

Bab IV

Perancangan, Pembuatan, dan Pengujian Game

4.1 Perancangan

4.1.1. Analisis dan Tujuan Game

Game “Indo Trivia” ini dirancang dengan tujuan supaya para pemain dapat lebih mengetahui tentang sejarah singkat beberapa Kerajaan Hindu terkenal di pulau jawa beserta beberapa peninggalan Kerajaan tersebut. Di game ini juga terdapat bermacam jenis gameplay yang dirancang supaya para pemain tidak bosan dengan gameplay yang monoton.

4.1.2. Genre, Platform, dan Target Audience

Game ini bergenre 2D platformer, Puzzle, dan Quiz. Game “Indo Trivia” ini dibuat untuk android supaya para pemain dapat memainkannya dimana saja dan kapan saja. Target pemain untuk permainan ini sekitar usia 6 tahun keatas.

4.1.3. Character & Storyline Singkat

Gambar 4.1 berikut adalah main character dalam game “Indo Trivia”



Gambar 4.1: Sprite Main Player

Character di sebelah merupakan seorang anak kecil jenius yang penasaran tentang peninggalan sejarah di Indonesia. Dia diberi tugas oleh gurunya untuk mengumpulkan relic, menatanya dan Bersama dengan gurunya, mempelajari relic tersebut.

4.1.4. Game Overview

“Indo Trivia” merupakan game untuk android dengan genre 2d platformer, puzzle dan quiz. Game ini merupakan game single player yang difokuskan untuk belajar tentang peninggalan sejarah dan sejarah Kerajaan Hindu di Indonesia. Dengan jenis permainan yang lebih dari 1 serta gambar-gambar peninggalan

4.1.5. Alur

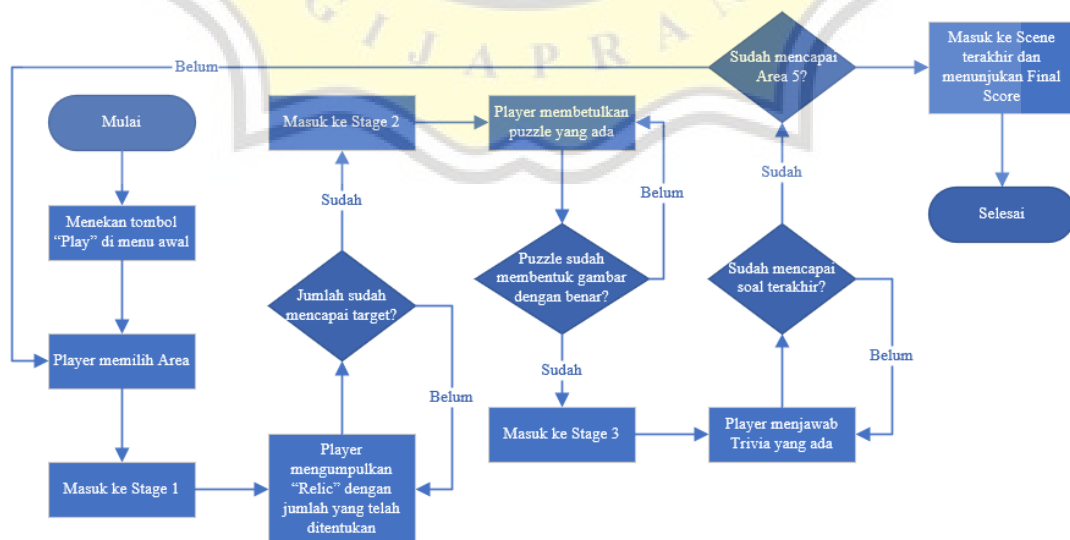
sejarah yang ada di setiap stage dan letak Kerajaan yang menyesuaikan posisi aslinya akan membuat pemain lebih tahu tentang Kerajaan Hindu di Indonesia.

Game “Indo Trivia” ini dibuat menggunakan Unity Engine 2D versi 2018.4. Game “Indo Trivia” sendiri memiliki 5 Level (Area) dan setiap level (Area) terdapat 3 stage yang harus diselesaikan.

Area 1 yaitu daerah Kerajaan Tarumanegara, Area 2 yaitu daerah Kerajaan Mataram Hindu, Area 3 yaitu daerah Kerajaan Kediri, Area 4 yaitu daerah Kerajaan Singasari, Area 5 yaitu daerah Kerajaan Majapahit. Latar belakang di setiap area akan menyesuaikan tempat yang menggambarkan/yang menjadi khas Kerajaan tersebut.

Stage 1 yaitu collecting relic / mengumpulkan relic. Di stage ini pemain harus mengumpulkan sejumlah relic yang telah ditentukan supaya dapat lanjut ke stage berikutnya. Di stage 2, yaitu arrange the puzzle/ menata puzzle, kita harus menata puzzle supaya menjadi gambar yang benar. Di stage 3, Trivia Quiz, kita akan diberi beberapa pertanyaan seputar Kerajaan di area yang kita pilih di awal dan bisa kemungkinan pertanyaan seputar Kerajaan yang berhubungan dengan Kerajaan tersebut.

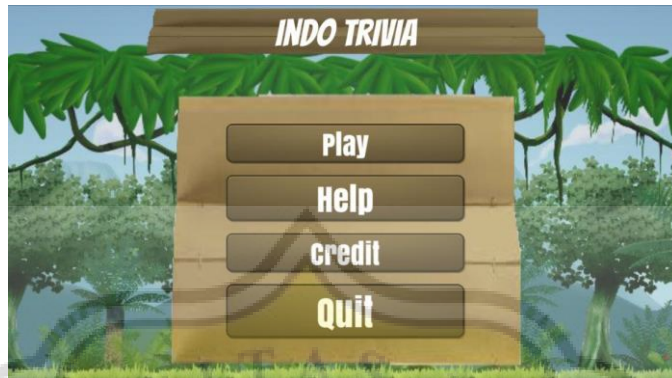
Berikut adalah Flowchart jalannya game “Indo Trivia”



Gambar 4.2 : Flowcart Jalannya Game “Indo Trivia”

4.2 Pembuatan Game

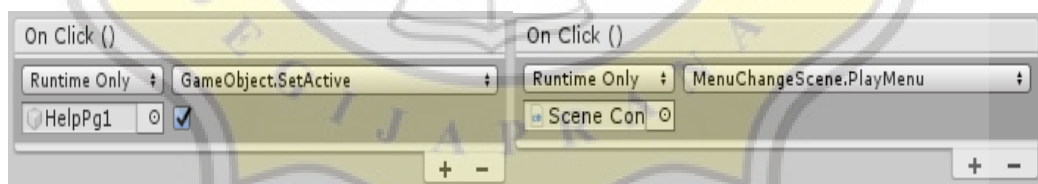
4.2.1. Menu Awal



Gambar 4.3 : Main Menu

Gambar 4.3 merupakan gambar menu utama dalam game. Untuk bermain pemain bisa menekan tombol “Play”. Untuk cara bermain dan tutorial singkat ada di bagian “Help”. Di bagian “Credit” akan ada asal dari aset atau gambar atau sound yang dipakai. Serta gambar 4.4 berikut adalah script yang digunakan serta *set-up* untuk bagian inspector.

```
public void PlayMenu()  
{  
    GameController.Level = 0;  
    AreaSelect = false;  
    AreaCh.SetActive(true);  
    Ui.SetActive(true);  
    IsSpawned = false;  
}
```



Gambar 4.4: Script untuk tombol play & *set-up* bagian inspector untuk tombol lainnya

4.2.2. Area Select



Gambar 4.5 : Area Selection

Gambar 4.5 diatas merupakan scene saat pemain memilih Area (Level). Level yang sudah selesai tidak akan bisa dimainkan kembali. Gambar 4.6 dibawah merupakan script yang digunakan untuk pindah scene dan men-set level sesuai dengan area yang dipilih ketika menekan lingkaran hijau

```

public void Lv1()
{
    if (OneCleared == false)
    {
        Destroy(PanelMenu);
        GameController.Level = 1;
        AreaCh.SetActive(false);
        AreaSelect = false;
        IsSpawned = false;
        SceneManager.LoadScene("MainZone");
    }
}
public void Lv2()
{
    if(TwoCleared == false)
    {
        Destroy(PanelMenu);
        GameController.Level = 2;
        AreaCh.SetActive(false);
        AreaSelect = false;
        IsSpawned = false;
        SceneManager.LoadScene("MainZone");
    }
}
public void Lv3()
{
    if(ThreeCleared == false)
    {
        Destroy(PanelMenu);
        GameController.Level = 3;
        AreaCh.SetActive(false);
        AreaSelect = false;
        IsSpawned = false;
        SceneManager.LoadScene("MainZone");
    }
}
}

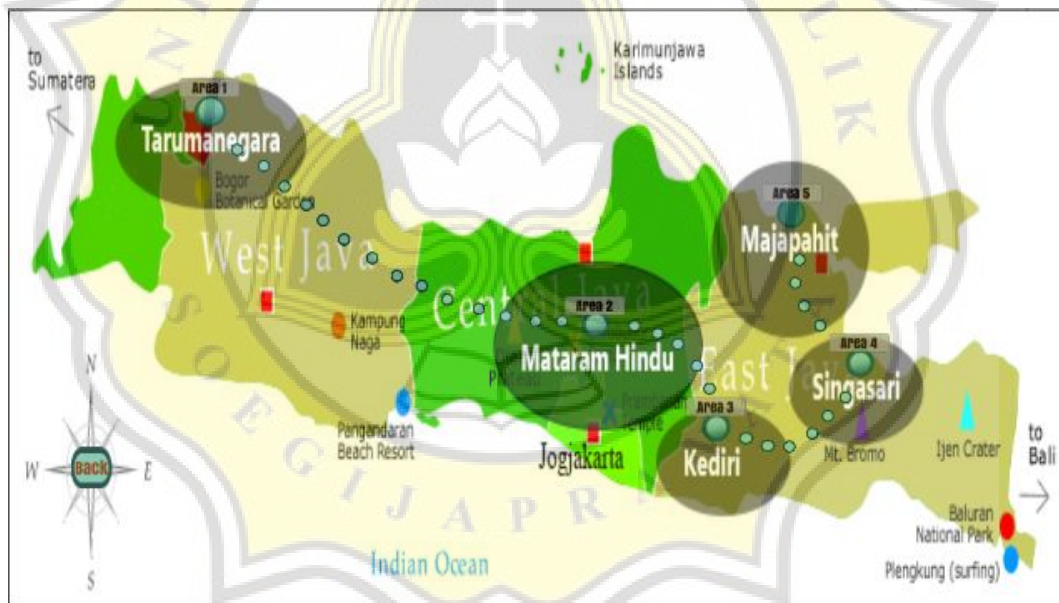
```

```

public void Lv4()
{
    if (FourCleared == false)
    {
        Destroy(PanelMenu);
        GameController.Level = 4;
        AreaCh.SetActive(false);
        AreaSelect = false;
        IsSpawned = false;
        SceneManager.LoadScene("MainZone");
    }
}
public void Lv5()
{
    if (FiveCleared == false)
    {
        Destroy(PanelMenu);
        GameController.Level = 5;
        AreaCh.SetActive(false);
        AreaSelect = false;
        IsSpawned = false;
        SceneManager.LoadScene("MainZone");
    }
}

```

Gambar 4.6: Script masuk ke Area



Gambar 4.7: tampilan layar ketika pemain telah menyelesaikan 4 area

Gambar 4.7 diatas merupakan tampilan setelah pemain menyelesaikan 4 area. Dapat dilihat, lingkaran yang terang merupakan area yang telah selesai. Saat pemain telah menyelesaikan 1 area, maka lingkaran hijau gelap untuk area tersebut akan tertutup dan menjadi terang sebagai tanda bahwa pemain telah selesai memainkannya serta area tersebut sudah tidak dapat dimainkan kembali.

```

if (Answer1.QnumInt1 == 9 && GameController.Level <= 4)
{

```

```

unanswererQuestion = null;
MenuChangeScene.IsSpawned = false;
if (GameController.Level == 1)
{
    MenuChangeScene.OneCleared = true;
}
else if (GameController.Level == 2)
{
    MenuChangeScene.TwoCleared = true;
}
else if (GameController.Level == 3)
{
    MenuChangeScene.ThreeCleared = true;
}
else if (GameController.Level == 4)
{
    MenuChangeScene.FourCleared = true;
}
else if (GameController.Level == 5)
{
    MenuChangeScene.FiveCleared = true;
}

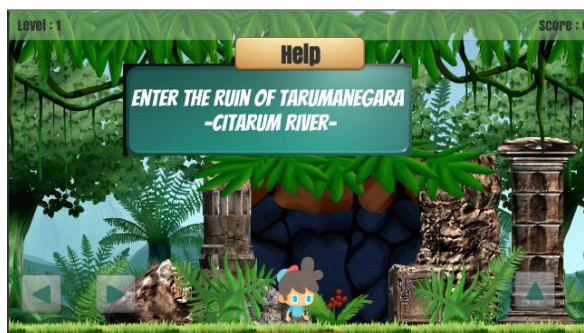
if (OneCleared == true)
{
    Cleared[0].SetActive(true);
}
if (TwoCleared == true)
{
    Cleared[1].SetActive(true);
}
if (ThreeCleared == true)
{
    Cleared[2].SetActive(true);
}
if (FourCleared == true)
{
    Cleared[3].SetActive(true);
}
if (FiveCleared == true)
{
    Cleared[4].SetActive(true);
}

```

Gambar 4.8: Script untuk mengaktifkan penutup area ketika pemain menyelesaikan stage 3

Sesuai deskripsi pada gambar 4.8, ketika pemain menyelesaikan stage 3, maka script pada gambar 4.8 akan aktif. Tapi saat pemain menyelesaikan stage 3 pada area 5 (area Majapahit), maka pemain akan langsung dibawa ke layar game over.

4.2.3. Area Pre-Stage 1



Gambar 4.9: Pre-Stage 1

Gambar 4.9 merupakan area 'Pre-Stage' dimana di tempat ini player bisa tahu nama tempat yang akan dimasuki nantinya. Di tempat ini juga pemain juga dapat melihat tutorial singkat dengan menekan tombol 'help' di bagian atas layar. Tombol 'help' ini akan hilang ketika pemain memasuki stage 1.

```

if (TouchController.onTouchRight == true)
{
    XAwal = cam.transform.position.x;
    X = X + Speed;
    camX = camX + Speed;
    Adventurer.transform.position = new Vector3(X, Y, Z);
    cam.transform.position = new Vector3(camX, camY, camZ);
    if (Adventurer.transform.position.x != cam.transform.position.x)
    {
        cam.transform.position = new Vector3(XAwal, camY, camZ);
    }
    if (!facerright)
    {
        Flip();
        facerright = true;
    }
}
else if (TouchController.onTouchLeft == true)
{
    X = X - Speed;
    camX = camX - Speed;
    Adventurer.transform.position = new Vector3(X, Y, Z);
    cam.transform.position = new Vector3(camX, camY, camZ);

    if (facerright)
    {
        Flip();
        facerright = false;
    }
}

else if (TouchController.onTouchJump == true)
{
    if (Grounded == true)
    {
        Grounded = false;
        Rb.AddForce(Vector2.up * jumpStr);
        cam.transform.position = new Vector3(camX, camY, camZ);
    }
}

```

```

        if (faceright)
        {
            Flip();
            faceright = false;
        }
    }
}

```

```

public void Left()
{
    left = true;
    onTouchLeft = true;
    if (Input.touchCount == 1)
    {
        onTouchLeft = true;
    }
}

public void Right()
{
    right = true;
    onTouchRight = true;
    if (Input.touchCount == 1)
    {
        onTouchRight = true;
    }
}

public void Jump()
{
    jump = true;
    if (Input.touchCount == 1)
    {
        onTouchJump = true;
    }
}

```

Gambar 4.10: Script untuk movement

Gambar 4.10 merupakan script yang digunakan oleh character supaya dapat bergerak ketika tombol movement ditekan (gambar 4.10 bawah). lalu mengaktifkan boolean pada tombol untuk menjalankan script gambar 4.10 bagian atas yang diletakan pada player. Serta gambar 4.11 berikut adalah script untuk men set level, skor serta objective/status. Script ini akan selalu jalan untuk menset, level yang ada pada script lainnya serta sebagai tempat penyimpanan skor.

```

void Update()
{
    DontDestroyOnLoad(this.gameObject);
    ScoreTxT.text = ("Score : "+ score + "");
    LvTxT.text = ("Level : " + Level + "");
    StatusTxT.text = Status;
}

```

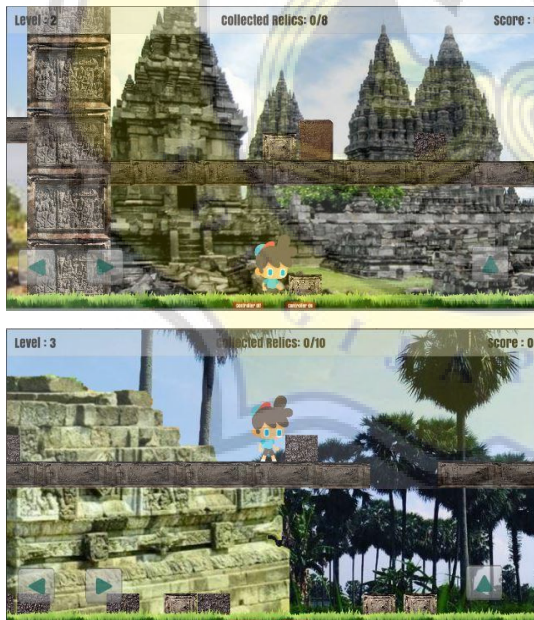
Gambar 4.11: Script untuk men-set skor, level serta status/objective (Game Controller)

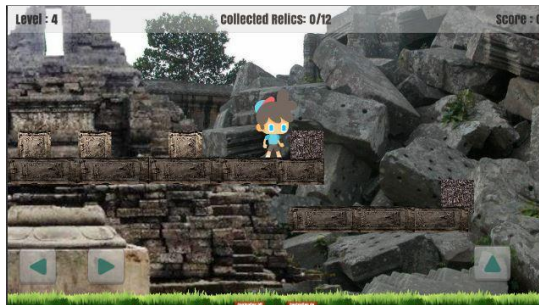
4.2.4. Stage 1



Gambar 4.12 : Stage 1 area 1

Gambar 4.12 merupakan tampilan saat player baru memasuki stage 1. Di stage 1 pemain harus mengumpulkan relic dengan jumlah yang telah ditentukan. Di Area 1 pemain mengumpulkan 6 relic, di Area 2 mengumpulkan 8 relic, di Area 3 mengumpulkan 10 relic, di Area 4 mengumpulkan 12 relic, di Area 5 mengumpulkan 15 relic untuk dapat lanjut ke stage berikutnya. Untuk setiap relic yang dikumpulkan, pemain akan mendapatkan 10 poin dan jika pemain mengumpulkan relic yang salah/buruk maka skor pemain akan dikurangi 10. Gambar 4.13 merupakan background untuk area 2, 3, 4, dan 5.





Gambar 4.13: Background di Stage 1 untuk Area 2 (kiri atas), Area 3 (kanan atas), Area 4 (bawah kiri), Area 5 (kanan bawah)

Gambar 4.14 merupakan script yang digunakan untuk memindahkan Relic saat player menghusap object serta script untuk men-spawn relic yang harus dikumpulkan serta script untuk menambah skor dan lanjut ke stage berikutnya.

```

if (GameController.Level == 1)
{
    RelicCount = 6;
}
else if (GameController.Level == 2)
{
    RelicCount = 8;
}
else if (GameController.Level == 3)
{
    RelicCount = 10;
}
else if (GameController.Level == 4)
{
    RelicCount = 12;
}
else if (GameController.Level == 5)
{
    RelicCount = 15;
}

for (int i = 0; i < RelicCount; i++)
{
    Instantiate(Relics[Random.Range(0, 1)], new Vector3(Random.Range(-70, 70),
    Random.Range(-4, 1.5f), -4), Quaternion.Euler(roatation));
    Instantiate(Rocks[Random.Range(0, 3)], new Vector3(Random.Range(-70, 70),
    Random.Range(-4, 1.5f), -4), Quaternion.Euler(roatation));
}
for (int i = 0; i < RelicCount - 7; i++)
{

```

```

if (RelicCount >= 7)
{
    Instantiate(Rocks[Random.Range(0, 3)], new Vector3(Random.Range(-70, 70),
Random.Range(-4, 1.5f), -4), Quaternion.Euler(rootation));
}
}

```

Gambar 4.14: Script untuk men-spawn Relic.

Script pada gambar 4.14 akan langsung jalan saat pertama kali masuk ke stage 1. Semakin besar levelnya, maka semakin banyak pula relic bagus dan relic jelek yang spawn.

```

if (PlayerNerby == true)
{
    if (Input.touchCount > 0)
    {
        Touch touch = Input.GetTouch(0);
        Vector2 touchPos = Camera.main.ScreenToWorldPoint(touch.position);
        if (touch.phase == TouchPhase.Began)
        {
            fp = touch.position;
            lp = touch.position;
            Collider2D touchedCollider = Physics2D.OverlapPoint(touchPos);
            if (col == touchedCollider)
            {
                }
            moveAllowed = true;
        }
        if (touch.phase == TouchPhase.Moved)
        {
            lp = touch.position;
            if (moveAllowed == true)
            {
                transform.position = new Vector2(touchPos.x, touchPos.y);

                if (Mathf.Abs(lp.x - fp.x) != DragDistance || Mathf.Abs(lp.y - fp.y) != DragDistance)
                {
                    this.gameObject.transform.position = new Vector3(Random.Range(-4.5f, 2), -2, 0);
                    ColletionItem.Collected = true;
                }
            }
        }
        if (touch.phase == TouchPhase.Ended)
        {
            moveAllowed = false;
        }
    }
}

```

Gambar 4.15: Script untuk memindahkan relic

Script pada gambar 4.15 digunakan untuk memindahkan relic ketika relic mendeteksi player didekatnya. Lalu script pada gambar 4.16 adalah script untuk menambah skor untuk setiap relic yang terkumpul dan saat mencapai target, maka akan langsung lanjut ke stage berikutnya.

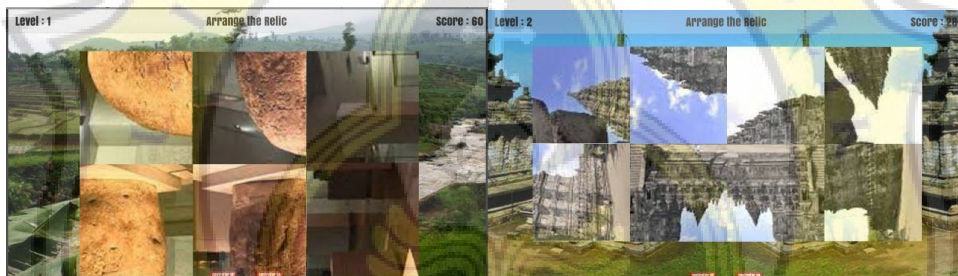
```

private void OnTriggerEnter2D(Collider2D collision)
{
    if (CollectionItem.Collecte == true)
    {
        if (collision.gameObject.tag.Equals("RelicItem"))
        {
            GameController.score = GameController.score + 10;
            CollectedRelic = CollectedRelic + 1;
            Debug.Log("collected Relic : " + CollectedRelic + "/" + RelicCount);
            if(CollectedRelic == RelicCount)
            {
                CollectedRelic = 0;
                GameController.Status = "Arrange the Relics";
                SceneManager.LoadScene("PuzzleZone");
            }
        }
        else if (collision.gameObject.tag.Equals("Rocks"))
        {
            GameController.score = GameController.score - 10;
        }
    }
}

```

Gambar 4.16 script untuk lanjut ke stage berikutnya ketika relic telah terkumpul

4.2.5. Stage 2



Gambar 4.17: Puzzle untuk Stage 2 Area 1 (kiri), Area 2 (kanan)

Di stage 2 pemain akan diberikan sejumlah gambar dengan posisi yang berantakan. pemain harus menekan gambar tersebut untuk memutar gambar 90°. saat gambar sudah tidak berantakan maka akan keluar nama dari gambar tersebut lalu pemain dapat lanjut ke stage 3. Di stage 2 ini untuk setiap potongan gambar, pemain akan diberikan 10 poin. Pada gambar 4.17 merupakan puzzle untuk area 1 dan 2, dan pada gambar 4.18 merupakan gambar puzzle untuk area 3, 4, dan 5



Gambar 4.18: Puzzle untuk Stage 2 Area 3 (kiri), Area 4 (kanan), Area 5 (bawah)

Gambar 4.19 merupakan script untuk memutar Puzzle. Script ini akan aktif ketika gambar disentuh oleh pemain. Saat gambar disentuh maka potongan puzzle akan diputar 90°. ketika semua gambar telah diposisikan dengan baik, maka nama gambar dan tombol untuk lanjut ke stage berikutnya akan dimunculkan.

```

rotateNum = rotateNum + 90;
if (rotateNum >= 360)
{
    rotateNum = 0;
}
this.gameObject.transform.rotation = Quaternion.Euler(0, 0, rotateNum);
GameController.Status = "Rotation";

```

Gambar 4.19: Script untuk memutar potongan puzzle

4.2.6. Stage 3



Gambar 4.20: Gameplay Stage 3

Gambar 4.20 merupakan tampilan saat player memasuki stage 3. Di stage 3, pemain akan diberikan soal yang muncul di bagian atas layar, jawaban akan keluar dari sisi kanan. Pemain dapat menekan layar supaya player bisa naik lalu jika layar tidak ditekan maka player akan turun dengan sendirinya. Untuk setiap jawaban yang benar, pemain akan diberi 10 poin dan untuk setiap jawaban yang salah, maka pemain tidak akan mendapatkan poin tambahan. Di stage ini akan terdapat 8 buah soal yang akan muncul.

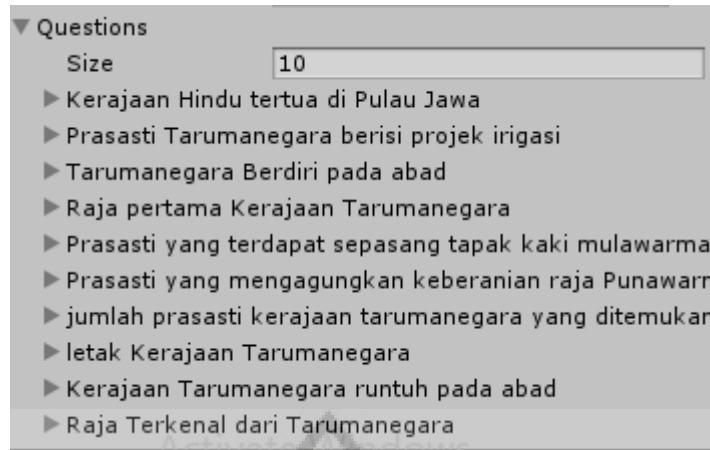
```
void Start()
{
    if (unanswererQuestion == null || unanswererQuestion.Count == 0)
    {
        unanswererQuestion = questions.ToList<QuestionController>();
    }
    GetRandomQuestion();

    IsSpawned = false;
    Answered = false;
    PosYRnd = Random.Range(1, 3);
    AnsSpawn();
}

void Update()
{
    if (!IsSpawned)
    {
        Start();
    }
}

void GetRandomQuestion()
{
    int randomQuestionIndex = Random.Range(0, unanswererQuestion.Count);
    currentQuestion = unanswererQuestion[randomQuestionIndex];
    Qtxt.text = currentQuestion.Question + "";
    unanswererQuestion.RemoveAt(randomQuestionIndex);
}
```

Gambar 4.21: Script untuk menapatkan soal random



Gambar 4.22: list soal yang akan diambil untuk pertanyaan

2 gambar diatas (gambar 4.21 dan 4.22) merupakan cara untuk men-set soal secara random. Gambar 4.21 merupakan script yang digunakan untuk mendapatkan soal dari 'List Soal' (gambar 4.22). 'list soal' sendiri termasuk dalam variable '*Question*'. Soal yang sudah keluar, tidak akan keluar lagi. Gambar 4.23 merupakan script yang digunakan untuk men-spawn jawaban

```
void AnsSpawn()
{
    if (!IsSpawned)
    {
        if (PosYRnd == 1 )
        {
            Instantiate(AnsA, new Vector2(PosX, PosA), Quaternion.identity);
            Instantiate(AnsB, new Vector2(PosX, PosB), Quaternion.identity);
            Instantiate(AnsC, new Vector2(PosX, PosC), Quaternion.identity);
            IsSpawned = true;
            AnsYA = PosA;
            AnsYB = PosB;
            AnsYC = PosC;
        }
        else if (PosYRnd == 2 )
        {
            Instantiate(AnsC, new Vector2(PosX, PosA), Quaternion.identity);
            Instantiate(AnsA, new Vector2(PosX, PosB), Quaternion.identity);
            Instantiate(AnsB, new Vector2(PosX, PosC), Quaternion.identity);
            IsSpawned = true;
            AnsYA = PosB;
            AnsYB = PosC;
            AnsYC = PosA;
        }
        else if (PosYRnd == 3 )
        {
            Instantiate(AnsB, new Vector2(PosX, PosA), Quaternion.identity);
            Instantiate(AnsC, new Vector2(PosX, PosB), Quaternion.identity);
            Instantiate(AnsA, new Vector2(PosX, PosC), Quaternion.identity);
            IsSpawned = true;
            AnsYA = PosC;
            AnsYB = PosA;
            AnsYC = PosB;
        }
    }
}
```

```

    }
  }
}

```

Gambar 4.23: Script untuk men-Spawn jawaban

saat jawaban telah spawn, maka jawaban akan bergerak dari kanan layar ke kiri layar. Saat jawaban menabrak player, maka jawaban akan berhenti untuk memberi tahu mana jawaban salah dan benar. saat layar disentuh lagi, maka soal dan jawaban akan digantikan dengan yang baru. jika jawaban yang benar yang ditabrak maka poin pada layar bagian pojok kanan atas akan ditambah 1. Saat ke-8 jawaban telah terjawab, maka jumlah jawaban yang benar akan menambah skor sejumlah 10 setiap jawaban yang benar. Gambar 4.24 ini adalah script untuk melakukan hal diatas.

```

void Update()
{
    X = X - BGScroll.speed;
    AnswerA.transform.position = new Vector3(X, Y, Z);
    Answer01.text = Questions_Change.currentQuestion.answer + "";
    if (Questions_Change.IsSpawned == true)
    {
        if (Questions_Change.Answered == true)
        {
            foreach (GameObject Choice in GameObject.FindGameObjectsWithTag("Answer"))
            {
                Player_Move.isActive = true;
                Destroy(Choice);
                CorrectOn = false;
                Questions_Change.IsSpawned = false;
                Questions_Change.Answered = false;
                BGScroll.speed = 0.1f;
            }
        }
    }

    if (CorrectOn == true)
    {
        Checkmark.SetActive(true);
    }

    else if (CorrectOn == false)
    {
        Checkmark.SetActive(false);
    }
}

```

Gambar 4.24: script untuk menggerakkan jawaban

Jawaban akan bergerak secepat jalannya background.

```

private void OnTriggerEnter2D(Collider2D AnsColl)
{
    if (AnsColl.gameObject.tag.Equals("Player"))
    {

```



```

FindObjectOfType<AudioManager>().Play("Picked");
CorrectOn = true;
Player_Move.isActive = false;
QnumInt1 = QnumInt1 + 1;
GameController.score = GameController.score + 10;
CorrectAns = CorrectAns + 1;
BGScroll.speed = 0;

}
}

```

Gambar 4.25: script ketika jawaban menabrak player

Gambar 4.25 merupakan script saat jawaban menabrak player. Script untuk jawaban benar dan salah hampir sama yang membedakan adalah bagian yang diberi Bold/ditebali. Bagian yang ditebali tidak ada pada script untuk jawaban yang salah karena bagian tersebut merupakan bagian untuk menambah nilai dan poin pada jumlah jawaban benar. Untuk bagian yang ditebalkan dan dimiringkan adalah bagian untuk memutar sound. Untuk bagian ini tetap ada tetapi sound yang diputar beda.



Gambar 4.25: Panel untuk sejarah singkat

Jika semua telah terjawab maka akan muncul panel sejarah singkat seperti pada gambar 4.25. Panel ini terdapat sejarah singkat dari Kerajaan pada area tersebut. Dari panel ini juga kita dapat tahu berbagai macam peninggalan sejarah dari Kerajaan tersebut. Script pada gambar 4.26 akan aktif ketika soal ke 8 selesai

```

if (Historys == true)
{
    HistoryPanel.SetActive(true);
    Historys = false;
    Answer1.CorrectAns = 0;
    Answer1.QnumInt1 = 0;
}

```

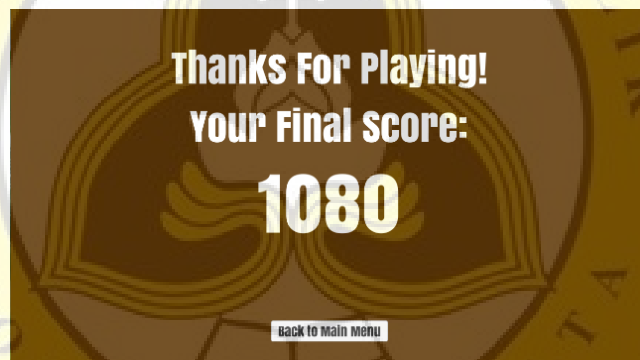
Gambar 4.26: script untuk mengaktifkan panel sejarah singkat (gambar 4.25)

Setelah itu pemain akan kembali ke “Area Selection” dengan lingkaran hijau yang telah dipilih tadi akan tertutup. Saat ke 5 area selesai, maka player akan dibawa ke screen “Game Over” dan akan diperlihatkan skornya. Gambar 4.27 merupakan script untuk mengaktifkan panel game over.

```
else if (Answer1.QnumInt1 == 9 && GameController.Level == 5)
{
    unanswererQuestion = null;
    MenuChangeScene.Finale = true;
    if (Historys == true)
    {
        HistoryPanel.SetActive(true);
        Historys = false;
        Answer1.CorrectAns = 0;
        Answer1.QnumInt1 = 0;
    }
    Destroy(GameObject.FindGameObjectWithTag("canvas"));
}
```

Gambar 4.27: script untuk mengaktifkan panel Game Over

4.2.7. Game Over



Gambar 4.28: Layar Game Over

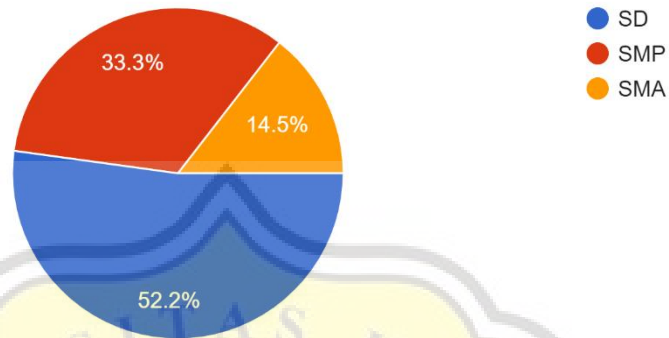
Layar terakhir ini (seperti pada gambar 4.28) akan muncul saat pemain telah menyelesaikan Area 5 stage 3. Script pada gambar 4.27 akan aktif ketika soal ke-8 pada Area 5 stage 3 selesai, setelah itu panel sejarah singkat akan diaktifkan. Lalu saat pemain selesai membaca sejarah singkat, maka pemain baru akan dibawa ke layar game over. Di layar ini pemain akan dapat melihat poin yang dikumpulkan. Semakin besar poin berarti semakin tahu anda tentang sejarah Kerajaan hindu beserta beberapa peninggalannya.

4.3 Pengujian Game

4.3.1 Profil Responden dan Data Kuesioner

1. Jenjang Pendidikan Responden

Jenjang pendidikan anda sekarang
69 responses



Gambar 4.29: Diagram Jenjang Pendidikan

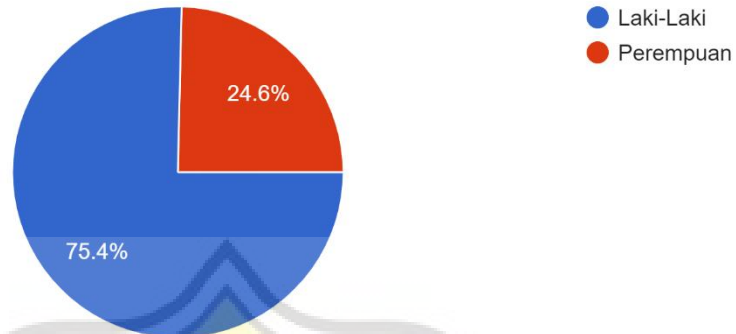
Diagram diatas(gambar 4.29) menunjukkan jenjang Pendidikan responden yang mencoba game “Indo Trivia” dan mengisi Form untuk responden. jumlah responden ada 69 orang dimana 36 responden atau 52.2% dari seluruh responden yang mencoba merupakan anak SD, 33.3% atau 23 dari 69 responden merupakan anak SMP, dan 14.5% atau 10 dari 69 responden merupakan anak SMA

2. Jenis Kelamin Responden

Diagram dibawah ini (gambar 4.30) menunjukkan jenis kelamin dari responden. sebanyak 75.4% atau 52 dari 69 responden merupakan anak laki laki, sedangkan 24.6% atau 17 dari 69 responden merupakan anak perempuan

Apa jenis kelamin anda?

69 responses



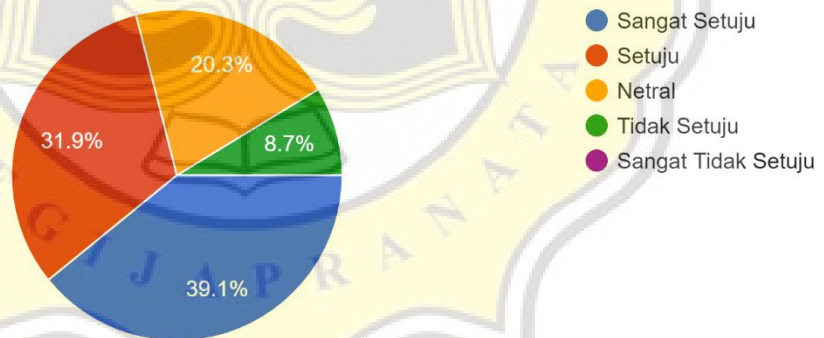
Gambar 4.30: Diagram Jenis Kelamin

3. Pembahasan Data Kuesioner Terhadap Game “Indo Trivia”

Data kuesioner yang didapatkan dari responden game “Indo Trivia” dapat dilihat di pada gambar 4.31 sampai 4.42:

1. Mempelajari cara bermain game “Indo Trivia” mudah bagi saya

69 responses

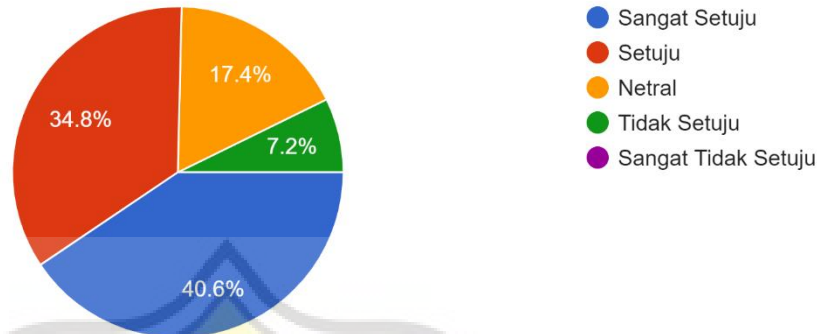


Gambar 4.31: Diagram EE1

Dari diagram diatas (Gambar 4.31), memperlihatkan sebanyak 39.1% menjawab Sangat setuju, 31.9% menjawab Setuju, 20.3% menjawab Netral, dan 8.7% menjawab Tidak setuju dengan pernyataan “mempelajari cara bermain game “Indo Trivia” mudah bagi saya”

2. Saya merasa game "Indo Trivia" mudah digunakan

69 responses

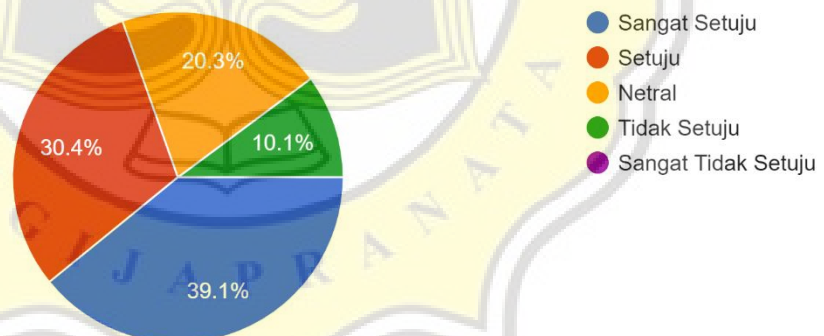


Gambar 4.32: Diagram EE2

Dari diagram diatas (Gambar 4.32), memperlihatkan sebanyak 40.6% menjawab Sangat setuju, 34.8% menjawab Setuju, 17.4% menjawab Netral, dan 7.2% menjawab Tidak setuju dengan pernyataan "saya merasa game "Indo Trivia" mudah digunakan"

3. Mudah bagi saya untuk mahir dalam memainkan game "Indo Trivia"

69 responses

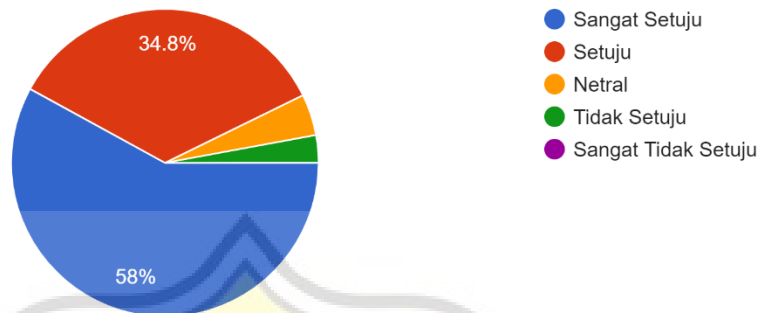


Gambar 4.33: Diagram EE3

Dari diagram diatas (Gambar 4.33), memperlihatkan sebanyak 39.1% menjawab Sangat setuju, 30.4% menjawab Setuju, 20.3% menjawab Netral, dan 10.1% menjawab Tidak setuju dengan pernyataan "Mudah bagi saya untuk mahir dalam memainkan game "Indo Trivia""

4. Saya merasa game "Indo Trivia" berguna untuk menambah wawasan saya tentang Sejarah Kerajaan Hindu

69 responses

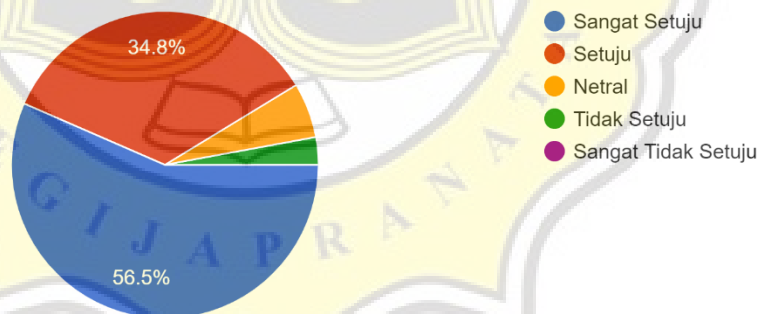


Gambar 4.34: Diagram PE1

Dari diagram diatas (Gambar 4.34), memperlihatkan sebanyak 58% menjawab Sangat setuju, 34.8% menjawab Setuju, 4.3% menjawab Netral, dan 2.9% menjawab Tidak setuju dengan pernyataan "Saya merasa game "Indo Trivia" berguna untuk menambah wawasan saya tentang sejarah Kerajaan Hindu"

5. Saya merasa game "Indo Trivia" membantu saya dalam mempelajari sejarah Kerajaan Hindu di Indonesia

69 responses

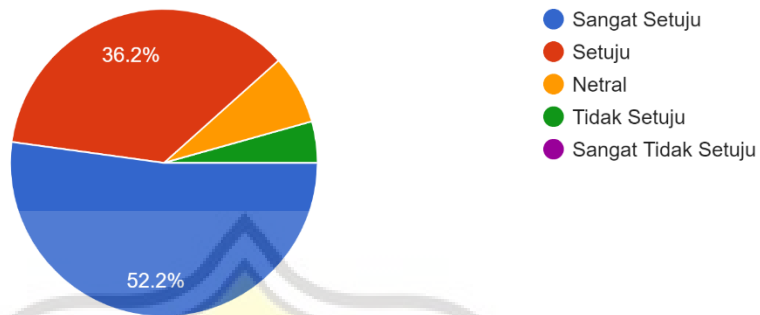


Gambar 4.35: Diagram PE2

Dari diagram diatas (Gambar 4.35), memperlihatkan sebanyak 56.5% menjawab Sangat setuju, 34.8% menjawab Setuju, 5.8% menjawab Netral, dan 2.9% menjawab Tidak setuju dengan pernyataan "Saya merasa game "Indo Trivia" membantu saya dalam mempelajari sejarah Kerajaan Hindu di Indonesia"

6. Saya merasa memainkan game "Indo Trivia" membantu meningkatkan pengetahuan saya tentang sejarah Kerajaan Hindu di Indonesia

69 responses

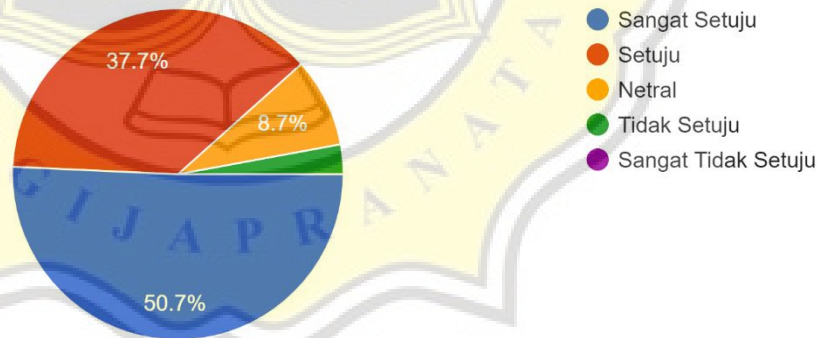


Gambar 4.36: Diagram PE3

Dari diagram diatas (Gambar 4.36), memperlihatkan sebanyak 52.2% menjawab Sangat setuju, 36.2% menjawab Setuju, 7.2% menjawab Netral, dan 4.3% menjawab Tidak setuju dengan pernyataan “Saya merasa memainkan game “Indo Trivia” membantu meningkatkan pengetahuan saya tentang Kerajaan Hindu di Indonesia”

7. Saya merasa bermain game "Indo Trivia" itu menyenangkan

69 responses

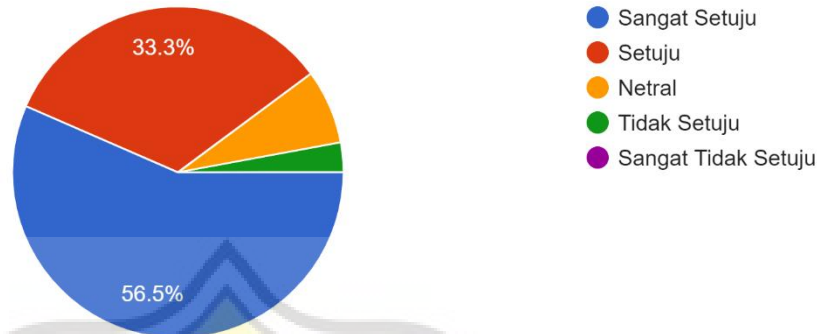


Gambar 4.37: Diagram HM1

Dari diagram diatas (Gambar 4.37), memperlihatkan sebanyak 50.7% menjawab Sangat setuju, 37.7% menjawab Setuju, 8.7% menjawab Netral, dan 2.9% menjawab Tidak setuju dengan pernyataan “saya bermain game “Indo Trivia” itu menyenangkan”

8. Saya merasa bermain game "Indo Trivia" itu menghibur

69 responses

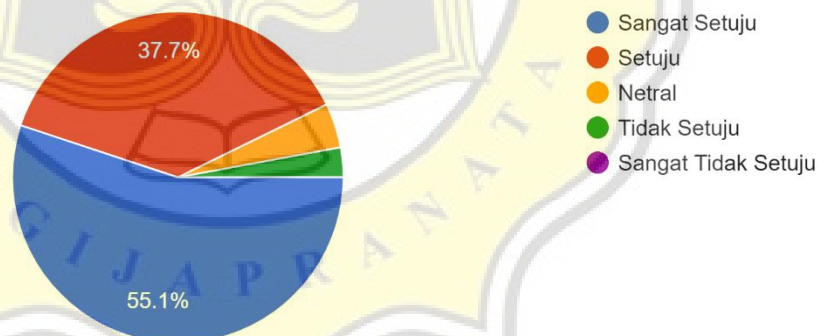


Gambar 4.38: Diagram HM2

Dari diagram diatas (Gambar 4.38), memperlihatkan sebanyak 56.5% menjawab Sangat setuju, 33.3% menjawab Setuju, 7.2% menjawab Netral, dan 2.9% menjawab Tidak setuju dengan pernyataan “Saya merasa bermain game “Indo Trivia” itu menghibur”

9. Saya merasa bermain game "Indo Trivia" itu mengasyikan

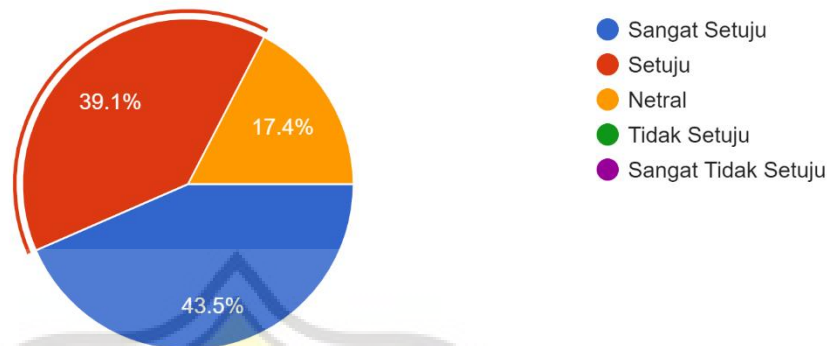
69 responses



Gambar 4.39: Diagram HM3

Dari diagram diatas (Gambar 4.39), memperlihatkan sebanyak 55.1% menjawab Sangat setuju, 37.7% menjawab Setuju, 4.3% menjawab Netral, dan 2.9% menjawab Tidak setuju dengan pernyataan “Saya merasa bermain game “Indo Trivia” itu mengasyikan”

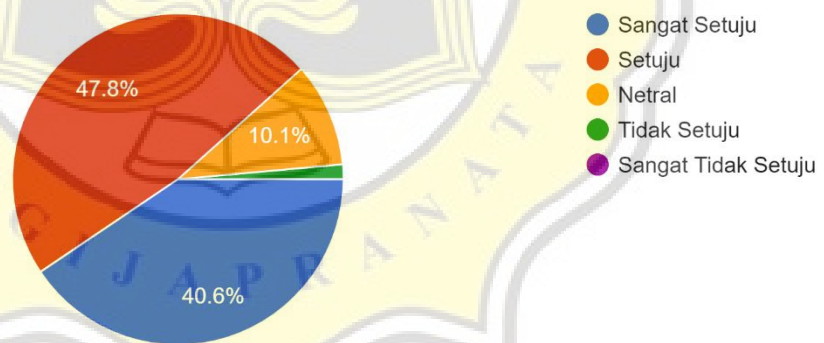
10. saya berencana untuk bermain game "Indo Trivia" dimasa yang akan datang
69 responses



Gambar 4.40: Diagram BI1

Dari diagram diatas (Gambar 4.40), memperlihatkan sebanyak 43.5% menjawab Sangat setuju, 39.1% menjawab Setuju, dan 17.4% menjawab Netral dengan pernyataan “Saya berencana bermain game “Indo Trivia” dimasa yang akan datang”

11. saya ingin terus mencoba bermain game "Indo Trivia" dalam kehidupan sehari hari
69 responses

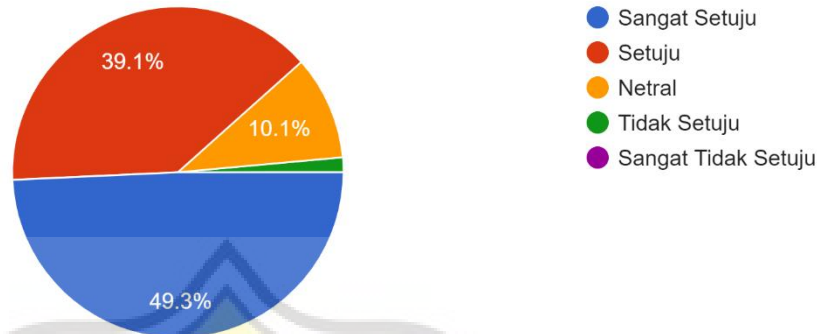


Gambar 4.41: Diagram BI2

Dari diagram diatas (Gambar 4.41), memperlihatkan sebanyak 40.6% menjawab Sangat setuju, 47.8% menjawab Setuju, 10.1% menjawab Netral, dan 1.4% menjawab Tidak setuju dengan pernyataan “Saya ingin terus mencoba bermain game “Indo Trivia” dalam kehidupan sehari hari”

12. saya ingin menyarankan game "Indo Trivia" kepada saya orang lain

69 responses



Gambar 4.42: Diagram BI3

Dari diagram diatas (Gambar 4.42), memperlihatkan sebanyak 49.3% menjawab Sangat setuju, 39.1% menjawab Setuju, 10.1% menjawab Netral, dan 1.4% menjawab Tidak setuju dengan pernyataan "Saya ingin menyarankan game "Indo Trivia" kepada orang lain"

4.3.2 Pengujian dengan SPSS

1. Uji Validitas

Rotated Component Matrix^a

	Component		
	1	2	3
EE1	-.017	-.062	.936
EE2	-.032	-.075	.926
EE3	.321	.104	.794
PE1	.092	.907	.055
PE2	.211	.893	-.068
PE3	.209	.864	-.073
HM1	.583	.564	.129
HM2	.735	.423	-.115
HM3	.750	.452	-.032
BI1	.849	.160	.142
BI2	.816	.101	.250
BI3	.808	.112	.049

Tabel 1: Hasil Uji Validitas

Tabel 1 diatas menunjukkan hasil dari uji validitas untuk mengetahui apakah kuesioner tersebut sudah valid. Semua variabel yang memiliki nilai diatas 0.5 maka

variable tersebut sudah valid. Dari data diatas kita dapat mengetahui Variabel EE memiliki nilai yang tinggi di kolom component 3 yaitu 0.936, 0.926 dan 0.794. Variabel PE memiliki nilai yang tinggi di kolom component 2 yaitu 0.907, 0.893 dan 0.864. Variabel HM memiliki nilai yang tinggi di kolom component 1 yaitu 0.583, 0.735 dan 0.750. serta Variabel BI memiliki nilai yang tinggi di kolom component 1 juga yaitu 0.849, 0.816, dan 0.808.

2. Uji Reliabilitas

Cronbach's Alpha	Internal Consistency
$\alpha \geq 0.9$	Excellent
$0.9 > \alpha \geq 0.8$	Good
$0.8 > \alpha \geq 0.7$	Acceptable
$0.7 > \alpha \geq 0.6$	Questionable
$0.6 > \alpha \geq 0.5$	Poor
$0.5 > \alpha$	Unacceptable

Tabel 2: Rentang Nilai Uji Reliabilitas

Table 2 merupakan table untuk mengetahui apakah nilai dari variabel yang ada dapat dipertanggungjawabkan atau tidak. Tabel 3 dibawah ini menunjukkan hasil dari uji Reliabilitas.

Dari tabel 3 dibawah kita dapat mengetahui semua variabel memiliki nilai diatas 0.8 maka kuesioner dapat dipertanggungjawabkan. Variabel EE memiliki Cronbach's Alpha 0.871 dan Variabel BI memiliki Cronbach's Alpha 0.874 maka Internal Consistency-nya Good, serta Variabel PE memiliki Cronbach's Alpha 0.907 dan Variabel HM memiliki Cronbach's Alpha 0.904 maka Internal Consistency-nua Excellent.

Variabel	Cronbach's Alpha	Internal Consistency
EE	0.871	Good
PE	0.907	Excellent
HM	0.904	Excellent
BI	0.874	Good

Tabel 3: Hasil Uji Reliabilitas

3. Uji Korelasi

	EE	PE	HM	BI
EE	1	-.029	.087	.240*
PE	-.029	1	.575**	.357**
HM	.087	.575**	1	.641**
BI	.240*	.357**	.641**	1

* Correlation is significant at the 0.05 level (2-tailed).

** Correlation is significant at the 0.01 level (2-tailed).

Tabel 4: Hasil Uji Korelasi

Dari table 4 diatas, kita dapat menyimpulkan bahwa nilai yang memiliki * merupakan nilai yang memiliki korelasi. Untuk nilai yang memiliki * korelasi signifikan pada level 0.05 atau 5%, sedangkan nilai yang memiliki ** korelasi signifikan pada level 0.01 atau 1%. Nilai di table 4 diatas merupakan nilai dari pearson correlation. Lalu untuk nilai dari r tabel yang digunakan adalah 0.2335 untuk level 5% dan 0.3035 untuk level 1%, jika variabel memiliki nilai lebih besar dari nilai r tabel, maka terdapat korelasi. Terlihat pada variabel EE dengan BI memiliki nilai 0.240 dimana lebih besar dari 0.2335 jadi terdapat korelasi antara EE dan BI. Untuk variabel PE ke BI memiliki nilai 0.357 dimana lebih besar dari 0.3035 jadi terdapat korelasi antara PE dengan BI. Untuk variabel HM dengan BI memiliki nilai 0.641 dimana lebih besar dari 0.3035 jadi terdapat korelasi antara HM dengan BI.

