

BAB IV PEMBAHASAN

4.1 Ulasan awal pengetahuan rambu-rambu lalu lintas dan kelengkapan berkendara sebelum bermain *game* berbasis *Virtual Reality*.

Keprihatinan peneliti terhadap angka kematian anak-anak pada usia 0-14 tahun yang disebabkan oleh kecelakaan lalu lintas menduduki posisi ketiga dalam daftar penyebab kematian anak-anak yang di terbitkan oleh WHO[1]. Hal tersebut mendorong peneliti untuk mengetahui seberapa jauh pengetahuan anak-anak SD di kota Semarang tentang rambu-rambu lalu lintas dan kelengkapan berkendara yang sesuai dengan peraturan berlaku. Kuisoner yang telah dibagikan kepada 30 responden dan diolah telah menghasilkan data sebagai berikut.

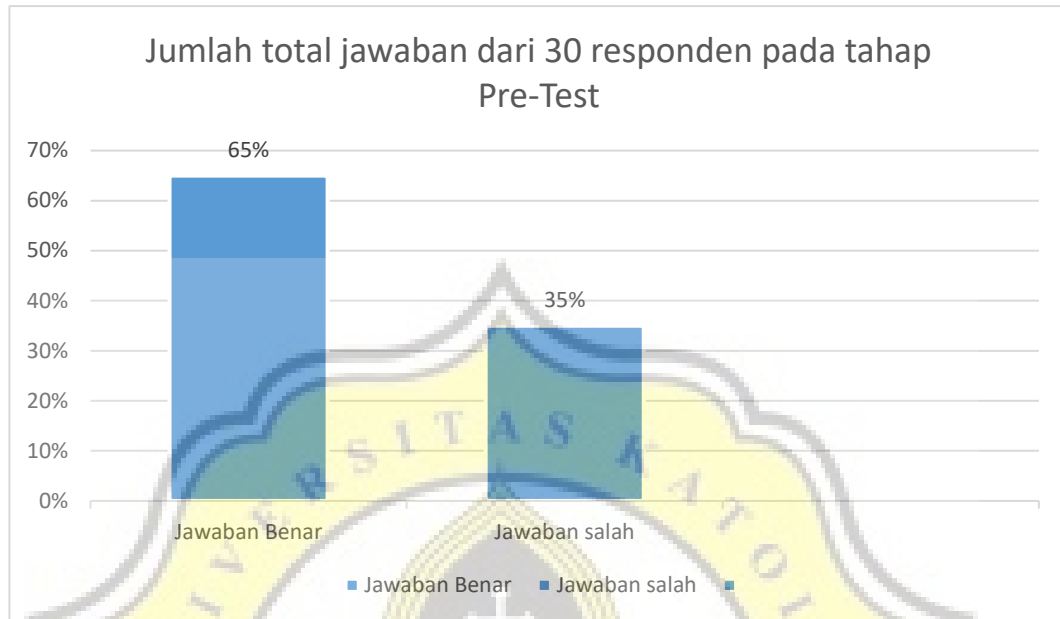
Data awal menunjukkan 97% (29 dari 30 anak) tidak pernah bermain *game* menggunakan teknologi *Virtual Reality*, seperti yang digambarkan oleh gambar 4.1 di bawah ini.



Gambar 4.1 Grafik Jumlah responden yang pernah bermain *game Virtual Reality*

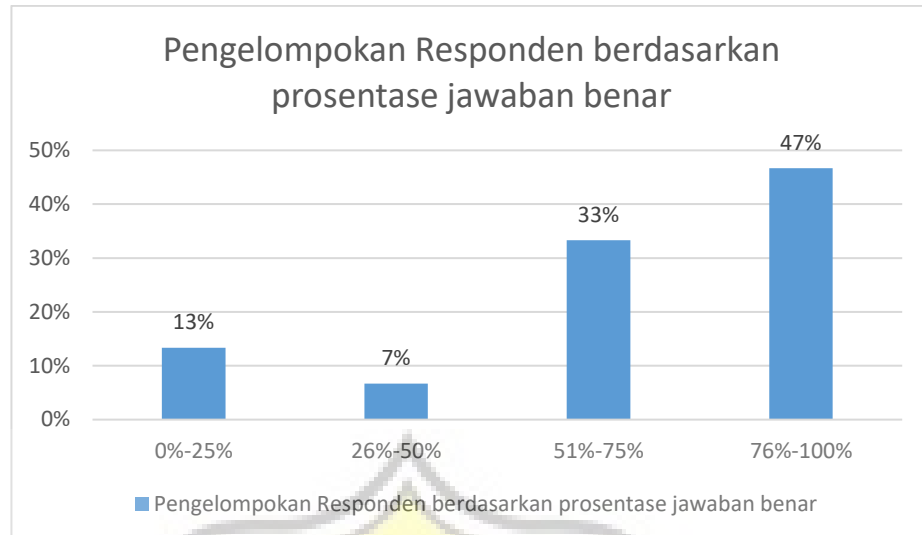
Dan berdasarkan jumlah total jawaban dari 30 responden mengenai rambu-rambu lalu lintas dan artinya yang diberikan dalam kuisoner, sebanyak 65% atau sejumlah

292 jawaban dari responden adalah benar sedangkan 35% atau 157 jawaban lainnya merupakan jawaban salah seperti tercatat dalam gambar 4.2 di bawah ini.



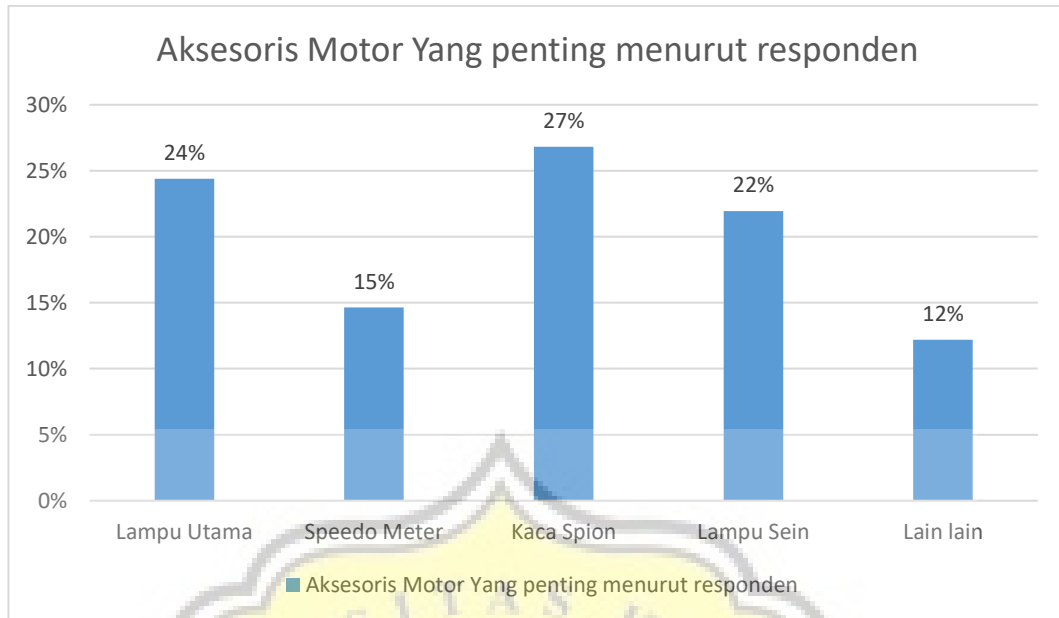
Gambar 4.2 Grafik Hasil Pre-Test tentang rambu dan artinya

Kemampuan jawab masing-masing responden sangatlah berbeda satu sama lain. Sejumlah 13% (4 responden dari 30 responden) hanya mampu menjawab dengan benar sebanyak 0-25% dari pertanyaan yang diberikan, kemudian 7% (2 responden dari 30 responden) mampu menjawab sebanyak 26%-50%. Sebanyak 33% (10 responden dari 30 responden) mampu menjawab dengan sebanyak 51%-75%, kemudian 47% (14 responden dari 30 responden) mampu menjawab dengan sebanyak 76%-100% seperti dipaparkan pada gambar 4.3 di bawah ini.



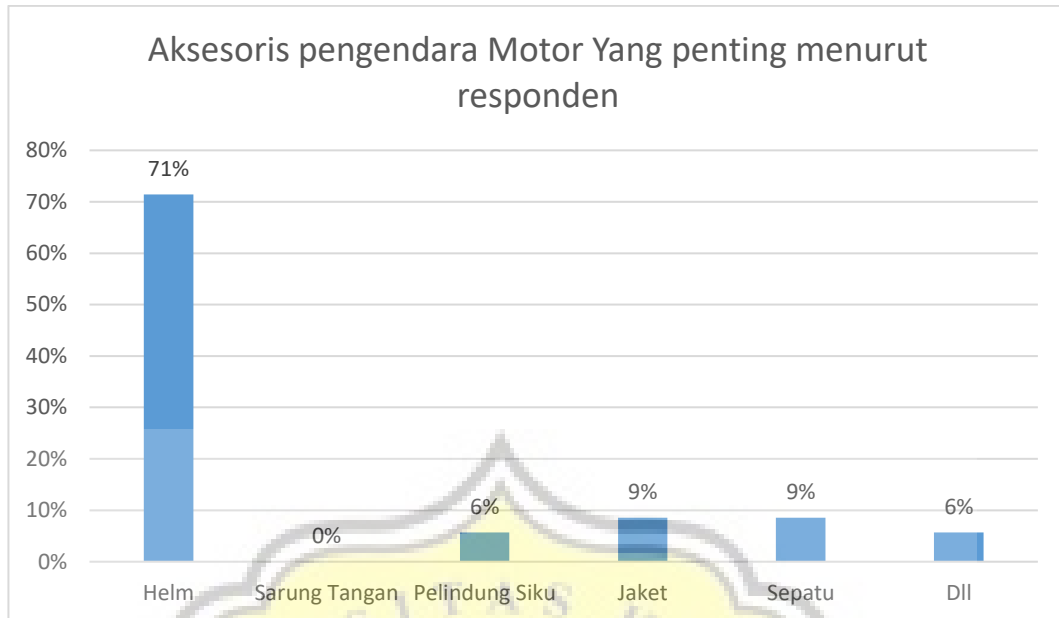
Gambar 4.3 Grafik Pengelompokan Responden berdasarkan prosentase jawaban benar

Pengetahuan anak-anak tentang apa yang harus ada dan dipakai sewaktu menaiki maupun mengendarai sepeda motor tersebar rata pada masing-masing bagian seperti tertampilkan di dalam gambar 4.4 dan gambar 4.5 di bawah ini. Pada gambar 4.4 menunjukkan bahwa kaca spion duduk di peringkat pertama pilihan responden dengan prosentase sebesar 27% kemudian disusul lampu utama sebesar 24%. Sedangkan lampu sein sebesar 22%, *speedometer* 15%, dan jawaban lain-lain sebesar 12%.

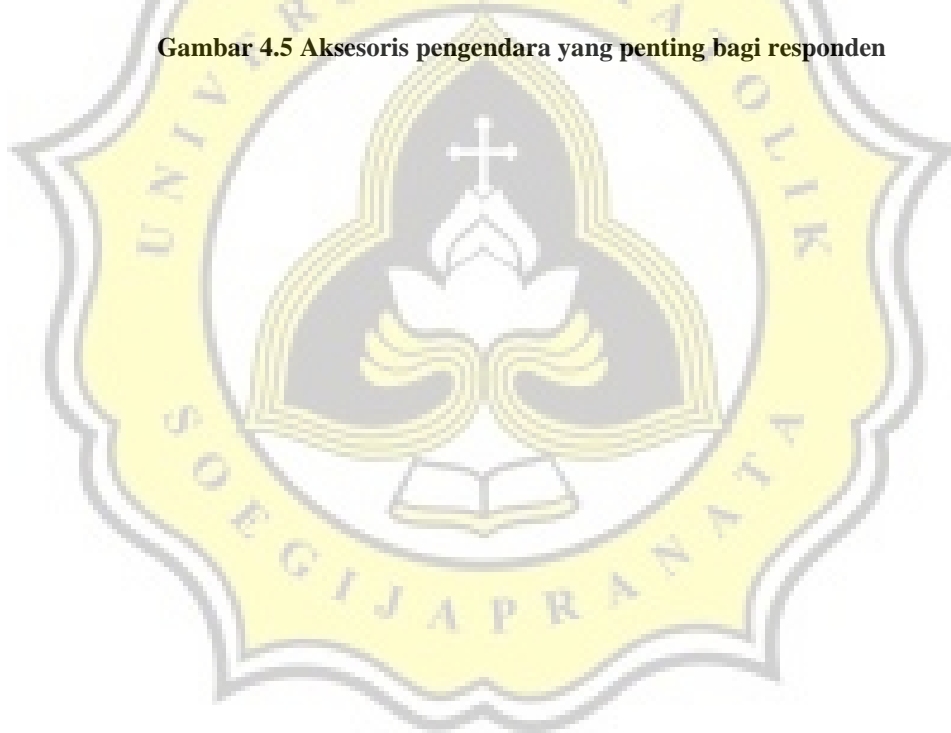


Gambar 4.4 Grafik Aksesoris Motor yang penting bagi responden

Dapat dilihat pada gambar 4.5 menyatakan sebesar 71% anak-anak mengerti dan tahu bahwa helm adalah salah satu aksesoris penting bagi mereka. 6% dari responden menyatakan pelindung siku penting bagi mereka, sedangkan sebanyak 9% mengisi jaket sebagai perlengkapan yang penting. Sepatu mendapatkan 9% dari seluruh pilihan, sisanya memilih lain-lain atau tidak menjawab.

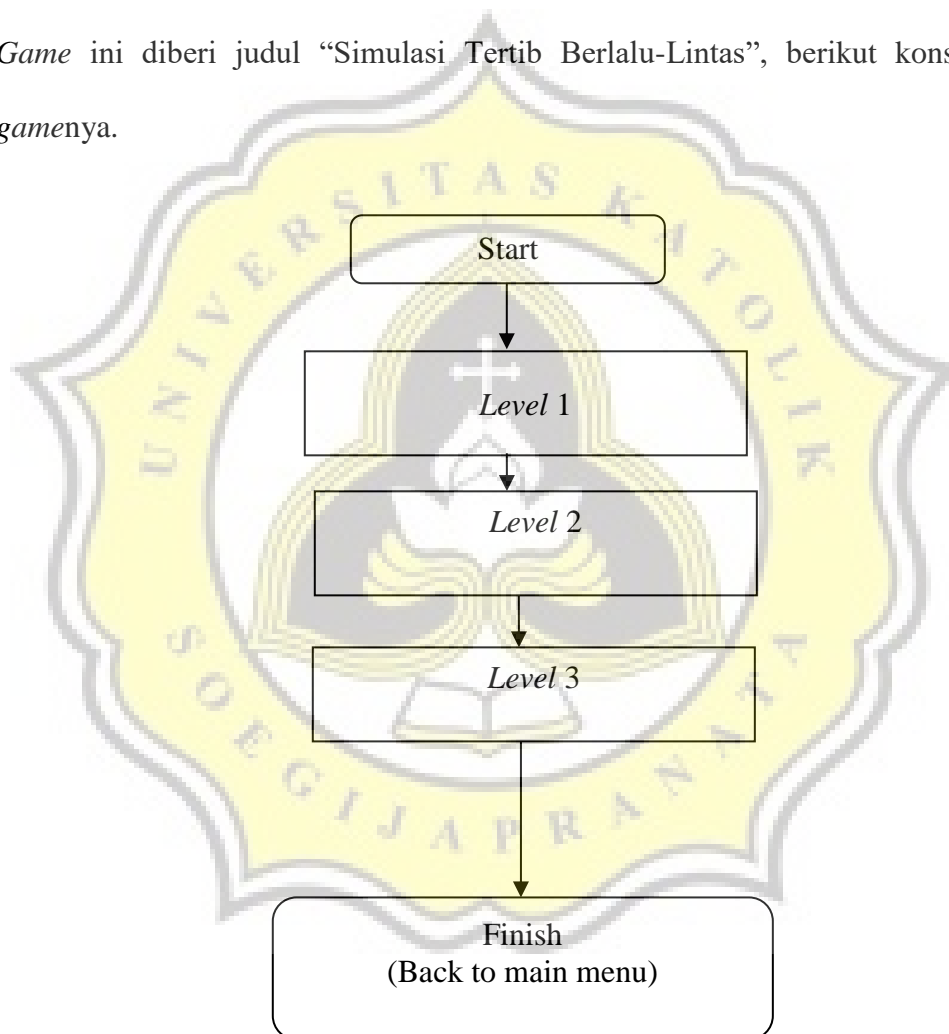


Gambar 4.5 Aksesoris pengendara yang penting bagi responden



4.2 Konsep Pembuatan *Game Virtual Reality* untuk Edukasi Berlalu Lintas

Berdasar pada penjelasan pada sub-bab sebelumnya, maka desain dari *game* yang dibuat haruslah menarik, sederhana dan mudah dipahami oleh pengguna. Sehingga efek dari penggunaan media *Virtual Reality* untuk edukasi berlalu lintas dapat dirasakan seperti nyata dan memberi kesenangan serta pengetahuan secara bersamaan walau menggunakan realitas alternatif yang dibangun melalui program. *Game* ini diberi judul “Simulasi Tertib Berlalu-Lintas”, berikut konsep alur *gamenya*.



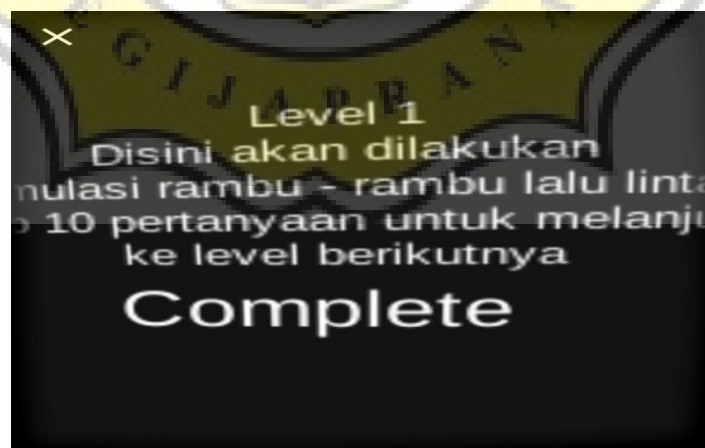
Gambar 4.6 Alur *Game*

Desain tampilan *game Virtual Reality* yang menggunakan objek 3 dimensi yang ditata untuk membangun suasana seperti dunia nyata. Pada *level 1 gameplay* dibangun seperti seorang polisi yang akan bertugas namun mengikuti ujian. *Level 1* dilakukan tanya jawab dengan lokasi di depan kantor polisi seperti gambar 4.7 di bawah ini.



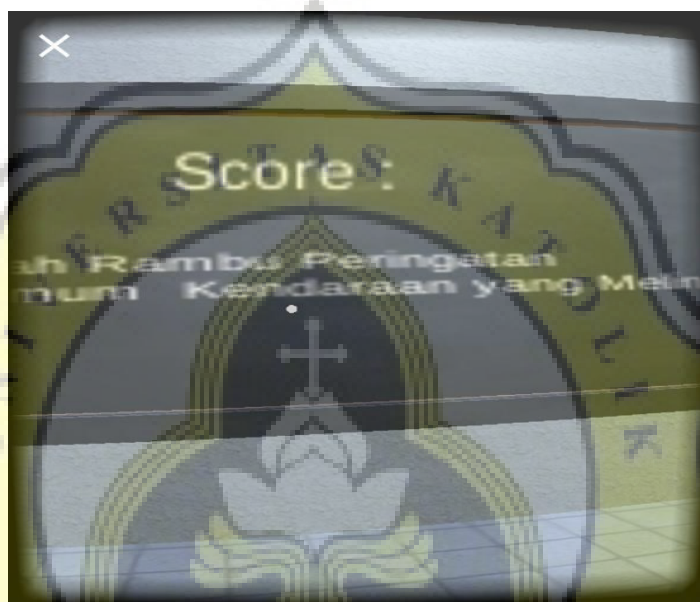
Gambar 4.7 Gameplay level 1

Setiap berpindah *level* akan muncul tampilan *loading* yang berisi tentang kondisi dan petunjuk untuk menang pada *level* tersebut seperti gambar 4.8 di bawah ini.



Gambar 4.8 loading screen

Pada *level 2* dibangun suasana kelas, dimana pemain merasakan kondisi ujian yang lebih nyata. Pada *level 2* seperti gambar diatas memperlihatkan pertanyaan yang ditempel pada papan tulis, sehingga menciptakan pengalaman yang cukup nyata. Dengan pertanyaan rambu-rambu yang jarang ditemui di tempat umum, membangun suasana yang lebih menegangkan bagi anak-anak seperti gambar 4.9 di bawah ini.



Gambar 4.9 *Gameplay level 2*

Pada *level 3* pemain akan dibawa menuju sebuah jalan dimana banyak pengendara motor yang lewat. Pemain akan menjadi seperti polisi yang sedang berpatroli dan menempelkan surat tilang pada para pelanggar. Disini pemain akan diuji perilaku mana yang melanggar dan mana yang tidak melanggar seperti gambar 4.10 di bawah ini.



Gambar 4.10 *Gameplay* level 3

4.2.1 Materi Pembelajaran dalam *Game Virtual Reality*

Berikut adalah tabel gambar materi rambu-rambu lalu lintas yang digunakan dalam *game Virtual Reality* berjudul "Simulasi Tertib Berlalu Lintas". Materi diambil berdasarkan buku dan observasi pribadi, buku yang digunakan berjudul "Pintas Mengenal Rambu Lalu Lintas" karya Ichan Suhada. Dengan menggabungkan pengamatan dan sumber buku, materi disusun dengan mengambil 17 rambu yang sering dilihat pada level 1 dan 14 rambu yang jarang dilihat pada level 2. Sehingga jumlah rambu yang ada pada *game Virtual Reality* ini berjumlah 31 rambu, yang terdiri dari 7 rambu peringatan, 12 rambu larangan, 6 rambu perintah, dan 6 rambu petunjuk.

No	Gambar Rambu	Arti Rambu	Jenis
1		Dilarang memutar arah	Larangan
2		Dilarang Belok Kanan	Larangan
3		Dilarang Belok Kiri	Larangan
4		Dilarang Berhenti	Larangan
5		Pejalan Kaki Dilarang Masuk	Larangan
6		Dilarang Parkir	Larangan
7		Hati-Hati Jalan Licin	Peringatan

8		Hati-Hati perlintasan kereta api	Peringatan
9		Hati-Hati Persimpangan	Peringatan
10		Hati-Hati Rawan Longsor	Peringatan
11		Hati-Hati Jalan Menanjak	Peringatan
12		Wajib Mengikuti Arah/Lajur Kiri	Perintah
13		Wajib Mengikuti Arah/Lajur Kanan	Perintah
14		Wajib Berbelok ke Kiri	Perintah

15		Wajib Berbelok ke Kanan	Perintah
16		Wajib Mengikuti Arah putaran	Perintah
17		Wajib Untuk Pengendara Sepeda	Perintah

Gambar 4.11 Tabel rambu level 1

No	Gambar Rambu	Arti Rambu	Jenis
1		Dilarang Masuk	Larangan
2		Dilarang Berhenti	Larangan
3		Dilarang masuk bagi semua kendaraan dari kedua arah	Larangan

4		Larangan Masuk bagi kendaraan dengan lebar maksimum...	Larangan
5		Dilarang Melebihi kecepatan.....	Larangan
6		Dilarang Berjalan Terus Apabila menghambat arus lalu lintas	Larangan
7		Pengarah tikungan Ke Kanan	Peringatan
8		Lintasan Pesawat Terbang	Peringatan
9		Keluar jalan Tol	Petunjuk
10		Masuk Jalan Tol	Petunjuk

11		Keluar dari wilayah Kota Semarang	Petunjuk
12		Memasuki wilayah Kota Semarang	Petunjuk
13		Boleh Memutar Balik	Petunjuk
14		Jalan Buntu	Petunjuk

Gambar 4.12 Tabel rambu level 2

4.2.2 Script Pembuatan Game Virtual Reality

Pada *level 1* dan *level 2* sistem yang mengatur *game* menggunakan algoritma bilangan acak yang dicek secara terpusat sehingga hanya memerlukan satu file program untuk *level 1* dan *level 2*. Berikut adalah *source code* yang di gunakan pada *level 1* dan *level 2*. Menggunakan metode input *GVR Cardboard* terbaru yaitu menggunakan sentuhan yang dilakukan melalui mekanisme tombol pada perangkat kaca mata sesuai dengan *SDK* yaitu versi 1.6. Berikut pembahasan berbagai fungsi yang di gunakan dalam *game*.

Void Start pada level 1 dan 2 berguna sebagai tahapan pertama yang akan dijalankan oleh *script*. Pada *void Start*, *script* akan memuat semua komponen yang di butuhkan yang sebelumnya sudah dideklarasikan. Pada

script diatas ditunjukkan beberapa komponen seperti teks untuk menampilkan pertanyaan dan mencatat skor, kemudian menginialisasi *array* penyimpanan, dan memanggil fungsi yang akan bertugas berikutnya seperti yang ditunjukkan *script* 4.1.

```
// Use this for initialization
void Start () {
    catat= GameObject.Find("catatscore").GetComponent<Text>();
    pertanyaan = GameObject.Find ("pertanyaan").GetComponent<Text> ();

    simpannomorobject = new int[3];
    simpanpertanyaan = new string[16];
    simpanyangdijawab = new int[11];
    z = 0;
    ready = 0;

    var soundsource = GetComponents<AudioSource> ();
    bgm = soundsource [2];
    correct = soundsource [0];
    wrong = soundsource [1];
    bgm.Play ();
    buatobjek ();
    countera = 0;
}

```

Script 4.1 Void Start level 1 dan 2

Void *buatobjek* pada level 1 dan 2 berisi berbagai fungsi dari menentukan jawaban dan pertanyaan, mengacak penempatannya, serta mengatur animasi dan posisi, pada *script* 4.2 di bawah ini ditunjukkan cara memilih jawaban dan pertanyaan menggunakan indeks *array* secara acak, kemudian mengacak penempatannya.

```
void buatobjek (){
    Kunci = Random.Range (0, jumlahrandom);
    if (simpanyangdijawab.Contains (Kunci)) {
        Kunci = Random.Range (0, jumlahrandom);
        checksoal ();
    }
    jawaban = nomorasset [Kunci];
    simpanjawaban = jawaban.name;
}

```

```
wadahkunci = Random.Range (0, 3);
simpannomorobject [wadahkunci] = Kunci;
```

Script 4.2 Mengacak pertanyaan dan jawaban

Void buatobjek pada level 1 dan 2 berisi berbagai fungsi dari menentukan jawaban dan pertanyaan, mengacak penempatannya, serta mengatur animasi dan posisi, pada *script* 4.3 di bawah ini ditunjukkan cara memilih jawaban dan pertanyaan menggunakan indeks *array* secara acak, kemudian mengacak penempatannya.

```
for (int i = 0; i < 2; i++) {
    wadah = Random.Range (0, 3);
    if (wadah == wadahkunci) {
        cektempat ();
    }
    buatrando = Random.Range (0, jumlahrando);
    print (buatrando);
    while (simpannomorobject.Contains (buatrando) && buatrando != Kunci ) {
        buatrando = Random.Range (0, jumlahrando);
        print (buatrando);
    }
    simpannomorobject [wadah] = buatrando;
}
```

Script 4.3 Mengacak posisi dan pilihan yang lain

Pada level 1 dan level 2 menggunakan *script* yang sama seperti *script* 4.4 di bawah ini juga bertugas untuk menampilkan pertanyaan, menampilkan jawaban, serta menambahkan beberapa komponen pada objek seperti *Box Collider* dan *Event Trigger*.

```
target1 = Instantiate (nomorasset [objek1], new Vector3 (0, 0,
4.0f), Quaternion.identity);
target2 = Instantiate (nomorasset [objek2], new Vector3 (4.0f, 0, 0.0f), Quate
rnion.Euler (0, 90, 0));
target3 = Instantiate (nomorasset [objek3], new Vector3 (-4.0f, 0, -
0.0f), Quaternion.Euler (0, -90, 0));
```



```

pertanyaan.text = simpanpertanyaan [Kunci];

target1.transform.localScale = new Vector3(0.7f, 0.7f, 0.7f);
target2.transform.localScale = new Vector3(0.7f, 0.7f, 0.7f);
target3.transform.localScale = new Vector3(0.7f, 0.7f, 0.7f);

target1.AddComponent<BoxCollider> ();
target2.AddComponent<BoxCollider> ();
target3.AddComponent<BoxCollider> ();

EventTrigger eventTrigger1 = target1.AddComponent<EventTrigger> ();
EventTrigger eventTrigger2 = target2.AddComponent<EventTrigger> ();
EventTrigger eventTrigger3 = target3.AddComponent<EventTrigger> ();

EventTrigger.Entry entry1 = new EventTrigger.Entry ();
EventTrigger.Entry entry2 = new EventTrigger.Entry ();
EventTrigger.Entry entry3 = new EventTrigger.Entry ();

entry1.eventID = EventTriggerType.PointerDown;
entry1.callback.AddListener (( data ) => {
    OnPointerDownDelegate ((PointerEventData)data);
});

entry2.eventID = EventTriggerType.PointerDown;
entry2.callback.AddListener (( data ) => {
    OnPointerDownDelegate ((PointerEventData)data);
});

entry3.eventID = EventTriggerType.PointerDown;
entry3.callback.AddListener (( data ) => {
    OnPointerDownDelegate ((PointerEventData)data);
});

eventTrigger1.triggers.Add (entry1);
eventTrigger2.triggers.Add (entry2);
eventTrigger3.triggers.Add (entry3);

ready = 1;
}

```

Script 4.4 Menampilkan jawaban dan pertanyaan, menambahkan komponen pada objek level 1 dan 2

Fungsi *void* checksoal pada level 1 dan 2 dipanggil saat memilih jawaban dan pertanyaan untuk di tampilkan, fungsi ini berguna untuk memeriksa apakah hasil acak tersebut pernah di tampilkan atau tidak setelah pertama kali diperiksa oleh fungsi buatobjek.

```
void checksoal () {
```

```

if (simpanyangdijawab.Contains (Kunci)) {
    Kunci = Random.Range (0, jumlahrandom);
    simpanyangdijawab [z] = Kunci;
} else {
    simpannomorobject [wadahkunci] = Kunci;
}
return;
}

```

Script 4.5 Memeriksa ulang pertanyaan dan jawaban yang di tampilkan

Fungsi cektempat berguna untuk memeriksa penempatan jawaban dan soal setelah sebelumnya diperiksa juga pada fungsi buatobjek seperti yang ditunjukkan pada *script* 4.6.

```

void cektempat () {
    wadah = Random.Range (0, 2);
    if (wadah == wadahkunci) {
        cektempat ();
    }
    return;
}

```

Script 4.6 Memeriksa ulang penempatan pertanyaan dan jawaban yang di tampilkan

Fungsi *void Update* pada level 1 dan 2 berisi tentang berbagai fungsi yang harus dipanggil setiap saat seperti animasi putaran, memeriksa score setiap saat sehingga ketika score memenuhi kondisi *game* akan lanjut ke level berikutnya seperti *script* 4.7 di bawah ini.

```

void Update () {

    if (ready == 1) {
        putar ();
    }

    if ( score == 10 ){
        bgm.Stop ();
        SceneManager.LoadSceneAsync ("Loading 2", LoadSceneMode.Single);
    }
}

```

Script 4.7 Void Update

Fungsi *Void* putar adalah menganimasikan objek yang ada di level 1 dan 2 *game*.

```
public void putar ()
{
    target1.transform.Rotate(0,100* Time.deltaTime,0);
    target2.transform.Rotate(0,100* Time.deltaTime,0);
    target3.transform.Rotate(0,100* Time.deltaTime,0);
}
```

Script 4.8 script untuk menganimasikan objek

Menggunakan input sentuhan pada layar fungsi *OnPointerDownDelegate* membaca perintah input yang diberikan. Dengan *Var* namaobjekterpilih menyimpan nama objek yang ditunjuk *pointer* dan membandingkan dengan kunci jawaban yang ada, kemudian bila benar menambah skor bila tidak lanjut ke pertanyaan selanjutnya pada level 1 dan 2.

```
public void OnPointerDownDelegate( PointerEventData data )
{
    var namaobjekterpilih = data.pointerCurrentRaycast.gameObject.name.Replace("(Clone)","");

    if (namaobjekterpilih.Equals (simpanjawaban)) {
        if(z==0){
            simpanyangdijawab [countera] = Kunci;
        }else{
            countera += 1;
            simpanyangdijawab[countera] = Kunci;
        }
    }
    correct.Play ();
    score += 1 ;
    simpanscore = score.ToString ();
    catat.text = "Score : " + simpanscore;
    z += 1;
    ready = 0;
    Destroy (target1);
    Destroy (target2);
    Destroy (target3);
    buatobjek ();
}
```

```

}
else
{
    wrong.Play ();
    ready = 0;
    z += 1;
    Destroy (target1);
    Destroy (target2);
    Destroy (target3);
    buatobjek ();
}
}
}
}

```

Script 4.9 Script input via tombol sentuh

Pada *level 3 script* yang digunakan ada dua yaitu *script* untuk menembak dan untuk mekanisme dasar. Pada *Void start script* utama level 3 hampir sama dengan *script* level 1 dan 2 dengan tambahan parameter *spawnstate* untuk mengindikasikan apakah *scene* level 3 siap untuk melakukan pemanggilan objek kembali seperti yang diterangkan *script* 4.10 dibawah ini.

```

// Use this for initialization
void Start () {
    spawnstate = 0;

    var soundsource = GetComponent<AudioSource> ();
    bgm = soundsource [0];

    bgm.Play ();

    buatobjek ();
    InvokeRepeating ("respawn", 0, 10);
}

```

Script 4.10 Void Start level 3

Pada level 3 *void* *buatobjek* memiliki fungsi untuk mengatur penempatan model 3 pelanggar, kecepatannya, serta berapa jumlah pelanggarnya seperti yang diperlihatkan di *scrip* 4.11, dimana metode *random* digunakan untuk mengacak jumlah pertanyaan serta penempatannya.

```

void buatobjek (){
tampil = new int[3];
jumlah = Random.Range (1, 4);
if (jumlah == 1) {
    pelanggaran = Random.Range (1, 4);
    z = Random.Range (0, 3);
    tampil [z] = pelanggaran;
} else if (jumlah == 2)
    for (i = 0; i < 2; i++) {
        pelanggaran = Random.Range (1, 4);
        z = Random.Range (0, 3);
        nomor = tampil [z];
        while (nomor != 0) {
            z = Random.Range (0, 3);
        }
        catat = z.ToString ();
        print (catat);
        tampil [z] = pelanggaran;
    }
else if (jumlah == 3) {
    tampil [0] = Random.Range (1, 4);
    tampil [1] = Random.Range (1, 4);
    tampil [2] = Random.Range (1, 4);
}
    pertama = tampil [0];
    kedua = tampil [1];
    ketiga = tampil [2];

    jarak1 = Random.Range (1.0f, 3.0f);
    jarak2 = Random.Range (4.0f, 6.0f);
    jarak3 = Random.Range (7.0f, 9.0f);

    speed1 = Random.Range (1, 10) * Time.deltaTime;
    speed2 = Random.Range (1, 10) * Time.deltaTime;
    speed3 = Random.Range (1, 10) * Time.deltaTime;

    objek1 = Instantiate (chara[pertama],new Vector3 (Random.Range(-10.0f,-
15.0f), 0, 4.0f), Quaternion.Euler (0, 90, 0));
    objek2 = Instantiate (chara[kedua],new Vector3 (Random.Range(-10.0f,-
15.0f), 0, 7.0f), Quaternion.Euler (0, 90, 0));
    objek3 = Instantiate (chara[ketiga],new Vector3 (Random.Range(-10.0f,-
15.0f), 0, 10.0f), Quaternion.Euler (0, 90, 0));
}

```

Script 4.11 void buatobjek level 3

Pada level 3 *script* ada fungsi yang berbeda yaitu fungsi *public void respawn* dimana memiliki fungsi memeriksa apakah objek 3 dimensi sudah

dihilangkan dari *game* atau belum, ketika sudah maka fungsi *respawn* akan memanggil fungsi *buatobjek* untuk memanggil objek lainnya.

```
public void respawn(){
    if(spawnstate == 0){
        if (objek1 == null && objek2 == null && objek3 == null) {
            print ("coba");
            buatobjek ();
        }
    }
}
```

Script 4.12 Script respawn

Untuk menembak di *level 3*, menggunakan metode input sama seperti *level 1* dan *level 2*, dengan tambahan memanggil objek kemudian menggerakkan objek tersebut menuju kemanapun pemain melihat.

```
void Update () {
    var countibject = GameObject.FindGameObjectsWithTag ("bullet");
    if (countibject.Length == 5) {
        trigerred = 0;
    }else{
        trigerred = 1;
    }
    if (Input.GetButtonDown("Fire1")) {
        summon ();
    }
}

public void summon (){
    if (trigerred == 1){
        var headposition = GvrViewer.Instance.HeadPose.Position;
        var headorientation = GvrViewer.Instance.HeadPose.Orientation;
        spawned = Instantiate (bullet, new Vector3 (headposition.x, headposition.y + (2.0f), headposition.z), headorientation);
        go = spawned.GetComponent<Rigidbody> ();
        go.AddForce (spawned.transform.forward * speed);
    }
}
}
```

Script 4.4 Script menembak Level 3

4.3 Efek Desain dan Mekanisme *Game* Terhadap Responden

Game yang dirancang menarik, sederhana dan mudah dipahami oleh anak-anak kelas 3 sampai dengan kelas 6 SD dapat memainkannya. Dengan mengacak 10 pertanyaan dari bank soal level 1 dan level 2 responden akan merasakan efek acak sehingga akan penasaran dengan pertanyaan berikutnya. Pada level 3 dengan *Game* yang diberi nama “Simulasi Tertib Berlalu lintas” ini telah diujikan kepada responden secara acak di kota Semarang sebanyak 30 orang dengan rentang kelas 3 sampai dengan kelas 6 SD.



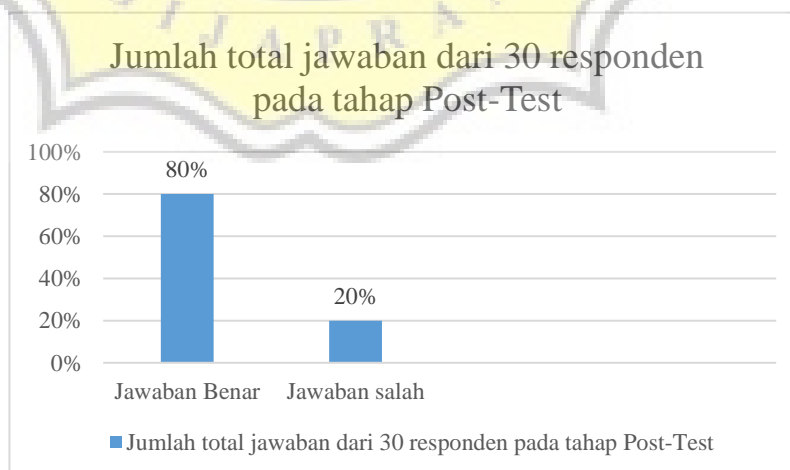
Gambar 4.13 Kegiatan Pengujian *game Virtual Reality*

Pengujian diperlukan untuk memastikan apakah desain dan materi yang ada didalam *game* dapat di terima dengan baik oleh responden. Pengujian dilakukan secara *door to door* di lokasi kota Semarang, selain itu metode *door to door* dipilih untuk meningkatkan pengetahuan para orang tua responden tentang *game* yang bertema edukasi bagi anak-anak.

4.3.1 Ulasan hasil pengujian dan perbandingan data hasil pengujian awal dan akhir

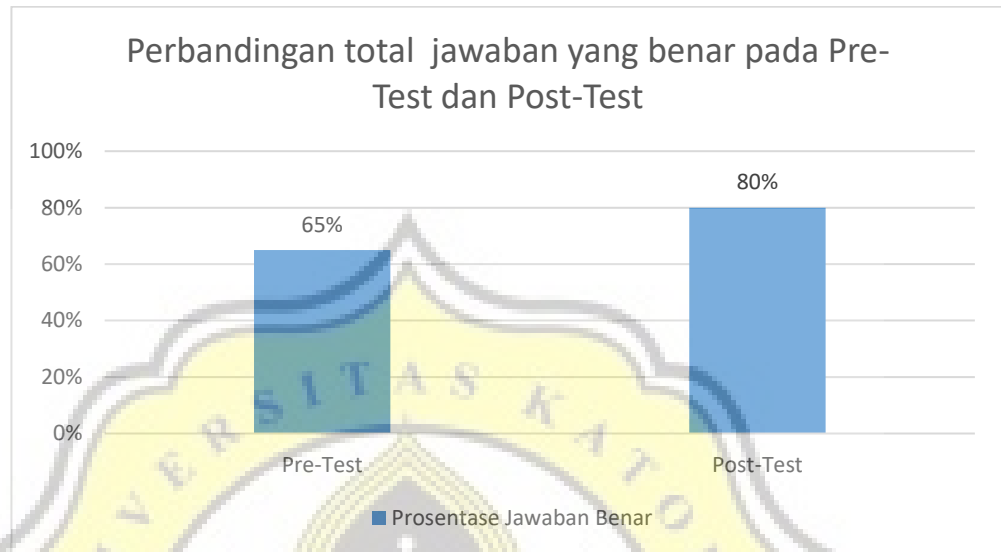
Kemajuan Teknologi telah mengubah banyak bidang ilmu pengetahuan, salah satu yang berkembang adalah teknologi *Virtual Reality*. Teknologi *Virtual Reality* memungkinkan manusia untuk membuat sebuah realitas alternatif[22], memungkinkannya untuk digunakan sebagai salah satu bentuk *game* edukasi bagi anak-anak. Menggunakan *game* sebagai media pendidikan dimana informasi secara lisa dan non-lisa dalam realita alternatif maka secara tidak sadar informasi yang terkandung dalam *game* akan di serap oleh pengguna[21].

Hasil Post-Test menunjukkan peningkatan pada pengetahuan responden terhadap rambu-rambu lalu lintas seperti gambar 4.6 ditunjukkan dengan prosentase total jawaban dari 30 responden yaitu sebesar 80% atau 360 jawaban benar yang sebelumnya sebesar 65% atau 292 jawaban seperti gambar 4.15. Hal ini menandakan formulasi materi yang dibuat mudah dan menyenangkan bagi anak-anak umur 7 sampai dengan 11 tahun.



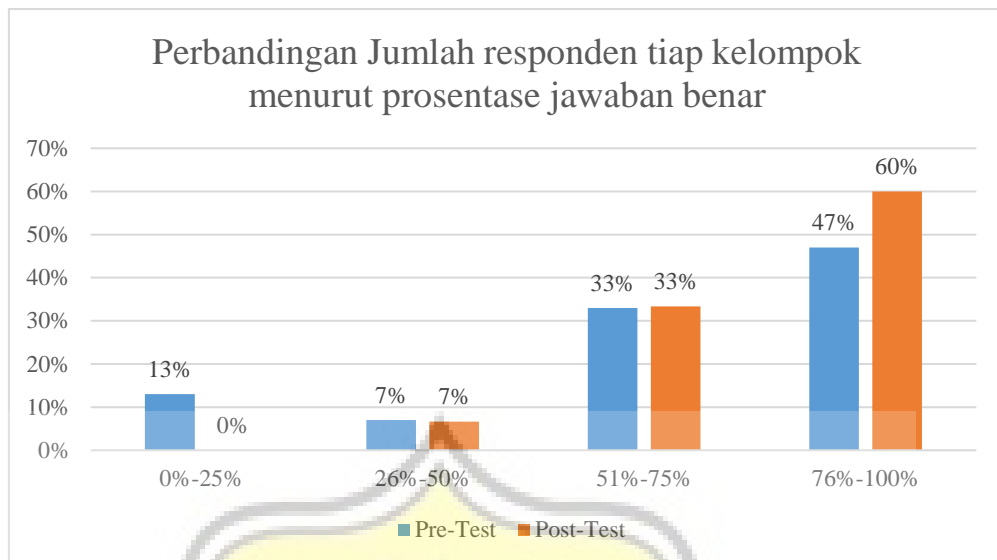
Gambar 4.14 Grafik Hasil Post-Test tentang rambu dan artinya

Perbandingan prosentase total jawaban yang benar antara Pre-test dan Post-test sebesar 15% (68 jawaban dari 450 jawaban) dari 80%-65% yang dapat dilihat pada gambar 4.16 di bawah ini.



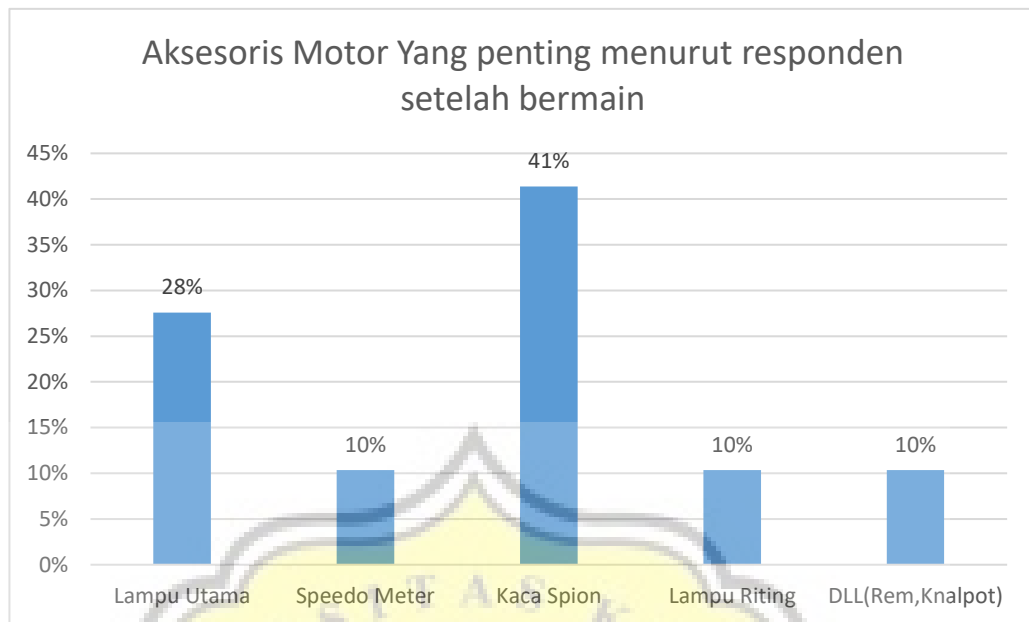
Gambar 4.15 Grafik Perbandingan Post-test dan Pre-tes

Peningkatan pengetahuan juga dapat dilihat dari peningkatan prosentasi pada kelompok responden dengan prosentase jawaban benar 76%-100% sebanyak 13% (4 responden) seperti gambar 4.17 di bawah ini.



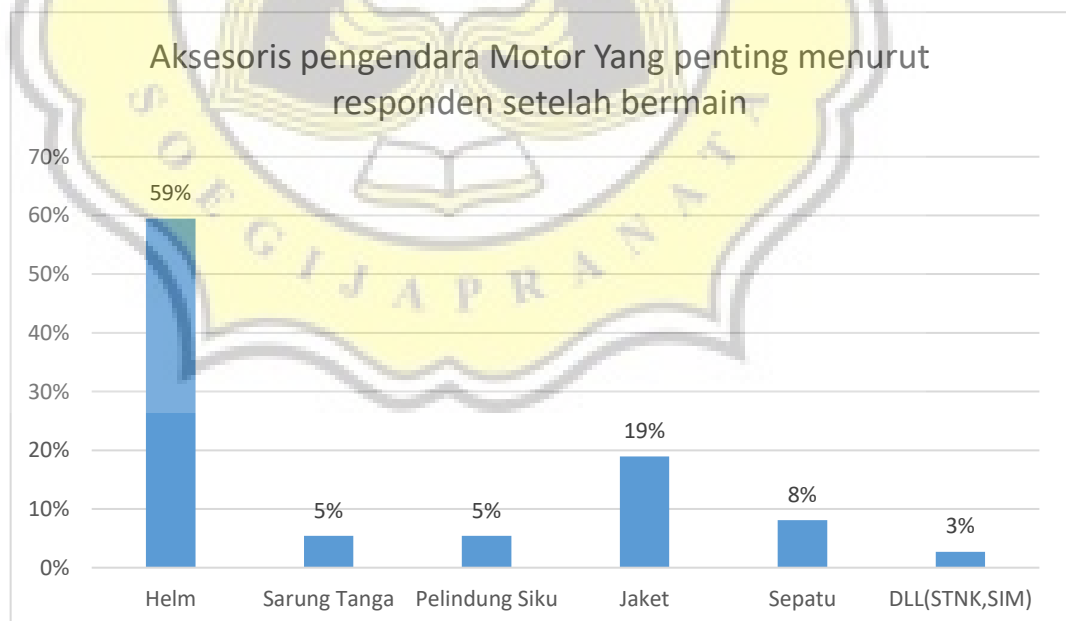
Gambar 4.16 Grafik Perbandingan jumlah responden tiap kelompok menurut prosentase jawaban benar

Pada *level 3* dimana responden diminta untuk menilang pengendara yang tidak lengkap baik aksesoris motornya maupun pengendaranya, mengajarkan responden untuk selalu menggunakan peralatan keselamatan serta memeriksa kelengkapan motornya sehingga tidak terjadi kecelakaan yang dapat menyebabkan luka serius, berikut adalah pemaparan pilihan bagian-bagian aksesoris pengendara dan motor yang dipilih oleh responden setelah bermain. Diagram di bawah menunjukkan bahwa responden memilih kaca spion lebih banyak daripada sebelumnya sebesar 42% diikuti dengan lampu utama sebesar 28% serta lampu sein, *speedometer*, dan kaca spion mengikuti dengan 10% seperti pada gambar 4.18.



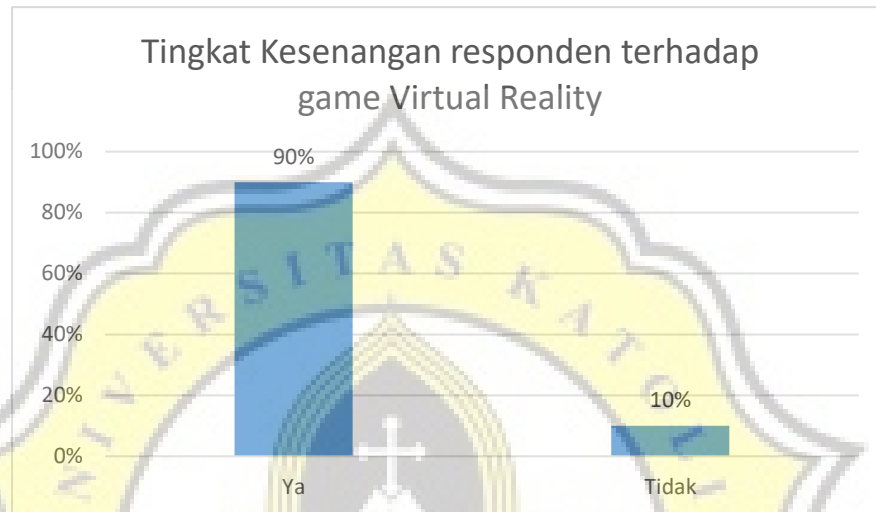
Gambar 4.17 Grafik Hasil Post-Test Kelengkapan aksesoris motor

Pada pertanyaan aksesoris motor jawaban responden mulai terbagi menjadi helm 60%, jaket 19%, sepatu 8%, sarung tangan dan pelindung siku 5%, serta 3% menjawab lain-lain seperti gambar 4.19 di bawah ini.



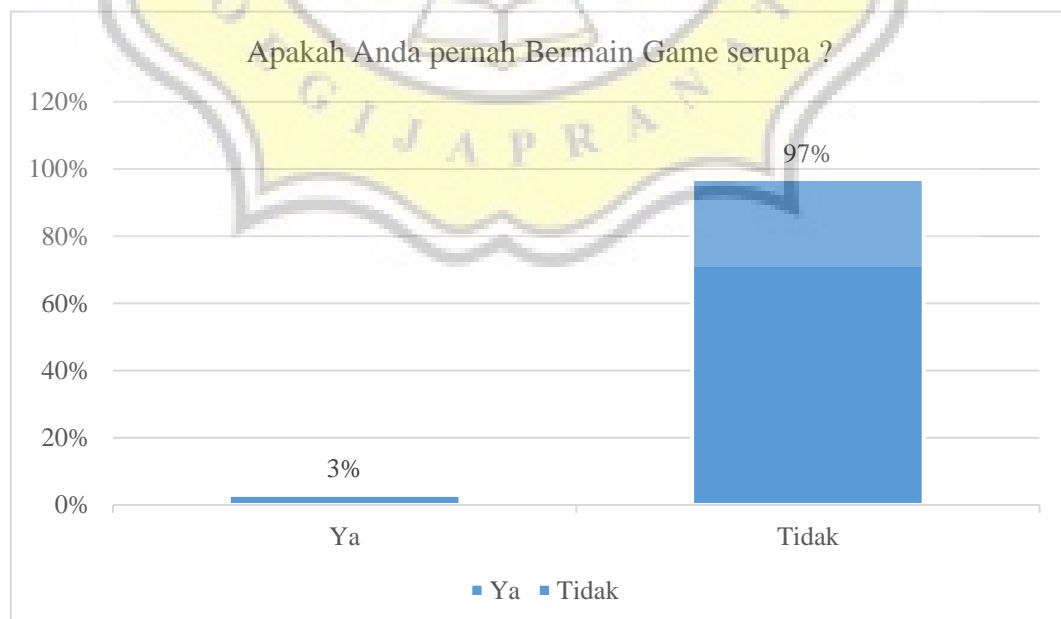
Gambar 4.18 Grafik Hasil Post-Test Kelengkapan aksesoris motor

Hasil uji menunjukkan bahwa 90% responden senang bermain *game Virtual Reality*, seperti yang ditunjukkan gambar 4.20 di bawah. Sesuai dengan rumusan masalah nomor dua, tingkat kesenangan anak-anak umur 7 sampai dengan 11 tahun sangatlah tinggi terhadap materi yang diberikan dengan desain dan spesifikasi yang cocok dan nyaman bagi mereka.



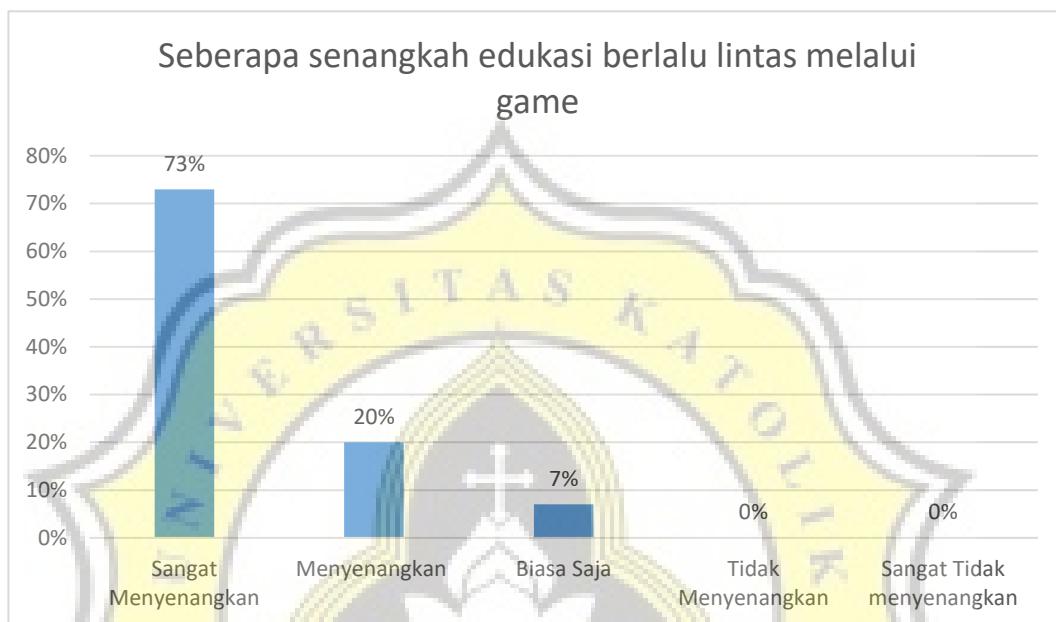
Gambar 4.19 Grafik Tingkat Kesenangan bermain *Virtual Reality*

Dari 30 responden 97% menyatakan bahwa ini kali pertama mereka bermain *game Virtual Reality* edukasi, seperti yang tersaji di gambar 4.21 di bawah ini.



Gambar 4.20 Grafik Prosentase pernah memainkan *game* serupa

Penelitian ini menunjukkan bahwa *game* berbasis *Virtual Reality* dapat memberikan edukasi dengan cara menyenangkan, gambar di bawah menyatakan 73% responden merasa sangat senang setelah bermain *game Virtual Reality* seperti yang ditampilkan pada gambar 4.22 di bawah ini.



Gambar 4.21 Grafik Tingkat Kesenangan edukasi melalui *game*

4.4 Temuan dari pengembangan *Game* berbasis *Virtual Reality*

Dalam pengembangan *game* berbasis *Virtual Reality* ini peneliti menemukan bahwa *SDK* atau *Software Development Kit* yang digunakan adalah versi 1.6 yang sebelumnya menggunakan 0.9. Perubahan versi ini membawa beberapa pengaruh seperti dihilangkannya dukungan tombol magnet yang menggunakan *magometer* sebagai metode masukan, dan diganti dengan *Capcitive button* dimana menggunakan metode sentuh pada layar yang dikemas dalam perangkat *Virtual Reality Second Generation*[27] seperti gambar 4.23 di bawah ini. Peneliti menemukan bahwa penggunaan model 3D diatas 25000 *Polygon* dapat membebani kinerja gawai secara signifikan.



Gambar 4.22 Google Cardboard Second Generation

4.5 Kekurangan dari *Game* berbasis *Virtual Reality*

Game ini masih memiliki kekurangan yang ditemui pada interval yang tidak tetap. Kerusakan aplikasi bisa terjadi secara tiba-tiba sehingga banyak responden mengatakan pusing ketika harus mengulang *level* yang sama terus menerus. Versi terbaru sudah diperbaiki sehingga meminimalkan kerusakan tersebut. Namun masih sering ditemui kerusakan pada *level 3* dimana *game* keluar secara paksa, penyebabnya tidak bisa dipastikan karena terjadi dalam interval yang acak.