasa.

```

import flash.events.MouseEvent;

stop();

Object(this).bunpo.addEventListener(MouseEvent.CLICK, gobunpo);

function gobunpo(event:MouseEvent):void
{
    gotoAndStop(2);
}

Object(this).kotoba.addEventListener(MouseEvent.CLICK, gokotoba);

function gokotoba(event:MouseEvent):void
{
    gotoAndStop(1, "kosakata");
}

Object(this).valik.addEventListener(MouseEvent.CLICK, hayuvalik);

function hayuvalik(event:MouseEvent):void
{
    gotoAndStop(102, "menu_utama");
}

NativeApplication.nativeApplication.addEventListener(KeyboardEvent.KEY_DOWN, handleKeyDown );

function handleKeyDown (e:KeyboardEvent) :void
{
    switch(e.keyCode)
    {
        case Keyboard.BACK:

```

```

        gotoAndStop(102, "menu_utama");

        break;

        case Keyboard.SEARCH:

        case Keyboard.MENU:

            e.preventDefault();

            break;

    }

}

import flash.events.MouseEvent;

stop();

hen.addEventListener(MouseEvent.CLICK, gohens);

function gohens(event:MouseEvent):void{

    gotoAndStop(7);

}

Object(this).naakan.addEventListener(MouseEvent.CLICK, gonaakan);

function gonaakan(event:MouseEvent):void

{

    gotoAndStop(11);

}

Object(this).temoe.addEventListener(MouseEvent.CLICK, gotemoe);

function gotemoe(event:MouseEvent):void

{

    gotoAndStop(13);

}

```

```

Object(this).teya.addEventListener(MouseEvent.CLICK, goteya);
function goteya(event:MouseEvent):void
{
    gotoAndStop(14);
}
Object(this).meirei.addEventListener(MouseEvent.CLICK, gomeirei);
function gomeirei(event:MouseEvent):void
{
    gotoAndStop(15);
}
neks.addEventListener(MouseEvent.CLICK, btneks);
function btneks(e:MouseEvent):void{
    nextFrame();
}
hommateri.addEventListener(MouseEvent.CLICK, gohome);
function gohome(e:MouseEvent):void{
    gotoAndStop(1, "tatabahasa");
}
import flash.events.MouseEvent;
stop();
Object(this).you.addEventListener(MouseEvent.CLICK, goyou);
function goyou(event:MouseEvent):void
{
    gotoAndStop(16);
}

```

```

Object(this).haru.addEventListener(MouseEvent.CLICK, goharu);
function goharu(event:MouseEvent):void
{
    gotoAndStop(17);
}
Object(this).yas.addEventListener(MouseEvent.CLICK, goyas);
function goyas(event:MouseEvent):void
{
    gotoAndStop(18);
}
Object(this).yoru.addEventListener(MouseEvent.CLICK, goyoru);
function goyoru(event:MouseEvent):void
{
    gotoAndStop(19);
}
Object(this).temau.addEventListener(MouseEvent.CLICK, gotemau);
function gotemau(event:MouseEvent):void
{
    gotoAndStop(20);
}
neks2.addEventListener(MouseEvent.CLICK, goneks);
function goneks(e:MouseEvent):void{
    nextFrame();
}
beks2.addEventListener(MouseEvent.CLICK, gobeks);

```

```

function gobeks(e:MouseEvent):void{
    prevFrame();
}
hommateri.addEventListener(MouseEvent.CLICK,gohome1);
function gohome1(e:MouseEvent):void{
    gotoAndStop(1, "tatabahasa");
}
import flash.events.MouseEvent;
stop();
Object(this).kaina.addEventListener(MouseEvent.CLICK, gokaina);
function gokaina(event:MouseEvent):void
{
    gotoAndStop(21);
}
Object(this).tans.addEventListener(MouseEvent.CLICK, gotan);
function gotan(event:MouseEvent):void
{
    gotoAndStop(22);
}
Object(this).toru.addEventListener(MouseEvent.CLICK, gotoru);
function gotoru(event:MouseEvent):void
{
    gotoAndStop(26);
}
Object(this).temi.addEventListener(MouseEvent.CLICK, gotemi);

```

```

function gotemi(event:MouseEvent):void
{
    gotoAndStop(27);
}
Object(this).kate.addEventListener(MouseEvent.CLICK, gokate);
function gokate(event:MouseEvent):void
{
    gotoAndStop(28);
}
neks3.addEventListener(MouseEvent.CLICK, goneks3);
function goneks3(e:MouseEvent):void{
    nextFrame();
}
beks3.addEventListener(MouseEvent.CLICK, gobeks3);
function gobeks3(e:MouseEvent):void{
    prevFrame();
}
hommateri.addEventListener(MouseEvent.CLICK, gohome2);
function gohome2(e:MouseEvent):void{
    gotoAndStop(1, "tatabahasa");
}
import flash.events.MouseEvent;
stop();
Object(this).toku.addEventListener(MouseEvent.CLICK, gotoku);
function gotoku(event:MouseEvent):void

```

```

{
    gotoAndStop(30);
}
Object(this).yanka.addEventListener(MouseEvent.CLICK, goyanka);
function goyanka(event:MouseEvent):void
{
    gotoAndStop(32);
}
Object(this).tageru.addEventListener(MouseEvent.CLICK, gotageru);
function gotageru(event:MouseEvent):void
{
    gotoAndStop(33);
}
Object(this).taru.addEventListener(MouseEvent.CLICK, gotaru);
function gotaru(event:MouseEvent):void
{
    gotoAndStop(34);
}
Object(this).temorota.addEventListener(MouseEvent.CLICK, gotemorota);
function gotemorota(event:MouseEvent):void
{
    gotoAndStop(35);
}
neks4.addEventListener(MouseEvent.CLICK, goneks4);
function goneks4(e:MouseEvent):void{

```

```

        nextFrame();
    }

    beks4.addEventListener(MouseEvent.CLICK,gobeks4);

    function gobeks4(e:MouseEvent):void{
        prevFrame();
    }

    hommateri.addEventListener(MouseEvent.CLICK,gohome3);

    function gohome3(e:MouseEvent):void{
        gotoAndStop(1, "tatabahasa");
    }

    import flash.events.MouseEvent;

    stop();

    Object(this).hitei.addEventListener(MouseEvent.CLICK, gohitei);

    function gohitei(event:MouseEvent):void
    {
        gotoAndStop(36);
    }

    beks5.addEventListener(MouseEvent.CLICK,gobeks5);

    function gobeks5(e:MouseEvent):void{
        prevFrame();
    }

    hommateri.addEventListener(MouseEvent.CLICK,gohome4);

    function gohome4(e:MouseEvent):void{
        gotoAndStop(1, "tatabahasa");
    }
}

```

Berikutnya pada setiap *frame* materi tata bahasa ditambahkan *actionscript* untuk memberikan perintah kepada *button* di tiap *frame* materi jika disentuh. *Template actionscript* yang diberikan adalah sama, hanya yang membedakan adalah tujuan *frame* dari tiap perintah yang diberikan. Berikut *actionscript* yang ditambahkan pada salah satu *frame* materi tata bahasa.

```
stop();  
lanjut1.addEventListener(MouseEvent.CLICK,derlanjut1);  
function derlanjut1(e:MouseEvent):void{  
    nextFrame();  
}  
balik1.addEventListener(MouseEvent.CLICK,derbalik1);  
function derbalik1(e:MouseEvent):void{  
    prevFrame();  
}  
bakkun.addEventListener(MouseEvent.CLICK,gobakkun1);  
function gobakkun1(e:MouseEvent):void{  
    gotoAndStop(2, "tatabahasa");  
}
```

Sedangkan dalam *scene* kosakata terdapat total 104 *frame* dan 84 *layer* dilengkapi beberapa *button* sebagai navigasi untuk *next* dan *back* pada tiap *frame* yang telah dibuat, serta *home* pada tampilan materi untuk kembali ke menu utama sama seperti pada *scene* tata bahasa. Berikut adalah *actionscript* yang ditambahkan pada *scene* kosakata.

```

import flash.events.MouseEvent;

stop();

agyo.addEventListener(MouseEvent.CLICK, goagyo);
function goagyo(event:MouseEvent):void
{
    gotoAndStop(3);
}

kagyo.addEventListener(MouseEvent.CLICK, gokagyo);
function gokagyo(event:MouseEvent):void
{
    gotoAndStop(7);
}

sagyo.addEventListener(MouseEvent.CLICK, gosagyo);
function gosagyo(event:MouseEvent):void
{
    gotoAndStop(9);
}

tagyo.addEventListener(MouseEvent.CLICK, gotagyo);
function gotagyo(event:MouseEvent):void
{
    gotoAndStop(14);
}

nagyo.addEventListener(MouseEvent.CLICK, gonagyo);
function gonagyo(event:MouseEvent):void

```

```

{
    gotoAndStop(16);
}

nexto.addEventListener(MouseEvent.CLICK, gonexto);
function gonexto(e:MouseEvent):void{
    nextFrame();
}

hommateri.addEventListener(MouseEvent.CLICK, golik);
function golik(e:MouseEvent):void{
    gotoAndStop(1, "tatabahasa");
}

import flash.events.MouseEvent;
stop();

hagyo.addEventListener(MouseEvent.CLICK, gohagyo);
function gohagyo(event:MouseEvent):void
{
    gotoAndStop(18);
}

magyo.addEventListener(MouseEvent.CLICK, gomagyo);
function gomagyo(event:MouseEvent):void
{
    gotoAndStop(20);
}

yagyo.addEventListener(MouseEvent.CLICK, goyagyo);

```

```

function goyagyoy(event:MouseEvent):void
{
    gotoAndStop(21);
}
wagyoy.addEventListener(MouseEvent.CLICK, gowagyoy);
function gowagyoy(event:MouseEvent):void
{
    gotoAndStop(22);
}
ngyoy.addEventListener(MouseEvent.CLICK, gongyoy);
function gongyoy(event:MouseEvent):void
{
    gotoAndStop(23);
}
bakkuyoy.addEventListener(MouseEvent.CLICK, gobakkuyoy);
function gobakkuyoy(e:MouseEvent):void{
    prevFrame();
}
hommateriy.addEventListener(MouseEvent.CLICK, golik1);
function golik1(e:MouseEvent):void{
    gotoAndStop(1, "tatabahasa");
}
import flash.events.MouseEvent;
stop();
araheny.addEventListener(MouseEvent.CLICK, goaraheny);

```

```

function goarahen(event:MouseEvent):void
{
    gotoAndStop(24);
}
akan.addEventListener(MouseEvent.CLICK, goakan);
function goakan(event:MouseEvent):void
{
    gotoAndStop(25);
}
akangana.addEventListener(MouseEvent.CLICK, gogana);
function gogana(event:MouseEvent):void
{
    gotoAndStop(26);
}
anjo.addEventListener(MouseEvent.CLICK, goan);
function goan(event:MouseEvent):void
{
    gotoAndStop(27);
}
ichibiri.addEventListener(MouseEvent.CLICK, goichi);
function goichi(event:MouseEvent):void
{
    gotoAndStop(28);
}
nekk.addEventListener(MouseEvent.CLICK, gonekk);

```

```

function gonekk(e:MouseEvent):void{
    nextFrame();
}
bakkun.addEventListener(MouseEvent.CLICK,bakkuna);
function bakkuna(e:MouseEvent):void{
    gotoAndStop(1);
}
import flash.events.MouseEvent;
stop();
iutemo.addEventListener(MouseEvent.CLICK, goiu);
function goiu(event:MouseEvent):void
{
    gotoAndStop(29);
}
ippen.addEventListener(MouseEvent.CLICK, goip);
function goip(event:MouseEvent):void
{
    gotoAndStop(30);
}
ee.addEventListener(MouseEvent.CLICK, goee);
function goee(event:MouseEvent):void
{
    gotoAndStop(31);
}
eenonka.addEventListener(MouseEvent.CLICK, goeen);

```

```

function goeen(event:MouseEvent):void
{
    gotoAndStop(32);
}
erai.addEventListener(MouseEvent.CLICK, goer);
function goer(event:MouseEvent):void
{
    gotoAndStop(33);
}
nekk.addEventListener(MouseEvent.CLICK, gonekk1);
function gonekk1(e:MouseEvent):void{
    nextFrame();
}
bakk.addEventListener(MouseEvent.CLICK, gobakk);
function gobakk(e:MouseEvent):void{
    prevFrame();
}
import flash.events.MouseEvent;
stop();
eraikoccha.addEventListener(MouseEvent.CLICK, goeraik);
function goeraik(event:MouseEvent):void
{
    gotoAndStop(34);
}
egetsunai.addEventListener(MouseEvent.CLICK, goege);

```

```

function goege(event:MouseEvent):void
{
    gotoAndStop(35);
}
okan.addEventListener(MouseEvent.CLICK, gookan);
function gookan(event:MouseEvent):void
{
    gotoAndStop(36);
}
oton.addEventListener(MouseEvent.CLICK, goton);
function goton(event:MouseEvent):void
{
    gotoAndStop(37);
}
ojin.addEventListener(MouseEvent.CLICK, gojin);
function gojin(event:MouseEvent):void
{
    gotoAndStop(38);
}
nekk.addEventListener(MouseEvent.CLICK, gonekk3);
function gonekk3(e:MouseEvent):void{
    nextFrame();
}
bakk.addEventListener(MouseEvent.CLICK, gobakk1);
function gobakk1(e:MouseEvent):void{

```

```

        prevFrame();
    }

    import flash.events.MouseEvent;

    stop();

    oban.addEventListener(MouseEvent.CLICK, goban);

    function goban(event:MouseEvent):void
    {
        gotoAndStop(39);
    }

    ohayo.addEventListener(MouseEvent.CLICK, gohayo);

    function gohayo(event:MouseEvent):void
    {
        gotoAndStop(40);
    }

    omoroi.addEventListener(MouseEvent.CLICK, gomoroi);

    function gomoroi(event:MouseEvent):void
    {
        gotoAndStop(41);
    }

    ookini.addEventListener(MouseEvent.CLICK, gookini);

    function gookini(event:MouseEvent):void
    {
        gotoAndStop(42);
    }

    arigato.addEventListener(MouseEvent.CLICK, goari);

```

```

function goari(event:MouseEvent):void
{
    gotoAndStop(43);
}
bakk.addEventListener(MouseEvent.CLICK,gobakk3);
function gobakk3(e:MouseEvent):void{
    prevFrame();
}
import flash.events.MouseEvent;
stop();
kanawan.addEventListener(MouseEvent.CLICK, gokana);
function gokana(event:MouseEvent):void
{
    gotoAndStop(44);
}
kamahen.addEventListener(MouseEvent.CLICK, gokama);
function gokama(event:MouseEvent):void
{
    gotoAndStop(45);
}
gyosan.addEventListener(MouseEvent.CLICK, gogyo);
function gogyo(event:MouseEvent):void
{
    gotoAndStop(46);
}

```

```

kishoi.addEventListener(MouseEvent.CLICK, gokis);
function gokis(event:MouseEvent):void
{
    gotoAndStop(47);
}
konni.addEventListener(MouseEvent.CLICK, gokon);
function gokon(event:MouseEvent):void
{
    gotoAndStop(48);
}
nekk.addEventListener(MouseEvent.CLICK, gonekk2);
function gonekk2(e:MouseEvent):void{
    nextFrame();
}
bakkun.addEventListener(MouseEvent.CLICK, bakkunk);
function bakkunk(e:MouseEvent):void{
    gotoAndStop(1);
}
import flash.events.MouseEvent;
stop();
kehen.addEventListener(MouseEvent.CLICK, gokeh);
function gokeh(event:MouseEvent):void
{
    gotoAndStop(49);
}

```

```

bakk.addEventListener(MouseEvent.CLICK,gobakk2);
function gobakk2(e:MouseEvent):void{
    prevFrame();
}

```

Sedangkan untuk pilihan materi kosakata nyang lebih rinci, ditambahkan juga *actionsript* dengan format yang sama, dan tujuan perintah yang dibedakan. Sebagai contoh *actionsript* yang ditambahkan pada salah satu *frame* pilihan materi dalam *scene* kosaata adalah sebagai berikut.

```

import flash.events.MouseEvent;
stop();
shomo.addEventListener(MouseEvent.CLICK, goshom);
function goshom(event:MouseEvent):void
{
    gotoAndStop(55);
}

```

```

shaa.addEventListener(MouseEvent.CLICK, goshaa);
function goshaa(event:MouseEvent):void
{
    gotoAndStop(56);
}

```

```

sunma.addEventListener(MouseEvent.CLICK, gosun);
function gosun(event:MouseEvent):void
{

```

```

        gotoAndStop(57);
    }

    sess.addEventListener(MouseEvent.CLICK, gosess);
    function gosess(event:MouseEvent):void
    {
        gotoAndStop(58);
    }

    sora.addEventListener(MouseEvent.CLICK, gosor);
    function gosor(event:MouseEvent):void
    {
        gotoAndStop(59);
    }

    nekk.addEventListener(MouseEvent.CLICK, gyosa1);
    function gyosa1(e:MouseEvent):void{
        nextFrame();
    }

    bakk.addEventListener(MouseEvent.CLICK, gyosb);
    function gyosb(e:MouseEvent):void{
        prevFrame();
    }

    bakkun.addEventListener(MouseEvent.CLICK, gsa1);
    function gsa1(e:MouseEvent):void{
        gotoAndStop(1);
    }

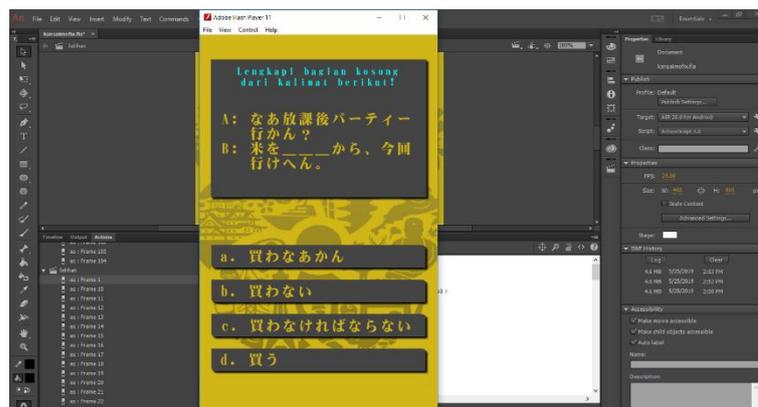
```

Untuk *actionscript* yang penulis tambahkan pada setiap isi dari materi kosakata adalah sebagai berikut.

```
import flash.events.MouseEvent;  
  
stop();  
  
bakku.addEventListener(MouseEvent.CLICK,bakki2);  
  
function bakki2(e:MouseEvent):void{  
  
    gotoAndStop(4);  
  
}
```

### 3. Pemberian *ActionScript* di Menu Latihan

Setelah semua bagian materi selesai ditambahkan *actionscript*, tahap selanjutnya adalah membuat menu latihan yang berfungsi sebagai bahan evaluasi atau sekedar mengukur kemampuan dialek Kansai pengguna. Latihan ini berupa soal kuis yang berisi total 20 kuis. Soal yang tersedia adalah soal yang penulis buat sendiri berdasarkan materi dari buku *Kiiteoboeru Kansai(Osaka)ben Nyuumon*.



**Gambar 4.16**  
Tampilan Menu Latihan

Berikut adalah *actionsript* yang digunakan pada *scene* latihan.

```
import flash.events.MouseEvent;

stop();

hajime.addEventListener(MouseEvent.CLICK, hajimaru);

function hajimaru(event:MouseEvent):void
{
    play();
}

nein.addEventListener(MouseEvent.CLICK, mbung);

function mbung(event:MouseEvent):void
{
    gotoAndStop(102, "menu_utama");
}

var nilai ;
var benar;
var salah;

nilai=0 ;
benar=0;
salah=0;
var jumlahsoal = 0;
function randomQuis(pr1, pr2)
{
    if (pr1 == undefined)
    {
        jumlahsoal++;
        gotoAndStop(pr2);
    }
    else
    {
        randomSoal();
    }

    trace(jumlahsoal+" :jumlah soal");
}

function randomSoal()
```

```

{

    if (jumlahsoal == 20)
    {
        gotoAndStop(31);
        trace("owari");
    }
    else
    {
        var randomNumber = Math.floor(Math.random() * 20 + 1);
        trace(randomNumber);
        switch (randomNumber)
        {
            case 1 :
                randomQuis(s1,11);
                break;
            case 2 :
                randomQuis(s2,12);
                break;
            case 3 :
                randomQuis(s3,13);
                break;
            case 4 :
                randomQuis(s4,14);
                break;
            case 5 :
                randomQuis(s5,15);
                break;
            case 6 :
                randomQuis(s6,16);
                break;
            case 7 :
                randomQuis(s7,17);
                break;
            case 8 :
                randomQuis(s8,18);
                break;
            case 9 :
                randomQuis(s9,19);
                break;
            case 10 :
                randomQuis(s10,20);
                break;
            case 11 :
                randomQuis(s11,21);
                break;
        }
    }
}

```

```

        case 12 :
            randomQuis(s12,22);
            break;
        case 13 :
            randomQuis(s13,23);
            break;
        case 14 :
            randomQuis(s14,24);
            break;
        case 15 :
            randomQuis(s15,25);
            break;
        case 16 :
            randomQuis(s16,26);
            break;
        case 17 :
            randomQuis(s17,27);
            break;
        case 18 :
            randomQuis(s18,28);
            break;
        case 19 :
            randomQuis(s19,29);
            break;
        case 20 :
            randomQuis(s20,30);
            break;
    }
}

```

Untuk tiap soal yang akan ditampilkan, ditambah *actionsript* sebagai berikut.

```

import flash.events.MouseEvent;

stop();
trace(nilai);
trace(benar);
trace(salah);

var s1;
piliha.addEventListener(MouseEvent.CLICK, soal1_a);
function soal1_a(e:MouseEvent):void
{
    randomSoal()
}

```

```

        salah++;
        s1 = 0;
    }
    pilihb.addEventListener(MouseEvent.CLICK, soal1_b);
    function soal1_b(e:MouseEvent):void
    {
        randomSoal()
        salah++;
        s1 = 0;
    }
    pilihc.addEventListener(MouseEvent.CLICK, soal1_c);
    function soal1_c(e:MouseEvent):void
    {
        nilai += 5 ;
        benar++;
        s1 = 1;
        randomSoal()
    }
    pilihd.addEventListener(MouseEvent.CLICK, soal1_d);
    function soal1_d(e:MouseEvent):void
    {
        randomSoal()
        salah++;
        s1 = 0;
    }
}

```

Dan pada akhir bagian penghitungan jumlah skor yang didapat adalah sebagai berikut.

```

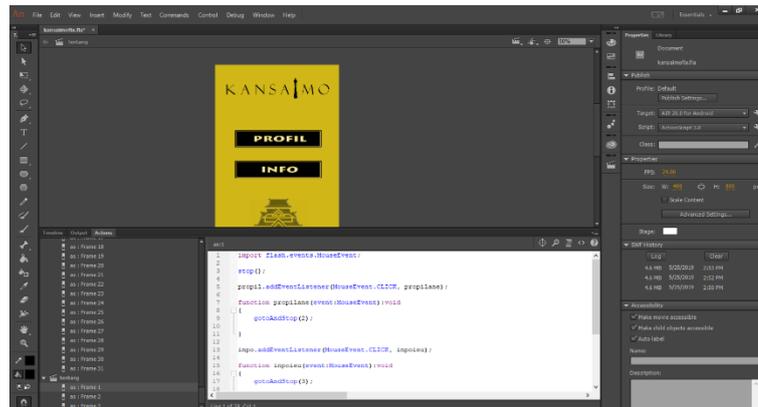
stop();
yarinaos.addEventListener(MouseEvent.CLICK, balikbdg);
function balikbdg(event:MouseEvent):void
{
    NativeApplication.nativeApplication.exit();
}
var angka : Number = 2 ;
benars.text = benar ;
salahs.text = salah ;
skor.text = nilai ;

trace(nilai);
trace(benar);
trace(salah);
jumlahsoal = 0;

```

#### 4. Pemberian *ActionScript* di Menu Tentang

Tahap terakhir dalam pembuatan aplikasi ini adalah menambahkan *actionscrip* pada *scene* ‘tentang’ yang sebelumnya sudah penulis rangkai sedemikian rupa. *Scene* ini berisikan informasi mengenai penulis dan *Kansaimo*.



**Gambar 4.17**  
Tampilan Menu Tentang

Berikut adalah *actionscrip* yang ditambahkan pada *scene* ini.

```
import flash.events.MouseEvent;
stop();
propil.addEventListener(MouseEvent.CLICK, propilane);
function propilane(event:MouseEvent):void
{
    gotoAndStop(2);
}
inpo.addEventListener(MouseEvent.CLICK, inpoieu);
function inpoieu(event:MouseEvent):void
{
    gotoAndStop(3);
}
hombotang.addEventListener(MouseEvent.CLICK, balikhom);

function balikhom(event:MouseEvent):void
```

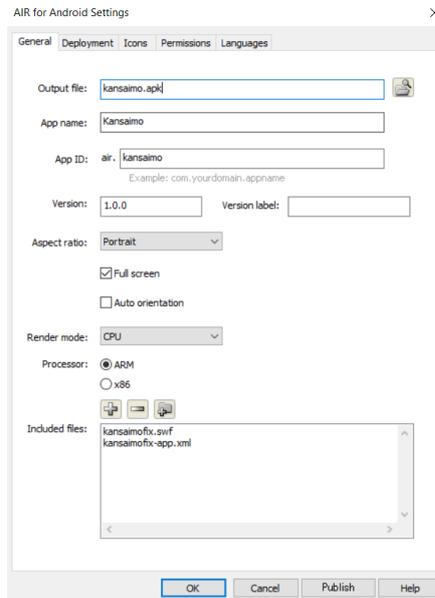
```
{
    gotoAndStop(102, "menu_utama"); }
```

Selain itu pada setiap tampilan informasi terdapat button yang berfungsi sebagai navigasi untuk kembali ke menu awal dari scene tentang, dengan actionscrip yang ditambahkan seperti dibawah ini.

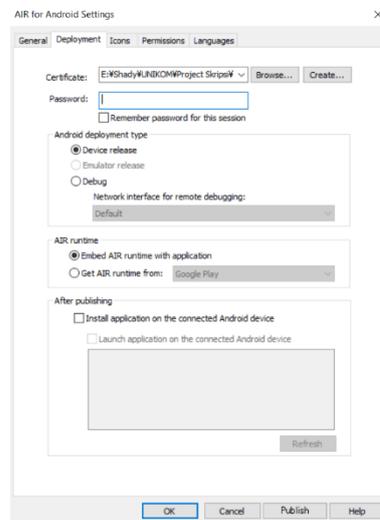
```
import flash.events.MouseEvent;
kembali.addEventListener(MouseEvent.CLICK, kembalilah);
function kembalilah(event:MouseEvent):void
{
    gotoAndStop(1);
}
```

#### **4.1.1.6 Publish Aplikasi**

Setelah serangkaian proses pembuatan aplikasi selesai, langkah terakhir untuk membuat *file* yang telah dirangkai menjadi sebuah aplikasi yang dapat dijalankan di semua *smartphone android* adalah dengan *publish file* aplikasinya. File aplikasi yang dapat diinstall di *smartphone android* adalah *file* dengan format *.apk*, sehingga pada tahap ini penulis membuat *file* aplikasi ini dengan nama *kansaimo.apk*. Langkah *publish* yang dilakukan penulis adalah dengan memilih pilihan *Publish* dalam menu *File*, selanjutnya penulis mengatur pengaturan aplikasi dengan memberi nama aplikasi, dan beberapa pengaturan teknis lainnya pada submenu *General*, dilanjutkan ke submenu *Deployment* dengan mengatur penyimpanan keamanan hak cipta, terakhir di menu *Icon* penulis menambahkan *icon* aplikasi untuk dijadikan *icon* dalam jajaran menu di *smartphone* berbasis *android*.



**Gambar 4.18**  
Pengaturan *General* pada Menu *Publish*

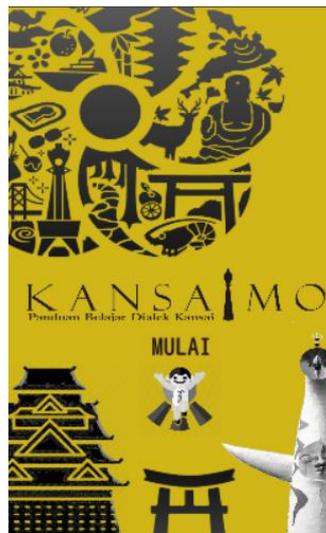


**Gambar 4.19**  
Pengaturan *Deployment* pada Menu *Publish*

#### 4.1.2 Tahap Penggunaan Aplikasi *Kansaimo*

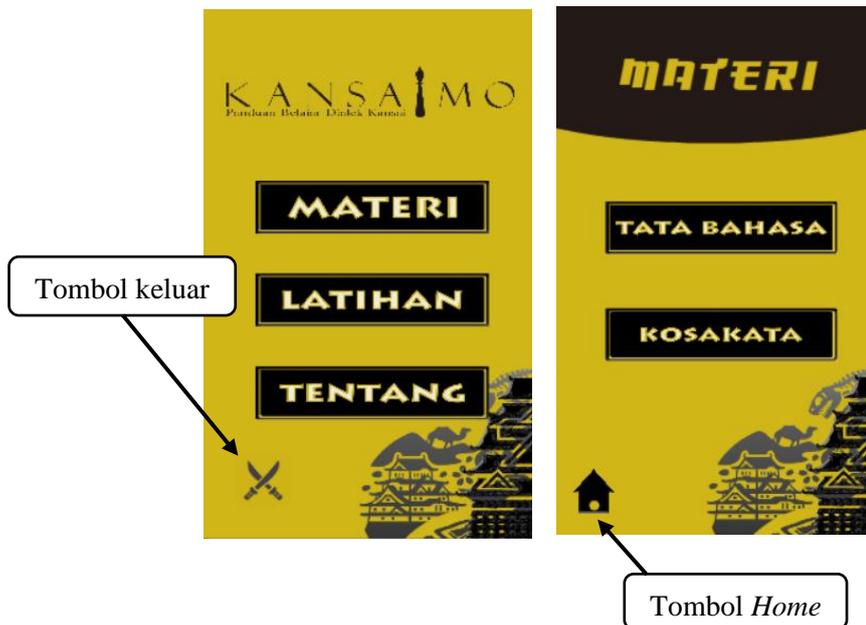
*Output* dari aplikasi yang telah penulis buat adalah file *kansaimo.apk*, berikut adalah tahapan untuk menggunakan aplikasi *Kansaimo*.

1. *Install* aplikasi *Kansaimo* pada *smartphone* berbasis *android*.
2. Setelah aplikasi terpasang, akan muncul tampilan *cover* aplikasi seperti yang sudah tersaji dalam gambar 4.20, sentuh *button* ‘mulai’ untuk memulai aplikasi dengan masuk ke menu utama. Dalam beberapa layer terdapat *icon* gapura merah, ini adalah *button* yang berfungsi sebagai *button back* atau kembali ke layer asal.



**Gambar 4.20**  
Tampilan *Cover* Aplikasi

3. Kemudian menu utama, simbol pedang yang disilang adalah untuk keluar, pilih *button* ‘materi’ untuk masuk ke bagian pemahaman materi, seperti yang sudah tersaji dalam gambar 4.21 berikut:



**Gambar 4.21**  
Tampilan Menu Utama dan Materi

- Di menu materi terdapat icon home untuk kembali ke manu utama, selanjutnya pilih *button* 'tata bahasa' untuk mempelajari pola kalimat umum dialek Kansai.



**Gambar 4.22**  
Tampilan Menu Tata Bahasa dan Isi

- Setelah itu kembali ke menu *Materi* untuk memilih *button* ‘kosakata’ untuk mempelajari kosakata beserta pengucapannya dalam dialek Kansai.



Tombol Gapura Merah

**Gambar 4.23**  
Tampilan Materi Kosakata

- Kembali lagi ke menu utama, sentuh *button* ‘latihan’ untuk mengetahui sejauh mana pengetahuan dialek Kansai yang telah dipelajari.



**Gambar 4.24**  
Tampilan Menu Latihan

7. Untuk memilih menu *Tentang*, di menu utama pilih button ‘tentang’, untuk masuk ke tampilan informasi, tersaji dalam gambar 4.25 berikut:



**Gambar 4.25**  
Tampilan Menu Tentang II

#### 4.1.3 Penilaian Terhadap Aplikasi *Kansaimo*

Untuk mengetahui penilaian dari responden mengenai aplikasi *Kansaimo*, penulis memberikan suatu perangkat smartphone android untuk melakukan pengujian aplikasi yang sudah dibuat, setelah responden melakukan percobaan terhadap aplikasi yang telah dibagikan, penulis juga menyertakan kuesioner melalui *Google Form* kepada mahasiswa Program Studi Sastra Jepang Universitas Komputer Indonesia yang sedang aktif sebagai mahasiswa tingkat tiga atau semester enam sebanyak 26 orang. Kuesioner berupa pertanyaan dengan dua macam kategori, yaitu mengenai tampilan aplikasi yang berjumlah enam soal dan

fungsi media yang berjumlah empat soal, maka jumlah keseluruhan pertanyaan adalah sepuluh soal.

Berikut adalah hasil dari penilaian responden terhadap aplikasi *Kansaimo* sebagai media pembelajaran dialek Kansai.

#### 4.1.3.1 Penilaian Responden Terhadap Tampilan Aplikasi

Berikut adalah tabel hasil penilaian responden terhadap pernyataan “animasi yang terdapat pada aplikasi *Kansaimo* menarik”, seperti yang sudah terpaparkan dalam tabel 4.1 berikut:

**Tabel 4.1**  
**Animasi**

		Frequency	Percent	Valid Percent	Mean
Valid	TS	1	3.8	3.8	
	S	16	61.5	61.5	
	SS	9	34.6	34.6	
	Total	26	100.0	100.0	4.27

Dari tabel 4.1 di atas, terlihat bahwa sebagian besar responden (61,5%) setuju bahwa animasi yang terdapat dalam aplikasi *Kansaimo* menarik, sebagian kecil responden (34,6%) sangat setuju dengan tampilan animasi dalam aplikasi *Kansaimo* menarik, dan sedikit sekali responden (3,8%) tidak setuju bahwa animasi yang terdapat dalam aplikasi *Kansaimo* menarik. Sehingga dengan nilai total mean sebesar 4,27 dapat diasumsikan bahwa penilaian responden berada dalam rentang sangat setuju dan setuju dengan animasi yang terdapat dalam aplikasi *Kansaimo* menarik.

Selanjutnya adalah tabel hasil penilaian responden terhadap pernyataan “pemilihan *font* dalam aplikasi *Kansaimo* terlihat jelas”, seperti yang sudah terpaparkan dalam tabel 4.2 berikut:

**Tabel 4.2**  
**Pemilihan *Font***

		Frequency	Percent	Valid Percent	Mean
Valid	TS	2	7.7	7.7	
	S	22	84.6	84.6	
	SS	2	7.7	7.7	
	Total	26	100.0	100.0	3.92

Dari tabel 4.2 di atas, terlihat bahwa sebagian besar responden (84,6%) setuju bahwa *font* dalam aplikasi *Kansaimo* terlihat jelas, sedangkan sedikit sekali responden (7,7%) sangat setuju *font* dalam aplikasi *Kansaimo* terlihat jelas, dan juga sedikit sekali (7,7%) tidak setuju bahwa *font* dalam aplikasi *Kansaimo* terlihat jelas. Dan dengan nilai total mean sebesar 3,92 dapat diasumsikan responden cenderung setuju bahwa *font* dalam aplikasi *Kansaimo* terlihat jelas.

Sedangkan tabel di bawah ini merupakan hasil penilaian responden terhadap pernyataan “teks dalam aplikasi *Kansaimo* mudah dibaca”, seperti yang sudah terpaparkan dalam tabel 4.3 berikut:

**Tabel 4.3**  
**Teks**

		Frequency	Percent	Valid Percent	Mean
Valid	TS	3	11.5	11.5	
	S	16	61.5	61.5	
	SS	7	26.9	26.9	
	Total	26	100.0	100.0	4.04

Dilihat dari tabel 4.3 di atas, sebagian besar responden (61,5%) setuju bahwa teks dalam aplikasi *Kansaimo* mudah dibaca, sedangkan sebagian kecil responden (26,9%) sangat setuju teks dalam aplikasi *Kansaimo* mudah dibaca, dan sedikit sekali responden (11,5%) tidak setuju bahwa teks dalam aplikasi *Kansaimo* mudah dibaca. Dan dengan nilai total mean sebesar 4,04 dapat diasumsikan bahwa responden setuju dengan teks dalam aplikasi *Kansaimo* mudah dibaca.

Untuk hasil penilaian responden terhadap pernyataan “tombol navigasi dalam aplikasi *Kansaimo* berfungsi dengan baik”, disajikan dalam tabel 4.4 berikut:

**Tabel 4.4**  
**Tombol Navigasi**

		Frequency	Percent	Valid Percent	Mean
Valid	TS	1	3.8	3.8	
	S	14	53.8	53.8	
	SS	11	42.3	42.3	
	Total	26	100.0	100.0	4.35

Dari tabel 4.4 di atas, terlihat bahwa lebih dari setengahnya responden (53,8%) setuju bahwa tombol navigasi dalam aplikasi *Kansaimo* berfungsi dengan

baik, dan hampir setengahnya responden (42,3%) sangat setuju bahwa tombol navigasi dalam aplikasi *Kansaimo* berfungsi dengan baik, sedangkan sedikit sekali responden (3,8%) tidak setuju bahwa tombol navigasi dalam aplikasi *Kansaimo* berfungsi dengan baik. Dan dengan nilai total mean sebesar 4,35 dapat diasumsikan bahwa penilaian responden berada dalam rentang sangat setuju dan setuju dengan tombol navigasi dalam aplikasi *Kansaimo* berfungsi dengan baik.

Untuk hasil penilaian responden terhadap pernyataan “efek suara pada aplikasi *Kansaimo* berfungsi dengan baik”, disajikan dalam tabel 4.5 berikut:

**Tabel 4.5**  
**Efek Suara**

		Frequency	Percent	Valid Percent	Mean
Valid	TS	1	3.8	3.8	
	S	18	69.2	69.2	
	SS	7	26.9	26.9	
	Total	26	100.0	100.0	4.19

Dari tabel 4.5 di atas, terlihat bahwa sebagian besar responden (69,2%) setuju bahwa efek suara pada aplikasi *Kansaimo* berfungsi dengan baik, dan sebagian kecil responden (26,9%) sangat setuju bahwa efek suara pada aplikasi *Kansaimo* berfungsi dengan baik, sedangkan sedikit sekali (3,8%) tidak setuju bahwa efek suara pada aplikasi *Kansaimo* berfungsi dengan baik. Dan dengan nilai total mean sebesar 4,19 dapat diasumsikan bahwa responden setuju dengan efek suara pada aplikasi *Kansaimo* berfungsi dengan baik.

Untuk hasil penilaian responden terhadap pernyataan “tampilan secara keseluruhan dari aplikasi *Kansaimo* menarik”, disajikan dalam tabel 4.6 berikut:

**Tabel 4.6**  
**Tampilan Secara Keseluruhan**

		Frequency	Percent	Valid Percent	Mean
Valid	TS	2	7.7	7.7	
	S	15	57.7	57.7	
	SS	9	34.6	34.6	
	Total	26	100.0	100.0	4.19

Dilihat dari tabel 4.6 di atas, lebih dari setengahnya responden (57,7%) setuju bahwa tampilan secara keseluruhan dari aplikasi *Kansaimo* menarik, sedangkan sebagian kecil responden (34,6%) sangat setuju bahwa tampilan secara keseluruhan dari aplikasi *Kansaimo* menarik, dan sangat sedikit responden (7,7%) yang tidak setuju bahwa tampilan secara keseluruhan dari aplikasi *Kansaimo* menarik. Dan dengan nilai total mean sebesar 4,19 dapat diasumsikan bahwa responden setuju dengan tampilan secara keseluruhan dari aplikasi *Kansaimo* menarik.

#### **4.1.3.2 Penilaian Responden Terhadap Fungsi Aplikasi**

Dari segi fungsinya, berikut hasil penilaian responden terhadap pernyataan “penjelasan materi dalam aplikasi *Kansaimo* mudah dipahami”, seperti yang sudah tersaji dalam tabel 4.7 di bawah:

**Tabel 4.7**  
**Penjelasan Materi**

		Frequency	Percent	Valid Percent	Mean
Valid	TS	1	3.8	3.8	
	S	20	76.9	76.9	
	SS	5	19.2	19.2	
	Total	26	100.0	100.0	4.12

Dilihat dari tabel 4.7 di atas, sebagian besar responden (76,9%) setuju dengan penjelasan materi dalam aplikasi *Kansaimo* mudah dipahami, dan sebagian kecil responden (19,2%) sangat setuju dengan penjelasan materi dalam aplikasi *Kansaimo* mudah dipahami, sedangkan sisanya sedikit sekali responden (3,8%) tidak setuju dengan penjelasan materi dalam aplikasi *Kansaimo* mudah dipahami. Dan dengan nilai total mean sebesar 4,12 dapat diasumsikan bahwa responden setuju penjelasan materi dalam aplikasi *Kansaimo* mudah dipahami.

Selanjutnya adalah tabel hasil penilaian responden terhadap pernyataan “contoh pengucapan kosakata yang terdapat dalam aplikasi *Kansaimo* memudahkan pengguna dalam mempelajari dialek Kansai”, yang sudah terpaparkan dalam tabel 4.8 berikut:

**Tabel 4.8**  
**Fitur Pengucapan Kosa Kata**

		Frequency	Percent	Valid Percent	Mean
Valid	S	19	73.1	73.1	
	SS	7	26.9	26.9	
	Total	26	100.0	100.0	4.27

Dilihat dari tabel 4.8 di atas, sebagian besar responden (73,1%) setuju dengan fitur pengucapan kosakata yang terdapat dalam aplikasi *Kansaimo* memudahkan pengguna dalam mempelajari dialek Kansai, dan sebagian kecil responden (26,9%) sangat setuju dengan contoh pengucapan kosakata yang terdapat dalam aplikasi *Kansaimo* memudahkan pengguna dalam mempelajari dialek Kansai. Dan dengan nilai total mean sebesar 4,27 dapat diasumsikan bahwa penilaian responden berada pada rentang antara sangat setuju dan setuju dengan contoh pengucapan kosakata yang terdapat dalam aplikasi *Kansaimo* memudahkan pengguna dalam mempelajari dialek Kansai.

Selanjutnya adalah tabel hasil penilaian responden terhadap pernyataan “aplikasi *Kansaimo* sangat membantu dalam mempelajari dialek Kansai”, tersaji dalam tabel 4.9 berikut:

**Tabel 4.9**  
**Aplikasi Kansaimo Sangat Membantu**

		Frequency	Percent	Valid Percent	Mean
Valid	S	13	50.0	50.0	
	SS	13	50.0	50.0	
	Total	26	100.0	100.0	4.5

Dilihat dari tabel 4.9 di atas, setengah dari responden (50%) masing masing setuju dan sangat setuju bahwa aplikasi *Kansaimo* sangat membantu dalam mempelajari dialek Kansai. Dan dengan nilai total mean sebesar 4,5 dapat

diasumsikan bahwa responden sangat setuju dengan aplikasi *Kansaimo* sangat membantu dalam mempelajari dialek Kansai.

Untuk hasil penilaian responden terhadap pernyataan “aplikasi *Kansaimo* lebih menarik daripada metode konvensional”, disajikan dalam tabel 4.10 berikut:

**Tabel 4.10**  
**Aplikasi Kansaimo Terhadap Metode Konvensional**

		Frequency	Percent	Valid Percent	Mean
Valid	TS	1	3.8	3.8	
	S	17	65.4	65.4	
	SS	8	30.8	30.8	
	Total	26	100.0	100.0	4.23

Dilihat dari tabel 4.10 di atas, sebagian besar responden (65,4%) setuju bahwa aplikasi *Kansaimo* lebih menarik daripada metode konvensional, dan sebagian kecil responden (30,8%) sangat setuju bahwa aplikasi *Kansaimo* lebih menarik daripada metode konvensional, sedangkan sangat sedikit responden (3,8%) tidak setuju dengan aplikasi *Kansaimo* lebih menarik daripada metode konvensional. Dan dengan nilai total mean sebesar 4,23 dapat diasumsikan bahwa responden setuju bahwa aplikasi *Kansaimo* lebih menarik daripada metode konvensional.

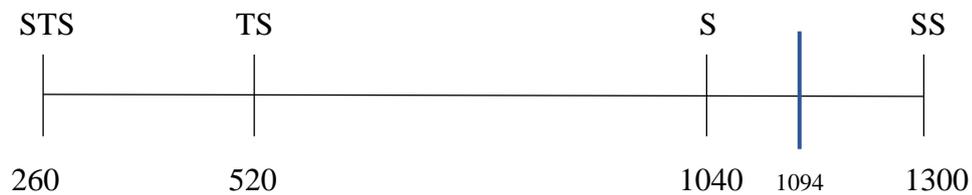
#### 4.1.3.3 Penghitungan Data Hasil Kuesioner Secara Keseluruhan

**Tabel 4.11**  
Hasil Responden Aplikasi *Kansaimo*

NO	PERTANYAAN	JAWABAN				Presentase
		SS	S	TS	STS	%
1	Animasi yang terdapat dalam aplikasi <i>Kansaimo</i> menarik.	9	16	1	0	
2	Pemilihan font dalam aplikasi <i>Kansaimo</i> terlihat jelas.	2	22	2	0	
3	Teks dalam aplikasi <i>Kansaimo</i> mudah dibaca.	7	16	3	0	
4	Tombol navigasi yang terdapat pada aplikasi <i>Kansaimo</i> berfungsi dengan baik.	11	14	1	0	
5	Efek suara pada aplikasi <i>Kansaimo</i> berfungsi dengan baik.	7	18	1	0	
6	Tampilan secara keseluruhan dari aplikasi <i>Kansaimo</i> menarik.	9	15	2	0	
7	Penjelasan materi dalam aplikasi <i>Kansaimo</i> mudah dipahami.	5	20	1	0	
8	Contoh pengucapan kosakata yang terdapat dalam aplikasi <i>Kansaimo</i> memudahkan pengguna dalam mempelajari dialek Kansai.	7	19	0	0	
9	Aplikasi <i>Kansaimo</i> sangat membantu dalam mempelajari dialek Kansai.	13	13	0	0	
10	Aplikasi <i>Kansaimo</i> lebih menarik daripada metode konvensional.	8	17	1	0	
$\Sigma$		78	170	12	0	
$\Sigma$ Kriterium = Nilai Kategori * Jawaban Pertanyaan		390	680	24	0	
<b>TOTAL NILAI KRITERIUM</b>		1094				
Nilai Presentase = Total Nilai Kriterium : Nilai Maksimum * 100 ( 1094 : 1300 *100 )						84.15%

Dari hasil data kuesioner secara keseluruhan yang didapat, hasil perhitungan *rating scale* adalah  $390 + 680 + 24 + 0 = 1094$ . Skor tersebut merupakan skor akhir dan dapat dimasukkan ke dalam *rating scale* dengan nilai maksimum 1300. Berikut

adalah skor akhir yang sudah ditentukan jarak intervalnya dengan menggunakan *rating scale* nilai maksimum.



**Gambar 4.26**  
Hasil Nilai Kriteria dari Kuesioner

Untuk mengetahui indikator penilaian berdasarkan dari skala jawaban, skor akhir juga disesuaikan dengan kategori *rating scale* yang sudah ada sebelumnya. Skor akhir yang didapatkan dari hasil penghitungan adalah 1094, garis biru pada *interval scale* berada pada skala 1040 – 1300 yang berarti secara keseluruhan responden menjawab sangat setuju pada setiap pernyataan yang ada pada kuesioner.

Setelah itu, hasil jawaban responden dihitung persentasenya dengan rumus sebagai berikut.

$$P = \frac{1094}{1300} \times 100\% = 84.15\%$$

Berdasarkan hasil perhitungan di atas, didapatkan presentase dengan nilai 84.15%. Dengan demikian, dapat disimpulkan bahwa sebagian besar responden menyatakan bahwa aplikasi *Kansaimo* merupakan media pembelajaran dialek Kansai yang sangat menarik.

## **4.2 Pembahasan Hasil Penelitian**

Pembahasan hasil penelitian ini akan mendeskripsikan hasil penelitian untuk menjawab beberapa masalah yang telah dibahas pada bab sebelumnya yaitu bagaimana proses pembuatan aplikasi *Kansaimo* berbasis android ini, kemudian bagaimana cara mengoperasikan aplikasi *Kansaimo* ini, serta bagaimana penilaian responden terhadap aplikasi *Kansaimo*. Penjelasannya adalah sebagai berikut.

### **4.2.1 Proses Pembuatan Aplikasi *Kansaimo***

Dalam proses pembuatan aplikasi *Kansaimo* ini, sesuai dengan tahapan pengembangan metode ADDIE, terdapat lima tahapan yang harus dilakukan, antara lain pertama adalah tahap analisis (*analysis*) yaitu dengan menganalisis masalah yang sedang dihadapi, dan memilah materi dan data yang tepat untuk dimasukkan kedalam aplikasi yang akan dibuat. Selanjutnya adalah tahap desain (*design*) dengan merancang aplikasi yang akan dibuat dengan membuat skema aplikasi, setelah itu membuat desain aplikasi, dalam membuat desain aplikasi *Kansaimo* ini, penulis menggunakan *software* desain pendukung yaitu Adobe Photoshop CC untuk membuat beberapa *button* navigasi, *background*, dan beberapa *icon* pendukung tampilan lain. Selanjutnya tahap ketiga adalah memasukan desain aplikasi yang telah dibuat dengan cara *import file* objek yang sudah didesain ke dalam *software* pembangun aplikasi *Kansaimo* yaitu Adobe Animate CC 2015. Tahap yang ketiga adalah tahap pengembangan (*development*) yaitu mengembangkan semua objek yang telah didesain sebelumnya dengan cara membuat animasi dari beberapa objek yang sudah di-*import* tadi, animasi yang

dibuat disini adalah *motion tween*, perpindahan *layer*, animasi *button* dan beberapa animasi *motion* yang dibuat di *background*. Setelah semua animasi selesai dibuat, selanjutnya adalah penambahan *actionscript* sebagai kunci utama bekerjanya animasi yang sudah dibuat sebelumnya, pemberian *actionscript* ini dilakukan pada setiap *layer* yang ada, seperti *layer* menu utama, materi, latihan, juga beberapa untuk menghidupkan fungsi *button* untuk navigasi yang ada di setiap *layer*. Dan terakhir setelah penambahan *actionscript* adalah dengan mem-*publish file* animasi yang telah dibuat menjadi *file* aplikasi agar dapat di-*install* oleh perangkat *android*. Tahap yang keempat adalah tahap implementasi (*implementation*), yaitu dengan melakukan uji coba pemakaian terhadap pengguna atau responden untuk memberikan penilaian terhadap aplikasi yang berhasil dibuat. Tahap yang terakhir adalah evaluasi (*evaluation*) yaitu dengan mengolah data hasil tahap implementasi untuk dapat hasil akhir dan melakukan evaluasi terhadap kekurangan aplikasi yang telah dibuat.

#### **4.2.2 Pengoperasian Aplikasi Kansaimo**

Untuk mengoperasikan aplikasi ini cukup mudah, tahap utama yang harus dilakukan adalah dengan memasang atau install aplikasi *Kansaimo* ini ke dalam perangkat *android*. Setelah itu akan muncul logo Universitas Komputer Indonesia dan akan hilang dengan sendirinya, bersamaan dengan itu akan muncul menu cover sebagai halaman utama dari aplikasi, cukup dengan menyentuh tombol 'mulai' maka pengguna akan diarahkan ke menu utama. Jika sudah masuk menu utama, pengguna akan bebas memilih menu apa saja yang terdapat dalam menu utama

yakni menu materi, latihan dan tentang. Juga ada tombol pedang berbentuk X yang berfungsi untuk keluar dari aplikasi.

Dalam menu materi, pengguna diarahkan kepada materi tata bahasa dan kosakata dialek Kansai, dalam menu tata bahasa terdapat materi yang sudah disediakan penulis hasil pengumpulan data. Untuk menu kosakata, maka akan terdapat menu yang mengarahkan materi sesuai huruf hiragana pertama yang digunakan untuk memudahkan pencarian kosakata yang diinginkan. Dalam menu materi ini juga terdapat tombol navigasi seperti *next* dan *back* yang umum digunakan, dan ada tombol *home* berbentuk kuil merah untuk tombol kembali ke menu utama.

Dalam menu latihan, pengguna akan diarahkan ke layer latihan untuk mengetahui sejauh mana kemampuan dialek Kansai yang dimiliki oleh pengguna. Di menu latihan ini terdapat 20 soal berbentuk melengkapi bagian yang kosong dalam suatu kalimat atau percakapan. Untuk menu tentang berisikan informasi mengenai aplikasi dan penulis.

Aplikasi *Kansaimo* ini juga memiliki media grafis atau visual yang mampu menyampaikan pesan atau isi materi berupa animasi atau gambar kepada pengguna, media audio yang tersedia di menu materi kosakata, dan media proyeksi yang berguna dalam perangsangan visual agar pembelajar tidak bosan dalam menggunakan media menurut Sadiman (1996). Hal ini juga didukung dengan pernyataan Huda (2013) yang menyatakan komponen aplikasi android meliputi *activities, services, contact provider, dan broadcast receiver*.

### 4.2.3 Penilaian Responden

Berdasarkan hasil dari penilaian responden melalui kuisisioner mengenai tampilan dan fungsi aplikasi yang penulis bagikan kepada 26 orang mahasiswa Sastra Jepang tingkat tiga Universitas Komputer Indonesia, sebagian besar responden menyatakan setuju bahwa tampilan aplikasi *Kansaimo* ini menarik, dan jika dinilai dari fungsinya sebagian besar responden menyatakan aplikasi *Kansaimo* memudahkan pengguna dalam mempelajari dialek Kansai.

Dari hasil penghitungan data kuisisioner secara keseluruhan, skor *rating scale* yang didapat adalah 1094 dari nilai maksimum 1300 yang berarti secara keseluruhan responden menjawab sangat setuju pada setiap pernyataan yang diberikan. Dalam penghitungan persentase didapat nilai 84.15% sehingga dapat disimpulkan bahwa sebagian besar responden menilai aplikasi *Kansaimo* merupakan media pembelajaran dialek Kansai yang sangat menarik.

Untuk penilaian responden dari ahli dalam bidang multimedia Universitas Komputer Indonesia, Acep Wanda, ST menyatakan aplikasi *Kansaimo* layak dijadikan aplikasi pembelajaran. Selain itu, aplikasi *Kansaimo* juga memenuhi unsur *hardware* dan unsur *message/software* sebagai syarat yang harus dipenuhi menurut Hasnida (2014) sebagai media pembelajaran. Juga aplikasi *Kansaimo* ini telah memenuhi enam kriteria penilaian multimedia interaktif menurut Thorn W. dalam buku Hujair H. A. Sanaky (2013) yaitu kemudahan navigasi, ada kandungan kognisi, pengetahuan dan presentasi informasi, integrasi media, menarik minat pelajar, serta fungsi secara keseluruhan.