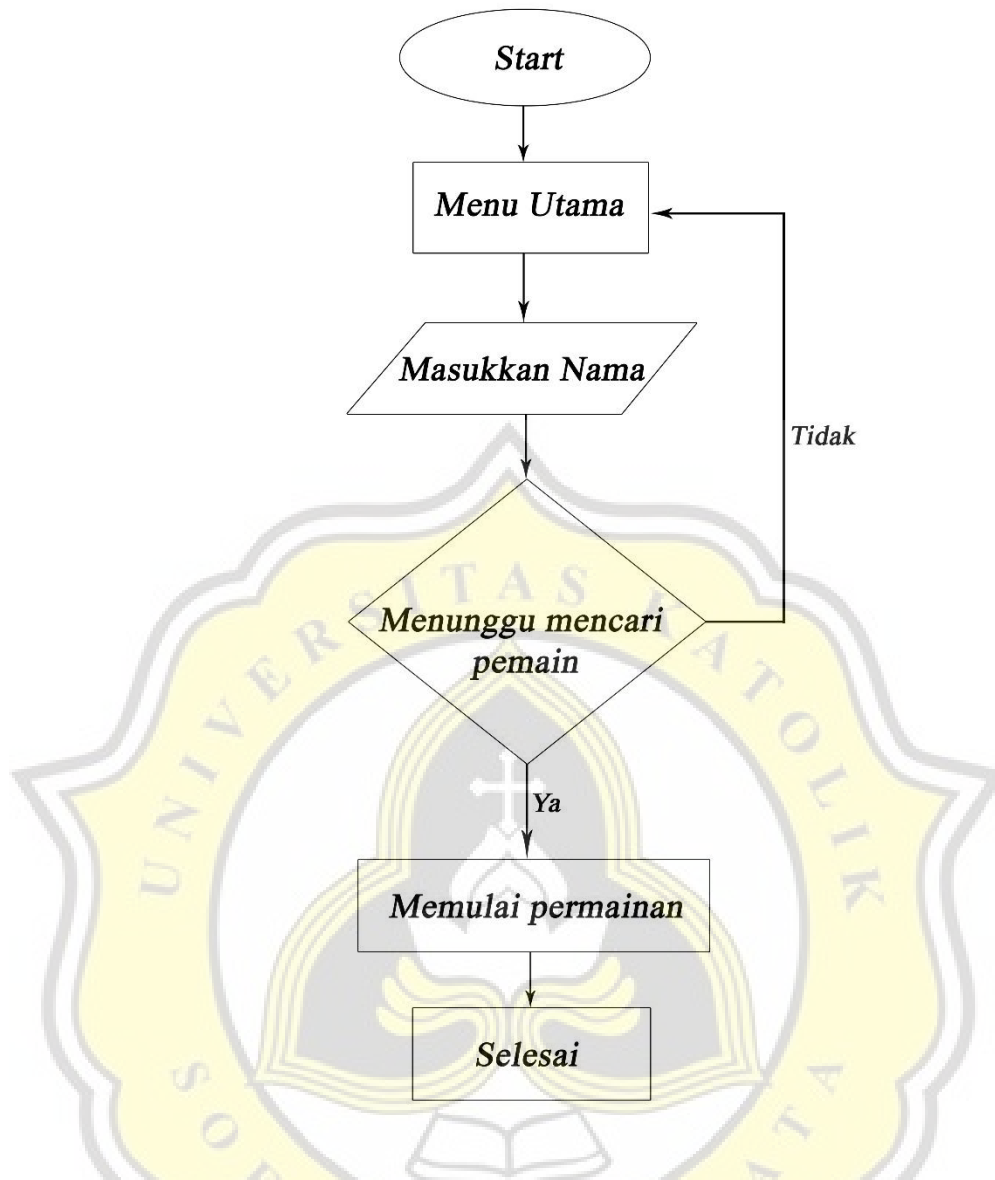


BAB IV

PEMBAHASAN

4.1 Perancangan Game

Dalam perancangan *game*, dibutuhkan beberapa analisa yang bertujuan untuk mengetahui *game* kuis apa yang sedang banyak dimainkan oleh para pengguna *smartphone*. Analisa dapat dilakukan dengan cara mencari sumber atau referensi dari internet, dari informasi tersebut dapat dibuat rancangan *game* seperti tampilan dalam *game*, soal – soal, dan alur dalam *game*. Pada perancangan *game* ini dibuat flowchart untuk menjelaskan bagaimana pemain dari awal membuka *game* sampai akhir dalam memainkan *game* tersebut. Alurnya adalah pemain membuka *game* tersebut dan menekan tombol *play*, setelah itu pemain akan diarahkan ke dalam sebuah tampilan yang berisi pengisian nama dan hasil *score* yang baru dimainkan, pemain memasukkan nama untuk nantinya akan ditampilkan dalam permainan, setelah pemain memasukkan nama lalu pemain menekan tombol mencari lawan, pemain akan diminta menunggu untuk dapat mencari lawan, setelah pemain telah mendapatkan lawan bermain, para pemain langsung dapat bermain dalam menjawab soal – soal yang ada, pemain diharuskan mendapat *score* tertinggi untuk dapat memenangkan permainan tersebut, setelah permainan selesai pemain akan ditampilkan *score* yang diperoleh dan nama pemenang dalam permainan tersebut.



Gambar 4. 2 Flowchart alur permainan

4.2 Pembuatan Game

Dalam pembuatan *game* ini diperlukan sebuah *game engine*, *game engine* yang dipakai adalah *game engine* Unity, dan untuk bisa memainkan secara *multiplayer* dibutuhkan *platform* tambahan yaitu Photon Unity Networking (PUN). Berikut pembuatan dan pembahasan pembuatan *game kuis online* ini.

4.2.1 ID App Photon Unity Networking

Sebelum membuat *game*, langkah awal harus mempunyai id app terlebih dahulu, gunanya untuk menghubungkan *game* unity dengan Photon tersebut. Id app dapat dibuat dengan cara membuat akun di website Photon Unity Networking, setelah membuat akun lalu membuat id app.

4.2.2 Bagian Depan Game

Dalam pembuatan bagian depan *game*, hanya sedikit *asset* yang di gunakan, hanya *background*, dan tombol *start*. Dapat dilihat seperti gambar dibawah ini



Gambar 4.2 Bagian Depan Game

```

15  IEnumerator loadScene()
16  {
17      yield return new WaitForSeconds(0.7f);
18      PhotonNetwork.LeaveRoom();
19      PhotonNetwork.Disconnect();
20      SceneManager.LoadScene(SceneBuildIndexToLoad);
21  }
22  }
23
24  // Start is called before the first frame update
25  void Start()
26  {
27      thisBtn = GetComponent<Button>();
28      thisBtn.onClick.AddListener(() => {
29          StartCoroutine("loadScene");
30          thisBtn.enabled = false;
31      });

```

Script 4.1 Script pada bagian depan

Script yang digunakan untuk bagian depan *game* ini sederhana, hanya untuk berpindah *scene* ke *scene* berikutnya. Dapat dilihat dari gambar 4.3 *coding* pada baris 18 – 19 *coding* untuk memastikan pemain tidak berada dalam room permainan dan belum memasuki server, selanjutnya pada baris 20 *coding* untuk berpindah ke *scene* menu utama, dan pada baris 25 – 30 *coding* untuk mengidentifikasi sebuah UI *button* yang digunakan untuk menjalankan perintah “*loadscene*”.

4.2.3 Menu Utama

Pada bagian menu utama terdapat beberapa bagian seperti nama, tombol pencarian lawan, tombol bantuan dan hasil pertandingan



Gambar 4.3 Tampilan Menu Utama

```

28 void Start () {
29
30     PhotonNetwork.ConnectUsingSettings();
31
32     PhotonNetwork.AutomaticallySyncScene = true;
33     UIController.instance.SetLoading(true);
34     UIController.instance.ShowMessage("Connecting..");
35
36 }

```

Script 4.2 Script koneksi ke server

Penjelasan pada *script* gambar 4.5 adalah untuk menghubungkan pemain pada server yang nantinya pemain dapat terhubung dengan pemain lain dalam satu server.

```

49 nama.text = "Nama : " + _playerName;
50 score.text = "Score : " + PlayerPrefs.GetInt("score").ToString();
51 highscore.text = "Highscore : " + PlayerPrefs.GetInt("highscore").ToString();
52 hasil.text = "Hasil : " + PlayerPrefs.GetString("hasil");

```

Script 4.3 Script tampilan hasil pertandingan

Pada *script* gambar 4.6 digunakan untuk menampilkan hasil pertandingan yang sudah atau baru dilakukan oleh pemain, yang akan ditampilkan berupa nama, *score*, dan hasil pertandingan menang atau kalah.

```

73 public void OnBeginClick()
74 {
75     Audio.instance.GetComponent<AudioSource>().Pause();
76     BeginButton.SetActive(false);
77     CancelButton.SetActive(true);
78     Home.SetActive(false);
79     Tips.SetActive(false);
80     PlayerName.gameObject.SetActive(false);
81     PlayerPrefs.SetString("PlayerName", PlayerName.text);
82
83     UIController.instance.SetLoading(true);
84
85     PhotonNetwork.JoinRandomRoom();
86

```

Script 4.4 Script memulai mencari pemain lawan

Pada *script* diatas menjelaskan kegunaan untuk memulai pencarian lawan bermain. Disaat pemain memulai mencari lawan bermain, pemain akan membuat *room* jika tidak ada *room* yang dibuat pemain lain dan menjadi *player 1*, sedangkan jika ada *room* yang telah dibuat, maka pemain akan bergabung secara otomatis masuk ke *room* tersebut dan menjadi *player 2*.

```

116 private void CreateRoom()
117 {
118     RoomOptions options = new RoomOptions() { IsOpen = true, IsVisible = true, MaxPlayers = (byte)GameSettings.instance.MaxPlayers };
119     PhotonNetwork.CreateRoom("Room " + UnityEngine.Random.Range(1, 1000) ,options);
120 }

```

Script 4.5 Script membuat room

Pada *script* diatas menjelaskan membuat *room* untuk memulai permainan antara 2 pemain, jika *room* sudah mencapai 2 pemain, maka *room* tersebut akan menutup supaya tidak ada lagi pemain yang ikut masuk kedalam *room* tersebut.

4.2.4 Pencarian lawan bermaian

Dibagian mencari lawan bermain, pemain akan menunggu untuk mendapatkan lawan bermain.



Gambar 4.4 Tampilan pencarian lawan bermain


```

88     public void OnCancelClick()
89     {
90         Audio.instance.GetComponent<AudioSource>().Play();
91         tipsBtn.SetActive(false);
92         BeginButton.SetActive(true);
93         CancelButton.SetActive(false);
94         Home.SetActive(true);
95         Tips.SetActive(true);
96         PhotonNetwork.LeaveRoom();
97         UIController.instance.Loading.SetActive(false);
98         UIController.instance.HideMessage();
99     }

```

Script 4.6 Script membatalkan mencari pemain lawan

Pada gambar diatas menjelaskan *script* yang digunakan untuk membatalkan pencarian lawan dan akan kembali ke menu utama

```

public override void OnJoinedRoom()
{
    base.OnJoinedRoom();

    numberOfPlayers = PhotonNetwork.PlayerList.Length;
    string name = PlayerPrefs.GetString("PlayerName");
    PhotonNetwork.NickName = name;

    CheckPlayer();

    _photonView.RPC("RPC_UpdateNumberPlayers", RpcTarget.All);
}

```

Script 4.7 Script bergabung dalam room

Script diatas digunakan untuk bergabung ke *room* yang telah dibuatkan oleh pemain lain, maksimal *room* yang bisa dimasuki hanya 2 pemain

4.2.5 Permainan

Pada bagian permainan, pemain akan berebut *score* dengan pemain lain dengan cara menjawab soal – soal, semakin banyak pemain menjawab soal, kemungkinan menang dalam permainan sangat tinggi.



Gambar 4.5 Tampilan Permainan kuis

```

45  _questions.Add(new Question(2, "Hari pramuka diperingati setiap tanggal ...", false ));
46  answersList.Add(new Answer(1, "14 Agustus", true));
47  answersList.Add(new Answer(2, "15 Agustus", false));
48  answersList.Add(new Answer(3, "16 Agustus", false));
49  answersList.Add(new Answer(4, "17 Agustus", false));
50  _answers.Add(answersList);
51
52  _questions.Add(new Question(9, "https://i.postimg.cc/y6T74tZM/Soal1.png", true));
53  answersList = new List<Answer>();
54  answersList.Add(new Answer(1, "6 ", true));
55  answersList.Add(new Answer(3, "5", false));
56  answersList.Add(new Answer(2, "7", false));
57  answersList.Add(new Answer(4, "4", false));
58  _answers.Add(answersList);
59

```

Script 4.8 Script memasukkan soal – soal kuis

Pada gambar diatas menjelaskan kegunaan *script* untuk menambah atau membuat soal – soal. Soal – soal yang ditampilkan bisa berupa soal teks dan soal bergambar.


```

352 public void ShowQuestion(Question question, List<Answer> answers)
353 {
354     StopAllCoroutines();
355     StartCoroutine("QuestionTimer");
356
357     if (question.IsImage)
358     {
359         QuestionText.gameObject.SetActive(false);
360         QuestionImage.gameObject.SetActive(true);
361         StartCoroutine(LoadImage(QuestionImage, question.Description));
362     }
363     else
364     {
365         QuestionImage.gameObject.SetActive(false);
366         QuestionText.gameObject.SetActive(true);
367         QuestionText.text = question.Description;
368     }
369 }

```

Script 4.9 Script menampilkan soal – soal kuis

Script diatas menjelaskan soal – soal yang telah dibuat akan ditampilkan pada permainan kuis. Jika ada soal bergambar maka soal text akan dihilangkan, sebaliknya jika tidak ada soal bergambar, maka yang keluar soal text.

```

418 public void UpdateScore(string playerName, int score)
419 {
420     int playerNumber = GetPlayerNumber(playerName);
421     Debug.Log("number: " + playerNumber + " name: " + playerName + " score: " + score );
422
423
424
425     switch (playerNumber)
426     {
427         case 1:
428             scorePlayer1.text = score.ToString();
429             //PlayerPrefs.SetInt("score", score);
430             break;
431         case 2:
432             scorePlayer2.text = score.ToString();
433             //PlayerPrefs.SetInt("score", score);
434             break;
435     }

```

Script 4.10 Script untuk menampilkan score pemain

Pada script diatas digunakan untuk menampilkan score pemain yang menjawab soal – soal dengan benar dan juga untuk mengupdate score pemain.

```

public void AddScore(string PlayerName, int Points)
{
    Player[] _players = PhotonNetwork.PlayerList;

    for (int i = 0; i < _players.Length; i++)
    {
        if (_players[i].NickName == PlayerName)
        {
            playerScore[i] += Points;
            GameUIController.instance.UpdateScore(PlayerName, playerScore[i]);
        }
    }
}

```

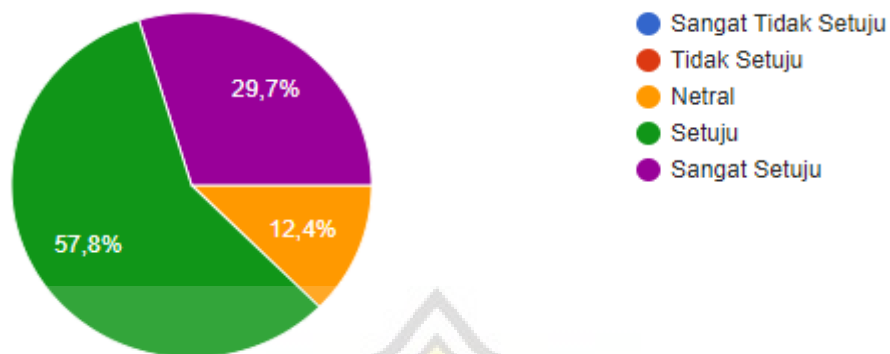
Script 4.11 Script untuk menambah score pemain

Pada *script* diatas digunakan untuk menambahkan *score* pemain. Setiap pemain di indetifikasikan sesuai *nickname* yang diisi di menu utama, jadi pemain yang menjawab soal dengan benar akan mendapatkan *score* dan diupdate di tampilan *score* sesuai *nickname* pemain.

4.3 Hasil Kuesioner

Setelah rancangan pembuatan *game* selesai, langkah selanjutnya adalah menyebarkan *game* kuis dan kuesioner yang nantinya akan di mainkan oleh responden sebanyak 185 orang. Untuk isi kuesioner yang akan diberikan seputar manfaat *game* kuis, kemudahan dalam memainkan, dan minat pemain terhadap *game* kuis tersebut. Berikut hasil kuesioner yang telah disebarkan kepada pemain :

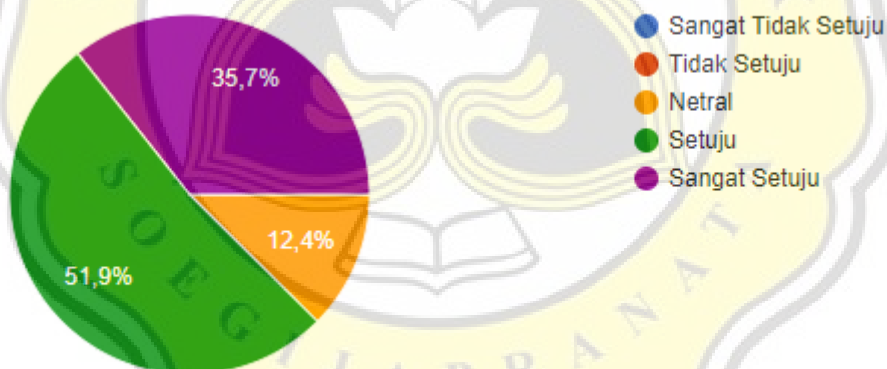
4.3.1 Pemain menikmati permainan edukasi



Gambar 4.6 Hasil jawaban kuesioner pernyataan 1

Dari 185 responden 55 (29,7%) sangat setuju, 107 (57,8%) setuju, dan 23(12,4%) netral dalam menikmati permainan edukasi

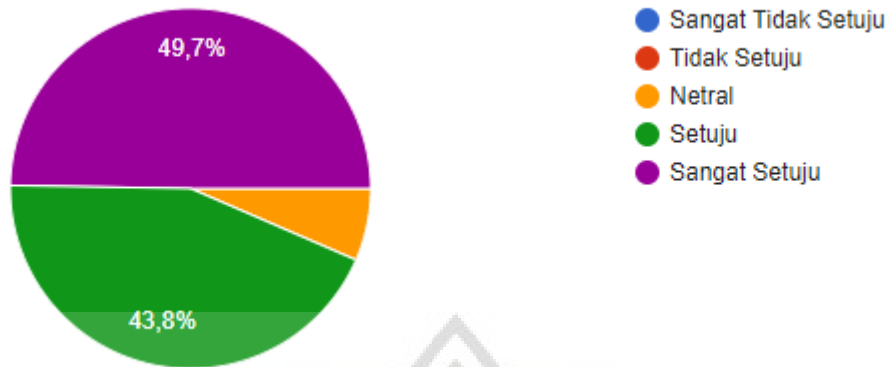
4.3.2 Pemain menikmati permainan edukasi 1 lawan 1



Gambar 4.7 Hasil jawaban kuesioner pernyataan 2

Dari 185 responden 66 (35,7%) sangat setuju, 96 (51,9%), dan 23 (12,4%) netral yang menikmati permainan edukasi 1 lawan 1.

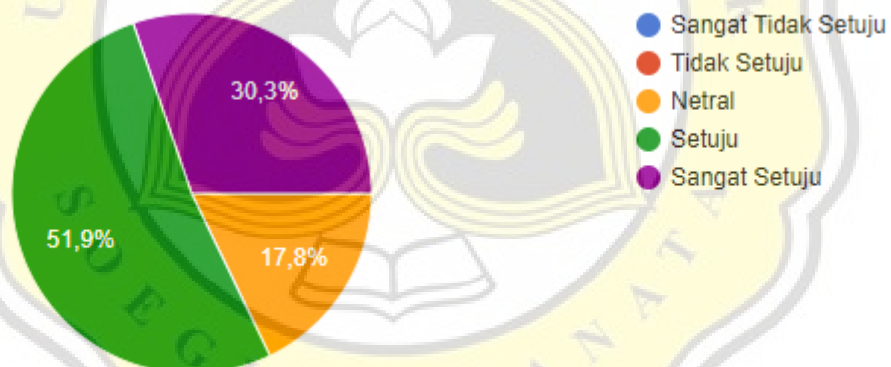
4.3.3 Pemain menikmati musik latar belakang di dalam permainan edukasi



Gambar 4.8 Hasil Jawaban kuesioner pernyataan 3

Dari 185 responden 92 (49,7%) sangat setuju, 81 (43,8%), dan 12 (6,5%) netral yang menikmati musik latar belakang dalam permainan edukasi

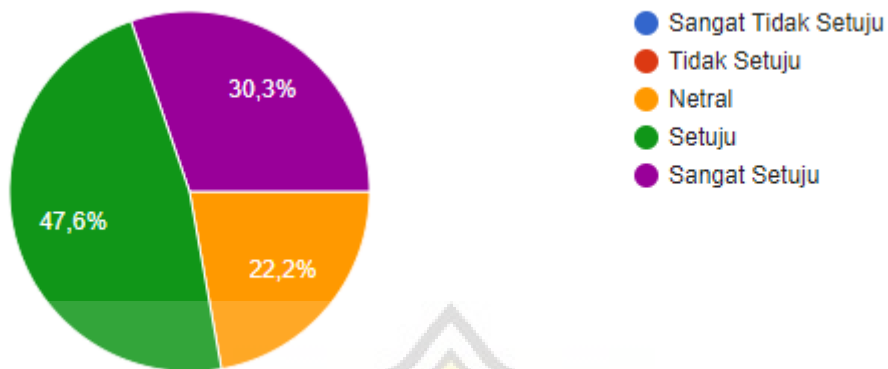
4.3.4 Saya mendapatkan manfaat dari permainan edukasi



Gambar 4.9 Hasil jawaban kuesioner pernyataan 4

Dari 185 responden 56 (30,3%) sangat setuju, 96 (51,9%) setuju, dan 33 (17,8%) netral yang mendapat manfaat dari permainan edukasi.

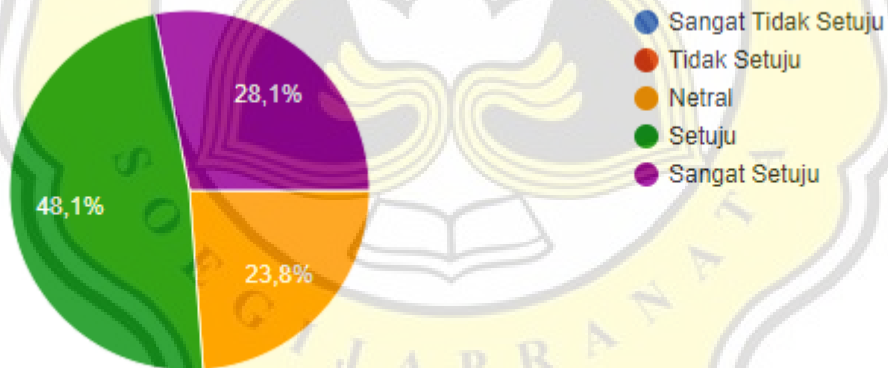
4.3.5 Saya terbantu dalam pembelajaran dengan memainkan permainan edukasi



Gambar 4.10 Hasil jawaban kuesioner pernyataan 5

Dari 185 responden 56 (30,3%) sangat setuju, 88 (47,6%) setuju, dan 41 (22,2%) netral yang terbantu dalam pembelajaran dengan memainkan permainan edukasi

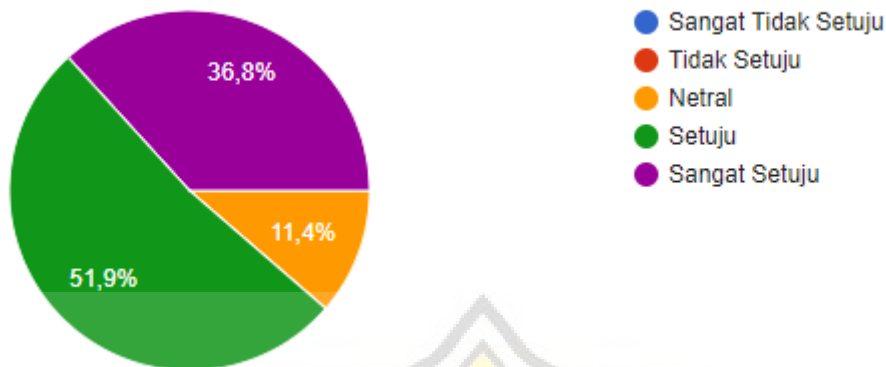
4.3.6 Permainan edukasi dapat mempermudah saya dalam proses pembelajaran



Gambar 4.11 Hasil jawaban kuesioner pernyataan 6

Dari 185 responden 52 (28,1%) sangat setuju, 89 (48,1%), dan 44 (23,8%) netral yang memilih permainan edukasi dapat mempermudah dalam proses pembelajaran.

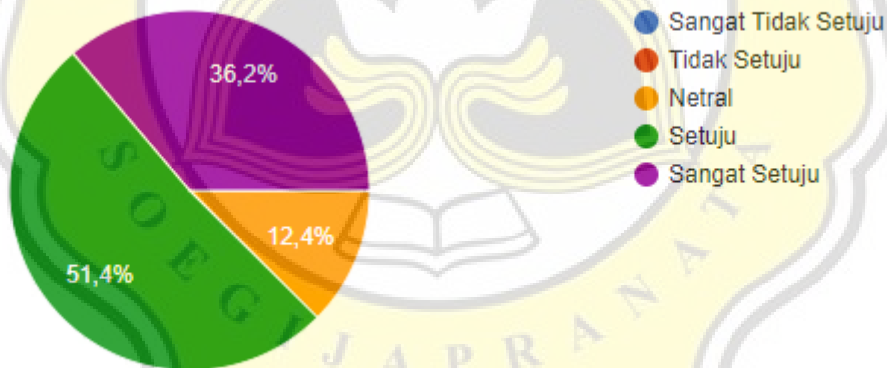
4.3.7 Saya mudah memainkan permainan edukasi ini



Gambar 4.12 Hasil jawaban kuesioner pernyataan 7

Dari 185 responden 68 (36,8%) sangat setuju, 96 (51,9%) setuju, dan 21 (11,4%) netral yang memilih mudah memainkan permainan edukasi.

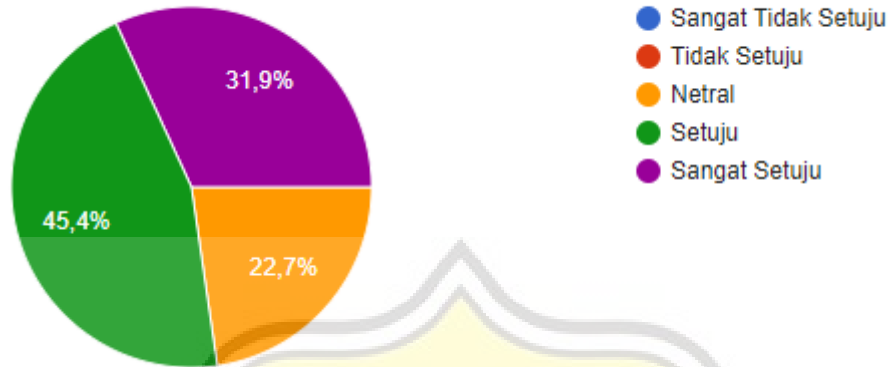
4.3.8 Saya mudah memahami soal – soal permainan edukasi ini



Gambar 4.13 hasil jawaban kuesioner pernyataan 8

Dari 185 responden 67 (36,2%) sangat setuju, 95 (51,4%) setuju, dan 23 (12,4%) netral yang mudah memahami soal – soal permainan edukasi.

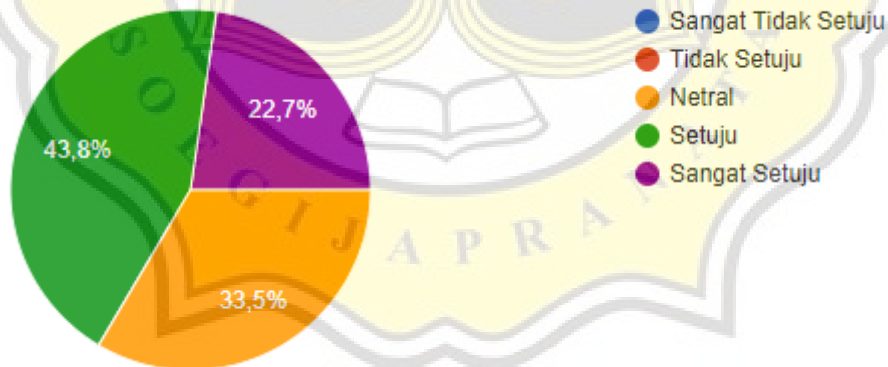
4.3.9 Saya terbantu dengan fitur bantuan untuk menjawab soal di permainan edukasi



Gambar 4.14 Hasil jawaban kuesioner pernyataan 9

Dari 185 responden 59 (31,9%) sangat setuju, 84 (45,4%) setuju, dan 42 (22,7%) netral yang terbantu dengan fitur bantuan untuk menjawab soal di permainan edukasi

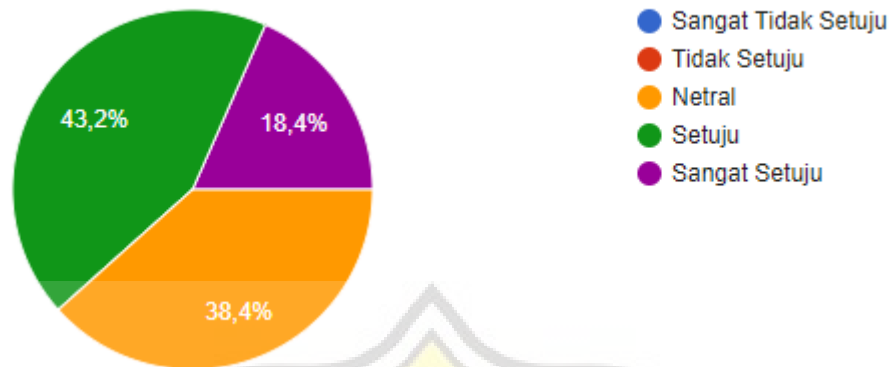
4.3.10 Saya berminat untuk terus memainkan permainan edukasi



Gambar 4.15 Hasil jawaban kuesioner pernyataan 10

Dari 185 responden 42 (22,7%) sangat setuju, 81 (43,8%) setuju, dan 62 (33,5%) netral yang berminat untuk terus memainkan permainan edukasi.

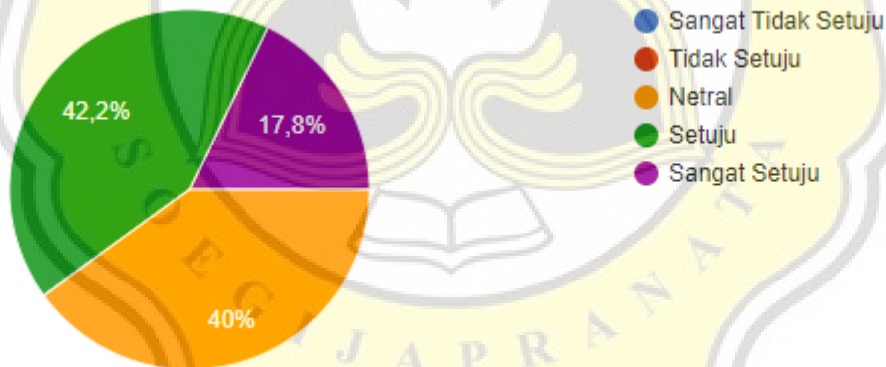
4.3.11 Di masa yang akan datang saya akan terus memainkan permainan edukasi



Gambar 4.16 Hasil jawaban kuesioner pernyataan 11

Dari 185 responden 34 (18,4%) sangat setuju, 80 (43,2%) setuju, dan 71 (38,4%) netral yang akan terus memainkan permainan edukasi di masa yang akan datang

4.3.12 Saya akan memainkan permainan edukasi untuk ke depannya



Gambar 4.17 Hasil jawaban kuesioner pernyataan 12

Dari 185 responden 33 (17,8%) sangat setuju, 78 (42,2%) setuju, dan 74 (40%) netral yang akan memainkan permainan edukasi untuk ke depannya.

4.4 Hasil Uji Statistik

Setelah mendapatkan hasil dari jawaban responden maka dapat diuji statistiknya. Uji statistiknya berupa uji validitas, uji reliabilitas, dan uji korelasi.

4.4.1 Uji Validitas

Dapat dilihat dari tabel disamping setiap variabel yang sama berada dalam komponen yang sama, itu menunjukkan bahwa setiap variabel valid

Rotated Component Matrix^a

	Component	
	1	2
HM1	,416	,672
HM2	,282	,745
HM3	-,010	,821
PE1	,664	,487
PE2	,707	,432
PE3	,797	,297
EE1	,396	,734
EE2	,396	,666
BI1	,867	,219
BI2	,879	,216
BI3	,874	,179

Tabel 4.1 Uji Validitas

4.4.2 Uji Reliabilitas

Variabel	Cronbach's alpha	Keterangan
HM	0,775	Acceptable
PE	0,907	Excellent
EE	0,898	Good
BI	0,937	Excellent

Tabel 4.2 Uji Reliabilitas

Dapat dilihat dari tabel diatas variabel HM (Hedonic Motivation) memiliki Cronbach's alpha 0,775 (Acceptable), variabel PE (Performance Expectancy) memiliki Cronbach's alpha 0,907 (Excellent), variabel EE (Effort Expectancy) memiliki Cronbach's alpha 0,898 (Good), dan variabel BI (Behavioral Intention) memiliki Cronbach's alpha 0,937 (Excellent).

4.4.3 Uji Korelasi

Correlations

		SHM	SPE	SEE1	SBI
SHM	Pearson Correlation	1	,595**	,618**	,482**
	Sig. (2-tailed)		,000	,000	,000
	N	185	185	185	185
SPE	Pearson Correlation	,595**	1	,569**	,706**
	Sig. (2-tailed)	,000		,000	,000
	N	185	185	185	185
SEE	Pearson Correlation	,618**	,569**	1	,559**
	Sig. (2-tailed)	,000	,000		,000
	N	185	185	185	185
SBI	Pearson Correlation	,482**	,706**	,559**	1
	Sig. (2-tailed)	,000	,000	,000	
	N	185	185	185	185

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

Tabel 4.3 Uji Korelasi

Dapat dilihat dari tabel diatas bahwa variabel BI (Behavioral Intention) saling berkorelasi dengan variabel HM(Hedonic Motivation), PE(Performance Expectancy), dan EE(Effort Expectancy)