



REPUBLIK INDONESIA  
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

# SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202258133, 27 Agustus 2022

## Pencipta

Nama : **Tedi Nizar Afrida dan Dinan Yulianto, S.T., M.Eng.**

Alamat : Tamanan Wetan, Tamanan, Banguntapan, Bantul, DI  
YOGYAKARTA, 55192

Kewarganegaraan : Indonesia

## Pemegang Hak Cipta

Nama : **UNIVERSITAS AHMAD DAHLAN**

Alamat : Jl. Pramuka 5F, Pandeyan, Umbulharjo, Yogyakarta, DI  
YOGYAKARTA, 55161

Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**

Judul Ciptaan : **Pengembangan Puzzle Dengan Augmented Reality Untuk  
Pembelajaran Bangun Ruang Siswa Sekolah Dasar**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 26 Agustus 2022, di Yogyakarta

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000373867

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n Menteri Hukum dan Hak Asasi Manusia  
Direktur Jenderal Kekayaan Intelektual  
u.b.  
Direktur Hak Cipta dan Desain Industri

Anggoro Dasananto  
NIP.196412081991031002

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

a. Desain Maker / Puzzle





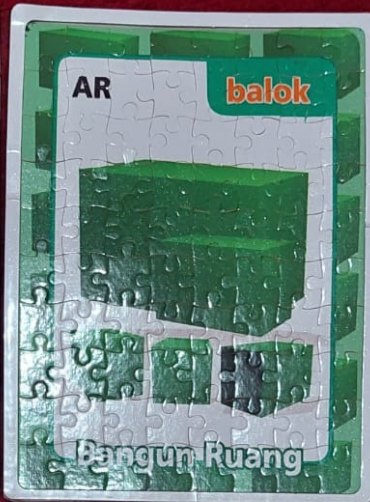


**b. Implementasi Marker / Puzzle**









### c. Tampilan Loading



```
1. public class Scene_Manager : MonoBehaviour {  
2.     public void open_menu()  
3.     {  
4.         SceneManager.LoadScene ("Menu");  
5.     }  
6. }
```

#### Keterangan:

1. Kode 1-6, merupakan kode dalam Scene\_Manager yang dideklarasikan dalam fungsi open\_main yang isinya adalah perpindahan scene sesuai dengan nama scene yang diketikkan pada inspector Unity (dalam hal ini scene menu).



```
1. public class backsound : MonoBehaviour
2. {
3.     // Start is called before the first frame update
4.     void Start()
5.     {
6.         DontDestroyOnLoad(gameObject);
7.     }
8.     // Update is called once per frame
9.     void Update()
10.    {
11.        if (SceneManager.GetActiveScene().name == "Menu")
12.        {
13.            if (!GetComponent().isPlaying)
14.            {
15.                GetComponent().Play();
16.            }
17.        }
18.    }
19. }
```

**Keterangan:**

1. Kode 4-7, merupakan kode yang dideklarasikan dalam fungsi *DontDestroyOnLoad* berfungsi supaya *backsound* didalam game-objek tidak terhapus ketika berpindah ke *scene* lain. Kondisi hanya akan *play* ketika sudah masuk kedalam *scene* main.
2. Kode 11-15, merupakan kode yang mengkondisikan suara *back-sound* ketika sudah berada didalam *scene* main.

#### d. Tampilan Main Menu



```
1. public class Scene_Manager : MonoBehaviour
2. {
3.     public void Buka_ar1()
4.     {
5.         SceneManager.LoadScene("AR");
6.     }
7.     public void Buka_ar2()
8.     {
9.         SceneManager.LoadScene("Panduan");
10.    }
11.    public void Buka_ar3()
12.    {
13.        SceneManager.LoadScene("kuis");
14.    }
15.    public void Buka_ar4()
16.    {
17.        SceneManager.LoadScene("Keluar");
18.    }
19.    public void Buka_ar7()
20.    {
21.        SceneManager.LoadScene("profil");
22.    }
```

```
23. }
```

**Keterangan:**

1. Kode 1-23, merupakan kode dalam Scene\_Manager yang dideklarasikan dalam fungsi Buka\_ar1, Buka\_ar2, Buka\_ar3, Buka\_ar4, Buka\_ar7 yang isinya adalah perpindahan scene sesuai dengan nama scene yang diketikkan pada inspector Unity (dalam hal ini scene AR, Panduan, kuis, Keluar dan profil).

```
1.  public class sfx_btn : MonoBehaviour, IPointerDownHandler
2.  {
3.      public void OnPointerDown(PointerEventData eventData)
4.      {
5.          if (GameObject.Find("sfx_button") != null)
6.          {
7.              GameObject.Find("sfx_button").GetComponent()
              .Play();
8.          }
9.      }
10. }
11. public void Buka_ar3()
12. {
13.     SceneManager.LoadScene("kuis");
14. }
15. public void Buka_ar4()
16. {
17.     SceneManager.LoadScene("Keluar");
18. }
19. public void Buka_ar7()
20. {
21.     SceneManager.LoadScene("profil");
22. }
23. }
```

Keterangan:

1. Kode 1-7, merupakan kode mencari gameobjek dari suara tombol dimana ketika gameobjek ditemukan maka suara tombol langsung dijalankan.

#### e. Tampilan Profil

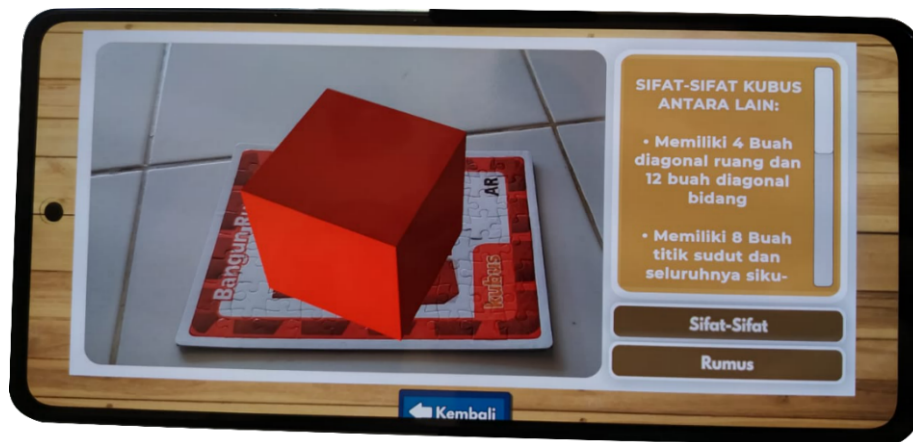


```
1. public class Scene_Manager : MonoBehaviour {
2.     public void open_menu()
3.     {
4.         SceneManager.LoadScene ("Menu");
5.     }
6. }
```

Keterangan:

1. Kode 1-6, merupakan kode dalam Scene\_Manager yang dideklarasikan dalam fungsi open\_main yang isinya adalah perpindahan scene sesuai dengan nama scene yang diketikkan pada inspector Unity (dalam hal ini scene menu).

## f. Tampilan Mulai AR



```
1. public class Rotasi3D : MonoBehaviour
2. {
3.     public bool Terputar = false;
4.     public Vector3 KecepatanPutaran;
5.
6.     // Update is called once per frame
7.     void Update()
8.     {
9.         if (Terputar)
10.        {
11.            transform.Rotate(
12.                KecepatanPutaran.x * Time.deltaTime
13.                * 10,
14.                KecepatanPutaran.y *
15.                Time.deltaTime * 10,
16.                KecepatanPutaran.z *
17.                Time.deltaTime * 10
18.            );
19.        }
20.    }
21.    private void OnMouseDown()
22.    {
23.        if (!Terputar)
24.        {
25.            Terputar = true;
26.            KecepatanPutaran = new Vector3(1, 1, 1);
27.        }
28.    }
29. }
```

```

22.             Terputar = true;
23.         }
24.     else
25.     {
26.         Terputar = false;
27.     }
28. }
29. }
30. public class Rotasi3D : MonoBehaviour
31. {
32.     public bool Terputar = false;
33.     public Vector3 KecepatanPutaran;

```

**Keterangan:**

1. Kode 3-4, merupakan kode untuk mendeklarasikan nilai yang akan dimasukkan kedalam objek melalui posisi x, y dan z.
2. Kode 12-14, merupakan kode untuk mengeksekusi putaran yang berada pada posisi sumbu x, y dan z. Menambahkan kondisi pada objek yang akan dijalankan. Kondisi berfungsi untuk menjalankan putaran dan berhenti memutar objek dengan menggunakan *sphere colider*.

```

1. public class Ganti : MonoBehaviour
2. {
3.     public GameObject imageAwal;
4.     public GameObject imageTujuan;
5.
6.     public void GantiImageBaru()
7.     {
8.         imageAwal.SetActive(false);
9.         imageTujuan.SetActive(true);
10.    }

```

11. }

Keterangan:

1. Kode 1-9, merupakan kode yang berfungsi untuk berpindah dari *image* awal ke *image* tujuan. Diterapkan pada rumus tombol rumus dan tombol sifat.

g. Tampilan Penggunaan AR dan Tampilan Penggunaan Latihan Soal



```
1. public class LINK : MonoBehaviour
2. {
3.     public string Url;
4.
5.     public void Open()
6.     {
7.         Application.OpenURL(Url);
```

```

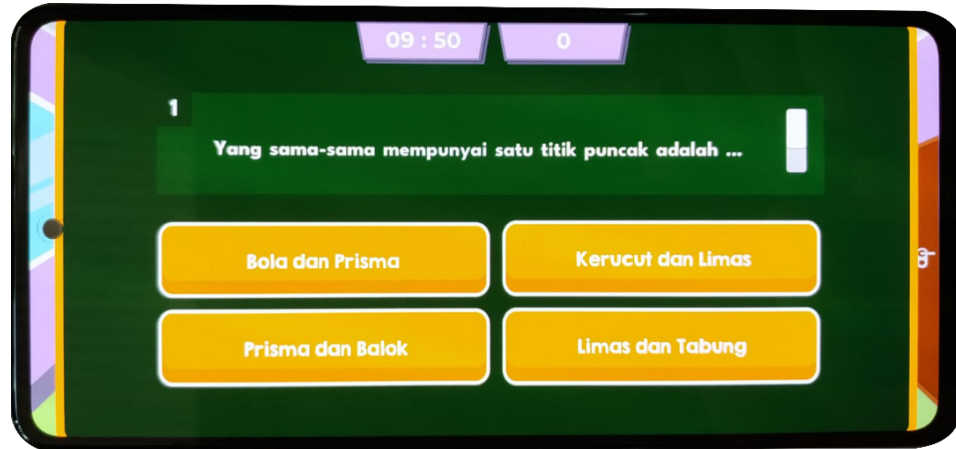
8.     }
9.
10.  }
11.  public class LINK : MonoBehaviour

```

Keterangan:

1. Kode 1-6, merupakan kode yang berfungsi untuk membuka url ketika aplikasi sedang dijalankan.

#### h. Tampilan Latihan Soal



```

1.  public class kuis : MonoBehaviour
2.  {
3.      [System.Serializable]
4.      public class Soal{
5.          [TextArea]
6.          [Header("soal")]
7.          public string soal;
8.          [Header("pilihan")]
9.          public string[] pilihan;
10.         [Header("kunci jawaban")]
11.         public bool[] jawaban;
12.         [Header("nilai")]

```



```

13.         public int nilai;
14.     }
15.         public bool random_soal, random_tombol;
16.
17.         public TMP_Text text_soal, text_nomor,
text_nilai, text_jwbbenar, text_jwbsalah,
text_nilaiakhir;
18.         public Button [] jawaban;
19.         public Animator papan;
20.         public GameObject feedback_benar,
feedback_salah, hasil_kuis;
21.         int nilai = 0, jwbbenar = 0, jwbsalah = 0,
nomor = -1, terjawab = 0;
22.
23.         public AudioSource suara_benar,
suara_salah;
24.         Vector3[] posisi_tombol;
25.         public Soal[] soal;
26.         public void savePos_tombol() {
27.
28.             posisi_tombol = new
Vector3[jawaban.Length];
29.             for (int i = 0; i <
posisi_tombol.Length; i++){
30.                 posisi_tombol[i] =
jawaban[i].transform.position;
31.             }
32.         }
33.         public void random_button() {
34.             bool[] pos = new
bool[posisi_tombol.Length];
35.             for (int i = 0; i < pos.Length; i++) {
36.                 pos[i] = false;
37.             }
38.
39.             int pos_random;
40.             for (int i = 0; i < pos.Length; i++) {

```

```

41.         do{
42.             pos_random =
43.             (int)Random.Range(0, pos.Length-0.6f);
44.         } while (pos[pos_random]);
45.             pos[pos_random] = true;
46.         jawaban[i].transform.position =
47.         posisi_tombol[pos_random];
48.     }
49.     public void set_soal()
50.     {
51.         if (random_soal){
52.             nomor =
53.             (int)Random.Range(0,soal.Length-0.6f);
54.             if (terjawab < soal.Length)
55.             {
56.                 if (soal[nomor].soal ==
57.                 "terjawab")
58.                 {
59.                     set_soal();
60.                 }
61.                 else
62.                 {
63.                     text_soal.text =
64.                     soal[nomor].soal;
65.                     for (int i = 0; i <
66.                     jawaban.Length; i++)
67.                     {
68.                         jawaban[i].GetComponent<Animator>().Play(0);
69.                         jawaban[i].transform.GetChild(0).GetComponent<TMP_Text>().text = soal[nomor].pilihan[i];
70.                     }
71.                     soal[nomor].soal =
72.                     "terjawab";

```

```
68.             terjawab++;
69.             text_nomor.text =
    terjawab.ToString();
70.             if (random_tombol) {
71.                 random_button();
72.             }
73.         }
74.     }
75.     else{
76. papan.gameObject.SetActive(false);
77.         hasil_kuis.SetActive(true);
78.     }
79. }
80. else
81. {
82.     nomor++;
83.     if (nomor < soal.Length)
84.     {
85.         papan.Play(0);
86.         text_soal.text =
    soal[nomor].soal;
87.         for (int i = 0; i <
    jawaban.Length; i++)
88.         {
89. jawaban[i].GetComponent<Animator>().Play(0); /
90. jawaban[i].transform.GetChild(0).GetComponent<T
    MP_Text>().text = soal[nomor].pilihan[i];
91.         }
92.         text_nomor.text =
    (nomor+1).ToString();
93.         if (random_tombol) {
94.             random_button();
95.         }
```

```
100.         }
101.         else
102.         {
103.     papan.gameObject.SetActive(false);
104.         hasil_kuis.SetActive(true);
105.         }
106.     }
107. }
108.     public void jawab(int index) {
109.         bool jawaban_benar = false;
110.         for(int i = 0; i < jawaban.Length;
111. i++){
111.             if(index == i &&
soal[nomor].jawaban[i]){
112.                 jawaban_benar = true;
113.                 i = jawaban.Length;
114.             }
115.         }
116.         if (jawaban_benar){
117.             suara_benar.Play();
118.             feedback_benar.SetActive(true);
119.     feedback_benar.GetComponent<Animator>().Play(0)
;
120.             nilai += soal[nomor].nilai;
121.             jwbbenar++;
122.         if (jawaban_benar){
123.             suara_benar.Play();
124.             feedback_benar.SetActive(true);
125.     feedback_benar.GetComponent<Animator>().Play(0)
;
126.             nilai += soal[nomor].nilai;
127.             jwbbenar++;
128.             jwbsalah--;
```

```

129.         } else{
130.             suara_salah.Play();
131.             feedback_salah.SetActive(true);

132.     feedback_salah.GetComponent<Animator>().Play(0)
        ;
133.         }
134.     }

135.     public void output(){
136.         text_nilai.text = nilai.ToString();
137.         text_jwbbenar.text =
jwbbenar.ToString();
138.         text_jwbsalah.text =
jwbsalah.ToString();
139.         text_nilaiakhir.text =
nilai.ToString();
140.     }

141.     void Update(){
142.         output();
143.     }

144.     void Start(){
145.         savePos_tombol();
146.         set_soal();
147.         jwbsalah = soal.Length;
148.     }

189.     }

```

**Keterangan:**

1. Kode 4-17, merupakan kode yang berfungsi untuk menampilkan fungsi pada unity.
2. Kode 19, merupakan kode yang berfungsi untuk menjalankan animasi papan untuk setiap soal.
3. Kode 21, merupakan kode yang berfungsi sebagai variabel untuk menyimpan value.
4. Kode 23, merupakan kode yang berfungsi sebagai tempat suara benar dan suara

salah yang dimunculkan ketika kuis berhasil dijawab.

5. Kode 26-48, merupakan kode yang berfungsi mengatur tombol dan soal supaya teracak.
6. Kode 49-104, merupakan kode berfungsi untuk mengatur keseluruhan soal yang teracak maupun soal yang tidak teracak, mengatur animasi papan dijalankan secara otomatis.
7. Kode 105-131, merupakan kode yang berfungsi untuk mengatur jawaban yang dipilih. Menjalankan kondisi ketika jawaban yang dipilih benar maka secara otomatis suara jawaban benar akan di jalankan di unity dan ketika jawaban salah maka secara otomatis suara jawaban salah akan dijalanankan di unity.
8. Kode 132-136, merupakan kode yang berfungsi untuk menampilkan nilai yang diubah kedalam string karena nilai awal yaitu int.
9. Kode 141-144, merupakan kode yang berfungsi untuk awal memulai kuis, soal harus diatur untuk pertama kalinya.

```
1. public class timer : MonoBehaviour
2. {
3.     public float menit, detik;
4.     public GameObject papan, hasil_kuis;
5.
6.     // Update is called once per frame
7.     void Update()
8.     {
9.         if (papan.activeSelf)
10        {
11            if (detik >= 0)
12            {
13                detik -= Time.deltaTime;
14            }
15            else
16            {
17                if (menit > 0)
```

```
18         {
19             detik = 60;
20             menit--;
21         }
22         else
23         {
24             papan.SetActive(false);
25             hasil_kuis.SetActive(true);
26         }
27     }
28 }
29 string menit_, detik_;
30
31 if (menit < 10)
32 {
33     menit_ = "0" + ((int)menit).ToString();
34 }
35 else
36 {
37     menit_ = ((int)menit).ToString();
38 }
39
40 if (detik < 10)
41 {
42     detik_ = "0" + ((int)detik).ToString();
43 }
44 else
45 {
46     detik_ = ((int)detik).ToString();
47 }
48 GetComponent<TMP_Text>().text = menit_ + " :
49 " + detik_;
```

Keterangan:

1. Kode 1-51, merupakan kode yang berfungsi untuk membuat, mengatur waktu selama kuis sedang dijalankan. *Deltatime* adalah putaran waktu dalam satu detik pada aplikasi unity. Beberapa kondisi diterapkan supaya meminimalisir terjadinya kesalahan.

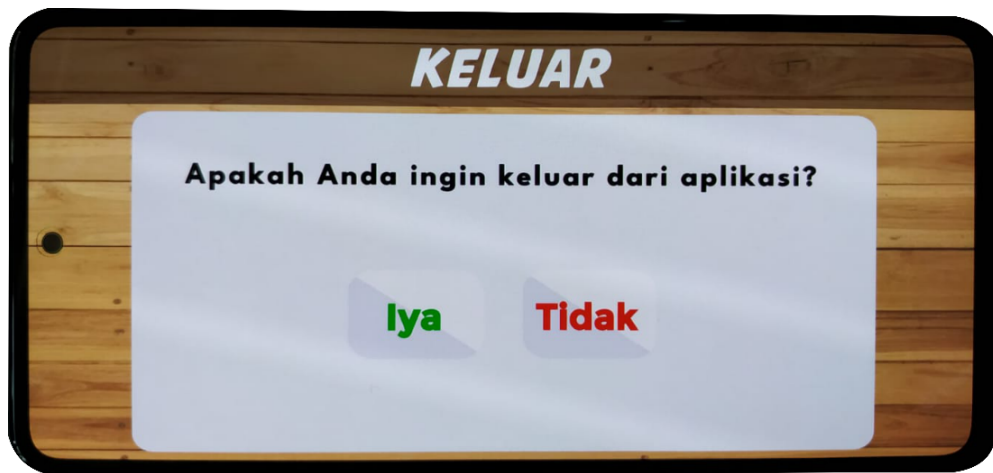
```
1. public class next_soal : MonoBehaviour
2. {
3.     public kuis kuis;
4.
5.     public void nextsoal(){
6.         kuis.set_soal();
7.     }
8. }
```

Keterangan:

1. Kode 1-8, merupakan kode yang berfungsi untuk berpindah soal ke soal yang lain melalui animasi yang sudah dibuat menggunakan animasi *event* untuk menghubungkannya.



## i. Tampilan Keluar Aplikasi



```
1. public void quit_app()
2.     {
3.         #if (UNITY_EDITOR)
4.             UnityEditor.EditorApplication.isPlaying =
5.             false;
6.         #elif (UNITY_STANDALONE || UNITY_ANDROID)
7.             Application.Quit();
8.         #elif (UNITY_WEBGL)
9.             Application.OpenURL("about:blank");
10        #endif
11    }
12
13 public void quit_app()
14     {
15         #if (UNITY_EDITOR)
16             UnityEditor.EditorApplication.isPlaying =
17             false;
18         #elif (UNITY_STANDALONE || UNITY_ANDROID)
19             Application.Quit();
20         #elif (UNITY_WEBGL)
21             Application.OpenURL("about:blank");
22         #endif
23    }
```

Keterangan:

1. Kode 1-6, merupakan kode yang berfungsi untuk keluar dari aplikasi. Tombol keluar digunakan pada unity editor, unity standalone atau unity android dan unity webgl.