

PEMBUATAN SKOR PENILAIAN PADA PERMAINAN *UCING BELING* MENGGUNAKAN ALGORITMA FINITE STATE MACHINE

MAKING AN ASSESSMENT SCORE IN *UCING BELING* GAME USING THE FINITE STATE MACHINE ALGORITHM

Arivo Trimajaya¹, Anton Siswo Raharjo Ansori,² Ashri Dinimaharawati,³

^{1,2,3} Universitas Telkom, Bandung

rivokidz@student.telkomuniversity.ac.id¹, raharjo2@telkomuniveristy.ac.id²,

ashridini@telkomuniversity.ac.id³

Abstrak

Game ucing beling adalah permainan tradisional anak-anak di tanah Sunda, Jawa Barat, Indonesia yang dimainkan agar melatih fokus, ketelitian anak dan juga hiburan. Untuk mengetahui siapa yang menang dan siapa yang kalah pada *game ucing beling* maka dibutuhkan skor penilaian pada *game ucing beling*. Pada penelitian ini akan menggunakan algoritma *Finite State Machine* dalam penerapan skor penilaian pada *game ucing beling*. *Finite State Machine* menggunakan tiga prinsip kerja yaitu *State* (keadaan), *Event* (kejadian) dan *Action* (tindakan). Hasil dari penelitian ini akan membuat skor pada *game ucing beling* dengan sistem pembuatan skor berdasarkan waktu, waktu akan dibuat dari 25 detik hingga 0 detik, apabila *player* menemukan beling pada detik 25 maka akan mendapatkan skor 100 point, apabila *player* tidak menemukan beling dalam waktu 1 detik maka akan dikurangi point skor sebesar 4 point per detiknya dan apabila waktu habis *player* tidak menemukan beling maka penilaian skor akan mejadi 0 point. Dalam penampilan skor akan menentukan juara 1, juara 2 dan seterusnya, dalam penampilan skor akan menggunakan fitur *highscore*, dalam fitur *highscore* menggunakan fitur *sorting* yang ada pada *unity2D*.

Kata kunci : *Game, Game Tradisional, Penilaian Skor, Finite State Machine.*

Abstract

Ucing beling game is a traditional children's game in Sundanese, West Java, Indonesia that is played to train the focus, accuracy of children and also entertainment. To know who won and who lost the *ucing beling* game, it takes a scoring score on the *ucing beling* game. In this study will use *Finite State Machine* algorithm in the application of scoring on *ucing beling* game. *Finite State Machine* uses three working principles namely *State* (state), *Event* (event) and *Action* (action). The result of this study will make the score in *ucing beling* game with a score-making system based on time, time will be made from 25 seconds to 0 seconds, if the player finds a rivet at 25 seconds it will get a score of 100 points, if the player does not find the beling within 1 second it will be reduced by the score point by 4 points per second and if the player does not find the beling eating the score assessment will be at 0 points. In the appearance of the score will determine the 1st place, 2nd place and so on, in the appearance of the score will use the *highscore* feature, in the *highscore* feature using the *sorting* feature that is on *unity2D*.

Keywords: *Game, Game Tradisional, Penilaian Skor, Finite State Machine.*

1. Pendahuluan

Game *ucing beling* adalah permainan tradisional anak-anak di tanah Sunda, Jawa Barat, Indonesia yang dimainkan agar melatih fokus, ketelitian anak dan juga hiburan. Untuk mengetahui siapa yang menang dan siapa yang kalah pada *game ucing beling* maka dibutuhkan skor penilaian pada *game ucing beling*. Sebuah *game* tidak akan menarik atau tidak akan seru apabila tidak ada goals yang akan dicapai atau tidak akan tertantang apabila tidak mempunyai skor penilaian dari permainan yang sudah kita mainkan. Skor penilaian adalah sebuah nilai yang akan menentukan siapa pemenang dari *game* tersebut yang berbentuk angka dan lain sebagainya.

Dari permasalahan di atas peneliti mempunyai ide agar *game* ini menjadi lebih menarik dengan membuat skor penilaian dari *game ucing beling*. Skor penilaian pada *game ucing beling* yang akan dibuat menggunakan algoritma *Finite State Machine*. Algoritma *Finite State Machine* adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *State* (keadaan), *Event* (kejadian) dan *Action* (tindakan). Pada skor penilaian di *game ucing beling* akan digunakan algoritma *Finite State Machine* pada saat *player* menemukan semua beling yang disembunyikan. Skor penilaian akan bergantung dengan pencarian beling/kaca pada bulatan yang telah ditentukan. Pencarian beling akan dilakukan oleh *player* dan akan dimulai dari waktu 25 detik untuk mencari beling, apabila *player* berhasil menemukan semua beling dalam waktu 25

detik, maka dia akan mendapatkan skor penilaian 100 dan apabila *player* mencari beling lebih dari waktu 25 detik maka tiap satu detik yang dilewatkan akan dikurangi skor sebesar 4 per detik dan akan berlanjut ke detik-detik berikutnya hingga *player* bisa menemukan semua beling yang telah di sembunyikan. Apabila semua beling yang disembunyikan telah ditemukan oleh *player*, maka *player* tersebut akan mendapatkan skor, skor yang didapatkan *player* bergantung dengan waktu yang dia gunakan dalam menemukan beling. Skor yang didapatkan masing-masing *player* akan ditampilkan pada saat *player* menemukan semua beling atau waktu yang berjalan habis.

2. Landasan Teori

2.1 Ucing Beling

Ucing Beling adalah permainan tradisional tanah Sunda. Permainan ini biasanya dimainkan oleh dua sampai lima orang anak. Dalam permainan *ucing* beling ini *ucing* akan mencari beling yang disembunyikan, di mana *ucing* sebagai orang pencari dan beling adalah benda pecahan/serpihan kaca yang akan dicari oleh *ucing*. Beling yang akan dicari pun tidak bisa disembunyikan di sembarang tempat akan tetapi ada aturan yang harus diikuti seperti beling harus berada di dalam garis batas tak teratur yang sudah disepakati oleh seluruh pemain. Dalam area tersebut beling akan disembunyikan atau disamarkan sampai tidak terlihat sama sekali, bisa diselipkan di antara rumput, ditutupi pasir dan juga ditindih batu agar si beling susah untuk ditemukan. Adapun alat bantu yang biasanya digunakan oleh *ucing* dalam mencari beling yang di sembunyikan seperti sebatang lidi, yang terpenting alat bantu tersebut berbentuk memanjang dan berujung lancip agar memudahkan mencungkil pasir agar si beling bisa ditemukan. Dalam permainan ini masing-masing anak atau pemain akan membuat sebuah area dan menyembunyikan belingnya, mereka saling mencari beling satu sama lain secara bergantian, misal A mencari beling yang disembunyikan oleh B, jika beling B ditemukan oleh A maka B akan mencari beling yang disembunyikan oleh C, dan jika B menemukan beling milik C maka C akan mencari beling milik D begitu seterusnya hingga orang terakhir akan mencari beling orang pertama. Manfaat dari permainan ini di antaranya adalah untuk melatih fokus, ketelitian dan kesabaran si anak[1].

2.2 Algoritma Finite State Machine

Finite State Machine (FSM) merupakan sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *State* (Keadaan), *Event* (Kejadian) dan *Action* (Aksi)[2].

2.3 Skor Penilaian

Pada hakikatnya pemberian skor adalah proses perubahan jawaban instrumen menjadi angka-angka yang merupakan nilai kuantitatif dari suatu jawaban terhadap item dalam instrumen. Angka-angka hasil penilaian selanjutnya diproses menjadi nilai-nilai. Menurut Kamus Besar Bahasa Indonesia (KBBI), penskoran adalah proses, cara, pembuatan skor.

2.4 C#

C# atau yang dibaca *C sharp* adalah bahasa pemrograman sederhana yang digunakan untuk tujuan umum, dalam artian bahasa pemrograman ini dapat digunakan untuk berbagai fungsi misalnya untuk pemrograman *server-side* pada *website*, membangun aplikasi desktop ataupun *mobile*, pemrograman *game* dan sebagainya. Selain itu *C#* juga bahasa pemrograman yang berorientasi objek, jadi *C#* juga mengusung konsep objek seperti *inheritance*, *class*, *polymorphism* dan *encapsulation*[3].

2.5 Unity Engine

Unity adalah aplikasi yang berguna untuk mengembangkan *game* lintas platform yang dirancang agar mudah digunakan. Unity luar biasa dan memiliki banyak kombinasi dengan aplikasi profesional. Editor Unity dibuat dengan antarmuka pengguna yang sederhana. Penerbit ini dibuat setelah menghabiskan ribuan jam mencoba untuk menjadi yang teratas dalam peringkat penerbit game. Grafik kesatuan dibuat dengan grafik tingkat tinggi untuk OpenGL dan DirectX. Unity mendukung semua format file dan mendukung format paling populer, terutama semua format untuk aplikasi artistic[4].

3. Perancangan Sistem

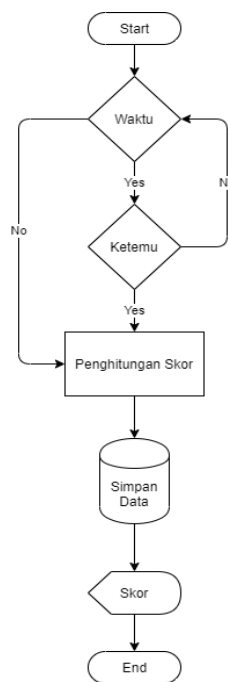
3.1 Desain Sistem

Sistem yang dirancang adalah sistem penilaian skor pada *game ucing* beling. Algoritma yang digunakan dalam perancangan sistem penilaian skor yaitu algoritma *Finite State Machine*, dalam perancangannya penulis menggunakan unity2D untuk merancang sistem penilaian skor menggunakan algoritma *Finite State Machine* yaitu *State* (keadaan),

State di sini akan digunakan untuk keadaan waktu dalam satuan detik, *Event* (Kejadian), *Event* di sini akan digunakan untuk kejadian yang terjadi misalkan *player* menemukan beling, *Action* (Aksi), *Action* disini akan digunakan untuk aksi misalkan *player* menemukan beling pada waktu dalam satuan detik ke berapa *player* menemukan beling yang disembunyikan dan pada waktu *player* menemukan beling akan diberikan aksi mendapatkan skor yang telah di desain oleh penulis.

Sistem penilaian skor kurang lebih akan menjadi seperti ini. Pencarian beling akan dilakukan oleh *player* dan akan dimulai dari waktu 25 detik untuk mencari beling, apabila *player* berhasil menemukan semua beling dalam waktu 1 detik, maka dia akan mendapatkan skor penilaian 100 dan apabila *player* mencari beling lebih dari waktu 1 detik maka setiap satu detik yang dilewatkan akan di kurangi skor sebesar 4 per detik dan akan berlanjut ke detik-detik berikutnya hingga *player* bisa menemukan beling dan apabila *player* tidak menemukan semua beling hingga waktu habis maka skor akan menjadi 0. Apabila beling yang disembunyikan telah ditemukan oleh *player*, maka *player* tersebut akan mendapatkan skor, skor yang didapatkan *player* bergantung dengan waktu yang dia gunakan dalam menemukan semua beling.

3.2 Diagram Blok



Gambar 1. Diagram Blok

Langkah – Langkah yang dilakukan dalam penilaian skor pada *game ucing* beling menggunakan algoritma *Finite State Machine* adalah sebagai berikut:

1. Dimulai dengan menekan tombol mulai permainan, maka *game* akan mulai dan waktu pun berjalan.
2. Apabila *player* menemukan beling maka akan masuk ke penghitungan skor dan jika *player* belum menemukan beling maka *player* akan terus mencari sampai *player* menemukan beling dan jika *player* belum menemukan beling sampai waktu berakhir maka kondisi *score* otomatis ditampilkan oleh sistem. Pada bagian waktu dan ketemu akan menggunakan algoritma *Finite State Machine*
3. Setelah selesai penghitungan skor maka akan disimpan pada data yang akan berguna untuk menentukan pemenang.
4. Skor setiap *player* yang bermain akan ditampilkan.

3.3 Fungsi dan Fitur

Penilaian skor pada *game ucing* beling ini akan digunakan untuk menentukan siapa pemenang pada *game* ini, agar sebuah *game* menjadi sempurna maka butuh penilaian skor untuk mencari pemenangnya dan juga berfungsi untuk kebanggaan terhadap diri sendiri atau penilaian pada diri sendiri, seberapa kemampuan yang bisa dicapai. Fiturnya nanti akan ada fitur *high score* pada *game ucing* beling.

4. Skenario Pengujian

4.1 Implementasi

4.1.1 Implementasi Fungsi Waktu

Dalam implementasinya fungsi ini merupakan waktu yang berjalan yang berguna untuk menghitung skor nantinya. Pembuatan fungsi waktu ini agar *player* bisa menemukan beling pada waktu yang telah ditentukan. Waktu diatur selama 25 detik hingga 0 detik dan dapat ditampilkan pada game. Berikut beberapa potongan kode yang disisipkan pada gameobject "Kaca".

```
public class RNGKACA : MonoBehaviour
{
    float currentTime = 0f;
    float startingTime = 25f;
    [SerializeField] Text waktuText;
    bool timerActive = false;
    void Start()
    {
        timerActive = true;
        currentTime = startingTime;
    }
    void Update()
    {
        if (timerActive)
        {
            currentTime -= 1 * Time.deltaTime;
            waktuText.text = currentTime.ToString("0");
        }
    }
}
```

4.1.2 Implementasi Fungsi Ketemu

Dalam implementasinya fungsi ini akan berguna untuk mematikan waktu yang berjalan serta menampilkan "popups" skor hanya pada saat beling di klik. Berikut beberapa potongan kode yang disisipkan pada gameobject "Kaca".

```
void Update()
{
    ketemu();
}

public void ketemu()
{
    if (belingcount == 5)
    {
        timerActive = false;
        Popups.SetActive(true);
    }
}
```

4.1.3 Implementasi Fungsi Perhitungan Skor

Dalam implementasinya fungsi ini merupakan tempat perhitungan skor berdasarkan waktu dan ketemu. Apabila *player* menemukan beling di waktu 25 detik maka skor akan 100 point, skor akan berkurang 4 point setiap detiknya hingga waktu abis. Apabila tidak ketemu dan waktu habis maka perhitungan skor akan dibuat menjadi 0 point. Skor

yang didapatkan setiap *player* akan disimpan pada *PlayerPrefs* yang akan digunakan oleh simpan data nantinya. Berikut beberapa potongan kode yang disisipkan pada gameobject "Kaca".

```

void Update()
{
    Skoring();
}

public void Skoring()
{
    if (timerActive)
    {
        currentTime -= 1 * Time.deltaTime;
        waktuText.text = currentTime.ToString("0");
        if (currentTime > 24)
        {
            scoreAmount = 100f;
            PlayerPrefs.SetFloat(PlayerPrefsScoreKey, scoreAmount);
        }
        if (currentTime <= 25)
        {
            scoreAmount -= pointIncreasedPerSecond *
Time.deltaTime;
            PlayerPrefs.SetFloat(PlayerPrefsScoreKey, scoreAmount);
        }
        if (currentTime <= 0)
        {
            currentTime = 0;
            scoreAmount = 0f;
            scoreAmount = pointIncreasedPerSecond * Time.deltaTime;
            PlayerPrefs.SetFloat(PlayerPrefsScoreKey, scoreAmount);
            Popsups.SetActive(true);
        }
    }
}
}

```

4.1.4 Implementasi Fungsi Simpan Data

Dalam implementasinya fungsi ini merupakan tempat penyimpanan data skor. Penyimpanan data skor akan di tempatkan pada *PlayerPrefs*, data skor yang disimpan akan digunakan untuk *highscore* dan penampilan skor nantinya. Beberapa potong kode yang disisipkan pada fungsi perhitungan skor adalah sebagai berikut:

```

private const string PlayerPrefsScoreKey = "Score";

PlayerPrefs.SetFloat(PlayerPrefsScoreKey, scoreAmount);

```

4.1.5 Implementasi Fungsi Highscore dan Fungsi Penampilan skor

Dalam implementasinya fungsi ini akan digunakan sebagai tempat penyortiran data yang sudah disimpan pada fungsi simpan data. Data yang sudah disimpan pada simpan data akan di panggil di fungsi ini, dan datanya akan disortir dan data yang sudah disortir akan ditempatkan pada *text* yang sudah di sediakan pada penampilan skor nantinya.

Data yang sudah disimpan akan dipanggil dan ditempatkan pada list, list atau nama lainnya array. Beberapa potongan kode pemanggilan data dan ditempatkan pada list adalah sebagai berikut :

```

public List<int> scores;

newscore1 = ((int)PlayerPrefs.GetFloat("score1"));
newscore2 = ((int)PlayerPrefs.GetFloat("score2"));
newscore3 = ((int)PlayerPrefs.GetFloat("score3"));
newscore4 = ((int)PlayerPrefs.GetFloat("score4"));
newscore5 = ((int)PlayerPrefs.GetFloat("score5"));

scores.Add(newscore1);
scores.Add(newscore2);
scores.Add(newscore3);
scores.Add(newscore4);
scores.Add(newscore5);

```

Data yang sudah ditambahkan pada list akan disortir berdasarkan dari nilai tertinggi ke nilai terendah dengan potongan kode sebagai berikut :

```

scores.Sort();
scores.Reverse();

```

Beberapa potongan kode yang dibuat pada fungsi *high score* dan fungsi penampilan skor adalah sebagai berikut:

```

public class ScoreWinner : NetworkBehaviour
{
    [Header("UI TEXT")]
    [SerializeField] private Text Player1Score;
    [SerializeField] private Text Player2Score;
    [SerializeField] private Text Player3Score;
    [SerializeField] private Text Player4Score;
    [SerializeField] private Text Player5Score;
    [SerializeField] private Text Player1Name;
    [SerializeField] private Text Player2Name;
    [SerializeField] private Text Player3Name;
    [SerializeField] private Text Player4Name;
    [SerializeField] private Text Player5Name;
    [SerializeField] private Text Winner;

    public int newscore1 = 0;
    public int newscore2 = 0;
    public int newscore3 = 0;
    public int newscore4 = 0;
    public int newscore5 = 0;

    public List<int> scores;
    public int FirstPlaceScore = 0;
    public int SecondPlaceScore = 0;
    public int ThirdPlaceScore = 0;
    public int FourthPlaceScore = 0;
    public int FifthPlaceScore = 0;

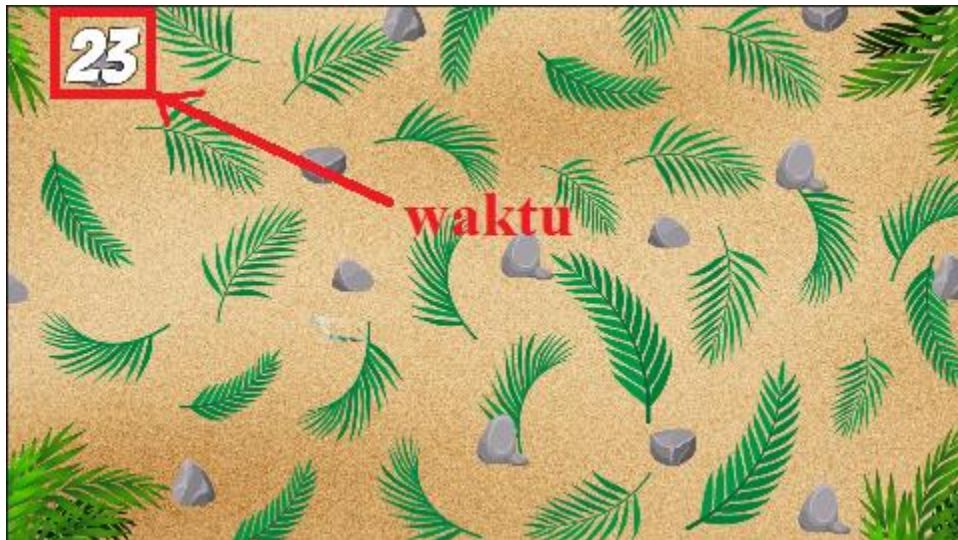
    public GameObject Player1panelhs;
    public GameObject Player2panelhs;
    public GameObject Player3panelhs;
    public GameObject Player4panelhs;
    public GameObject Player5panelhs;

```

4.2 Pengujian

4.2.1 Pengujian Tampilan Fungsi Waktu

Pengujian tampilan fungsi waktu bertujuan untuk menguji apakah waktu berjalan dari 25 detik hingga 0 detik sesuai dengan rancangan yang telah dirancang guna agar tidak terjadi bug atau error pada saat perhitungan skor nantinya.



Gambar 2. Hasil Pengujian Fungsi Waktu

4.2.2 Pengujian Tampilan Fungsi Ketemu

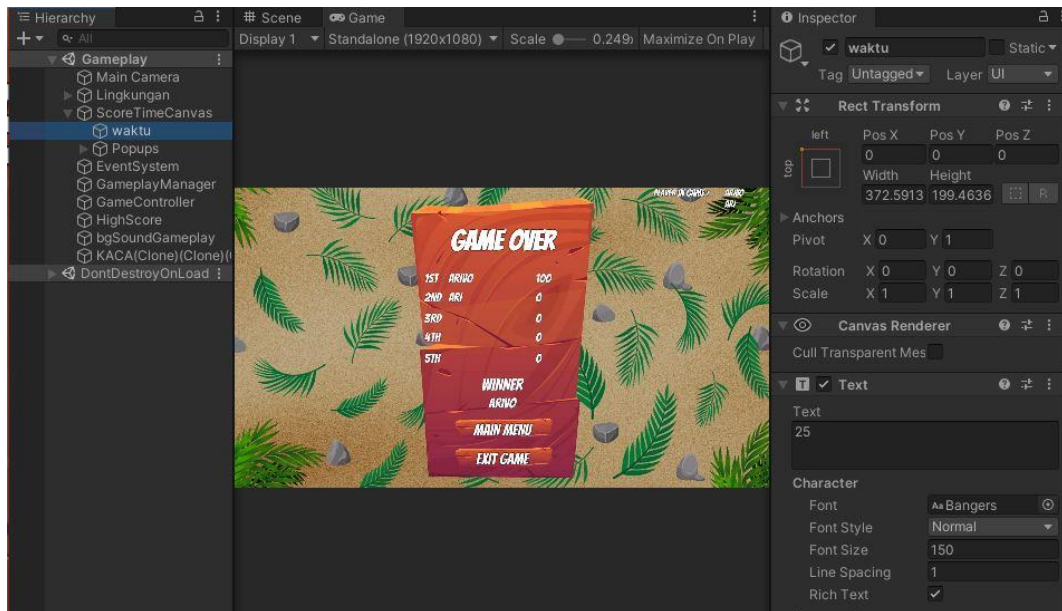
Pengujian tampilan fungsi ketemu bertujuan untuk menguji apakah fungsi ketemu dapat menampilkan *popups* dan mematikan waktu yang berjalan sesuai dengan yang dirancang guna agar tidak terjadi bug atau error pada saat penampilan skor dan perhitungan skor nantinya.



Gambar 3. Hasil Pengujian Fungsi Ketemu

4.2.3 Pengujian Tampilan Fungsi Perhitungan Skor, Fungsi Simpan Data, Fungsi Highscore Dan Fungsi Penampilan Skor

Pengujian Tampilan fungsi ini bertujuan untuk menguji apakah skor tampil sesuai dengan yang telah dirancang pada fungsi perhitungan skor, fungsi simpan data, fungsi *highscore* dan fungsi penampilan *score* dengan urutan dari terbesar hingga terkecil.



Gambar 4. Hasil Pengujian Fungsi Highscore Dan Tampilan Skor

5. Kesimpulan Dan Saran

5.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan pada tugas akhir ini, maka dapat ditarik kesimpulan bahwa:

1. Waktu dapat berjalan menggunakan algoritma *Finite State Machine* dan tidak ada kesalahan logika hingga waktu dapat tampil sesuai yang diharapkan.
2. Sistem skoring pada game *ucing* beling menggunakan algoritma *Finite State Machine* dapat berjalan sesuai yang diharapkan.
3. Skor dapat ditampilkan pada game *ucing* beling berdasarkan nilai tertinggi ke terendah menggunakan sorting sesuai yang diharapkan.

5.2 Saran

Hasil dari pembuatan skor penilaian pada game *ucing* beling menggunakan algoritma *Finite State Machine* yang telah dilakukan tentunya masih ada kekurangan, seperti sistem skoring hanya bisa dimainkan satu match atau sekali permainan saja. Walaupun dalam tujuannya pembuatan sistem skor sudah memenuhi tujuan Tugas Akhir yang dikerjakan penulis, untuk itu diperlukan lagi pengembangan dalam hal tersebut. Adapun pengembangan yang diperlukan antara lain adalah seperti berikut :

1. Sistem skor yang dapat dimainkan untuk beberapa kali bermain.
2. Tampilan sistem skor yang lebih menarik.

REFERENSI

- [1] Jery Yanuarlan, “*Ucing* Beling, Permainan Latih Fokus & Ketelitian Anak”, 2015. Available: <https://www.jeryanuar.web.id/2015/09/ucing-belingpermainan-latih-fokus.html> [Diakses 2 Desember 2020, 04:53:00 WIB].
- [2] Rich, E, Automata, Computability, and Complexity, Theory and Applications. United States of America : Pearson Prentice Hall, 2009.
- [3] Freddset, “Pengenalan Bahasa Pemrograman C#”, 2019. Available : <https://freddset.wordpress.com/2019/05/29/pengenalan-bahasapemrograman-c/> [Diakses 03 Desember 2020, 20:31:00 WIB]
- [4] Wahyupjl, “Apa Itu Unity 3D”, 2018. Available <https://eventkampus.com/blog/detail/1474/apa-itu-unity-3d> [Diakses 17 Juli 2021, 16:01:00 WIB]