

# LAMPIRAN



**SURAT KEPUTUSAN**  
**REKTOR IBI DARMAJAYA**  
**NOMOR : SK.0262/DMJ/DFIK/BAAK/V-17**  
Tentang  
**Dosen Pembimbing Skripsi**  
**Program Studi S1 Teknik Informatika**  
**REKTOR IBI DARMAJAYA**

- Memperhatikan : 1. Bahwa dalam rangka usaha peningkatan mutu dan peranan IBI Darmajaya dalam melaksanakan Pendidikan Nasional perlu ditingkatkan kemampuan mahasiswa dalam Skripsi.
- Menimbang : 1. Laporan dan usulan Ketua Program Studi **S1 Teknik Informatika**.  
2. Bahwa untuk mengefektifkan tenaga pengajar dalam Skripsi mahasiswa perlu ditetapkan **Dosen Pembimbing Skripsi**.
- Mengingat : 1. UU No.20 Tahun 2003 Tentang Sistem Pendidikan Nasional.  
2. Peraturan Pemerintah No.60 Tahun 2010 tentang Pendidikan Sekolah Tinggi  
3. Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia No.165/D/0/2008 tertanggal 20 Agustus 2008 tentang Perubahan Status STMIK-STIE Darmajaya menjadi Informatics and Business Institute (IBI) Darmajaya  
4. STATUTA IBI Darmajaya  
5. Surat Ketua Yayasan Pendidikan Alfian Husin No. IM.003/YP-AH/X-08 tentang Persetujuan Perubahan Struktur Organisasi  
6. Surat Keputusan Rektor 0383/DMJ/REK/X-08 tentang Struktur Organisasi.
- Menetapkan**
- Pertama : Mengangkat nama-nama seperti tersebut dalam lampiran Surat Keputusan ini sebagai Dosen Pembimbing Skripsi mahasiswa Program Studi S1 Teknik Informatika.
- Kedua : Pembimbing Skripsi berkewajiban melaksanakan tugasnya sesuai dengan jadwal yang telah ditetapkan.
- Ketiga : Pembimbing Skripsi yang ditunjuk akan diberikan honorarium yang besarnya sesuai dengan ketentuan peraturan dan norma penggajian dan honorarium IBI Darmajaya.
- Keempat : Surat Keputusan ini berlaku sejak tanggal ditetapkan dan apabila dikemudian hari terdapat kekeliruan dalam keputusan ini, maka keputusan ini akan ditinjau kembali.

Ditetapkan di : Bandar Lampung  
Pada tanggal : 22 Mei 2017  
Rektor IBI Darmajaya,  
Dekan Fakultas Ilmu Komputer



15	Qahar Genti Dwi Mulya	1211010046	Rancang Bangun Aplikasi Perencanaan Kala Seropang Asing Dalam Bahasa Indonesia Berbasis Android Menggunakan Metode Sequential	Fitria, S.T, M.Kom
16	Philip Andika Purba	1211010113	Metode Case Based Reasoning Untuk Menentukan Penempatan Guru Bantu Pada Daerah Tertinggal Di Provinsi Lampung	
17	Mochammad Adhyatma Cakri	1311010042	Rancang Bangun Aplikasi Tutorial Daun Ulang Sampah dan Limbah Berbasis Android	Haryanto Wibowo, S.Kom, M.T.I
18	Ardi Yansyah	1311010057	Rancang Bangun Aplikasi Pemeliharaan Sistem Injeksi Pada Sepeda Motor Honda Beat Berbasis Android	
19	Muhlisin	1311010120	Rancang Bangun Aplikasi Mobile Delivery Order Pada Simetra Digital Printing	Ketut Ardaye, S.Kom, M.T.I
20	Indra Sahari	1311010046	Portal Informasi Perusahaan Internal PT Indonusa Telemedia (Transvision) Berbasis Mobile	
21	I Made Adhitya Wijaya	1211010149	Rancang Bangun Media Pembelajaran Bahasa Kawi Berbasis Android Pada Sekolah Tinggi Agama Hindu (STAH)	Puput Budi Wintoro, S.Kom M.T.I
22	Yogi Murti 609	1311010062	Perancangan Sistem Kelayakan Naskah Untuk Buku Anak Dengan Algoritma Boyer - Moore	
23	Wayan Andre Naryana	1211010085	Klasifikasi Data Nasaabah Pada Koperasi Artha Jaya Menggunakan Metode Naive Bayes Classification	Rahmalia Syahputri, S.Kom M.Eng.Sc
24	Febri Maulana	1211010077	Visualisasi Penodelan Gedung dan Animasi Museum Lampung Berbasis 3D	
25	Raka Tamagola	1311010077	Visualisasi 3D Pengenalan Aset Kendaraan Tempur Darat Dan Laut Militer Brigadir Infanteri 3 Korps Marinir (TNI AL) Lampung Berbasis Android	Handi Yuga
26	Irvan Rimhot Sijabat	1311010102	Rancang Bangun Sistem Work Order Berbasis Mobile Pada PT. Indomarco Prismatama Cabang Lampung	
27	Kurnia Ahmad Aziz	1211010107	Visualisasi 3D Profil IBI Darmajaya Sebagai Media Promosi Internasional	Handi Yuga
28	Handi Yuga	1111010097	Perancangan Media Informasi Pusat Bahasa dan Pelatihan IBI Darmajaya Berbasis Website	

---

### Changescene

---

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ChaneScene :
MonoBehaviour {

    public void
chanemenuscene(string scenename)
    {
        Application.LoadLevel(scenename);
    }

    public void keluar()
    {
        Application.Quit();
    }
}
```

---

### Layananlogic

---

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class LayananLogic :
MonoBehaviour {

    public GameObject panelPenumpang;
    public GameObject panelBarang;
    public GameObject panelAset;

    // Use this for initialization
    void Start () {
        panelPenumpang.SetActive(true);
        panelBarang.SetActive(false);
        panelAset.SetActive(false);
    }

    public void PenumpangClicked()
```

```
{
    panelPenumpang.SetActive(true);
    panelBarang.SetActive(false);
    panelAset.SetActive(false);
}

public void BarangClicked()
{
    panelPenumpang.SetActive(false);
    panelBarang.SetActive(true);
    panelAset.SetActive(false);
}

public void AsetClicked()
{
    panelPenumpang.SetActive(false);
    panelBarang.SetActive(false);
    panelAset.SetActive(true);
}
}
```

---

### UserInterfaceButton

---

```
using UnityEngine;
using System.Collections;
using System.IO;

public class UserInterfaceButtons :
MonoBehaviour
{
    public float scalingSpeed =
0.0000000001f;
    public float rotationSpeed = 70.0f;
    public float translationSpeed =
5.0f;
    //
    public GameObject Model;
    bool repeatScaleUp = false;
    bool repeatScaleDown = false;
    bool repeatRotateLeft = false;
    bool repeatRotateRight = false;
    bool repeatPositionUp = false;
    bool repeatPositionDown = false;
    bool repeatPositionLeft = false;
    bool repeatPositionRight = false;

    void Update ()
    {
        if (repeatScaleUp) {
```



```

        ScaleUpButton ();
    }
    if (repeatScaleDown) {
ScaleDownButton ();
    }
    if (repeatRotateRight) {
RotationRightButton();
    }
    if (repeatRotateLeft) {
RotationLeftButton();
    }
    if (repeatPositionUp) {
PositionUpButton();
    }
    if (repeatPositionDown) {
PositionDownButton();
    }
    if (repeatPositionLeft) {
PositionLeftButton();
    }
    if (repeatPositionRight) {
PositionRightButton();
    }
}
public void CloseAppButton ()
{
    Application.Quit ();
}

```

```

    public void RotationRightButton
    ()
    {
        // transform.Rotate (0, -
rotationSpeed * Time.deltaTime, 0);
        GameObject.FindWithTag
("Model").transform.Rotate (0, -
rotationSpeed * Time.deltaTime, 0);
    }
    public void RotationLeftButton ()
    {
        // transform.Rotate (0,
rotationSpeed * Time.deltaTime, 0);
        GameObject.FindWithTag
("Model").transform.Rotate (0,
rotationSpeed * Time.deltaTime, 0);
    }
    public void
RotationRightButtonRepeat ()
    {
        // transform.Rotate (0, -
rotationSpeed * Time.deltaTime, 0);
repeatRotateRight=true;
    }
    public void
RotationLeftButtonRepeat ()
    {
        // transform.Rotate (0,
rotationSpeed * Time.deltaTime, 0);
repeatRotateLeft=true;
    }
    public void ScaleUpButton ()
    {
        // transform.localScale +=
new Vector3(scalingSpeed, scalingSpeed,
scalingSpeed);
        GameObject.FindWithTag
("Model").transform.localScale += new

```

```

Vector3 (scalingSpeed, scalingSpeed,
scalingSpeed);
    }

    public void ScaleUpButtonRepeat
()
    {
        repeatScaleUp = true;
        Debug.Log ("Up");
    }
    public void
ScaleDownButtonRepeat ()
    {
        repeatScaleDown = true;
        Debug.Log ("Down");
    }
    public void
PositionDownButtonRepeat ()
    {
        repeatPositionDown =
true;
    }
    public void
PositionUpButtonRepeat ()
    {
        repeatPositionUp = true;
    }
    public void
PositionLeftButtonRepeat ()
    {
        repeatPositionLeft = true;
    }
    public void
PositionRightButtonRepeat ()
    {
        repeatPositionRight =
true;
    }

    public void ScaleUpButtonOff ()
    {
        repeatScaleUp = false;
        Debug.Log ("Off");
    }
    public void ScaleDownButtonOff
()
    {
        repeatScaleDown = false;
        Debug.Log ("Off");
    }

    public void RotateLeftButtonOff
()
    {
        repeatRotateLeft = false;
        Debug.Log ("Off");
    }

    public void RotateRightButtonOff
()
    {
        repeatRotateRight = false;
        Debug.Log ("Off");
    }
    public void
PositionRightButtonOff ()
    {
        repeatPositionRight =
false;
        Debug.Log ("Off");
    }
    public void PositionLeftButtonOff
()
    {
        repeatPositionLeft =
false;
        Debug.Log ("Off");
    }
    public void PositionUpButtonOff
()
    {
        repeatPositionUp = false;
        Debug.Log ("Off");
    }
    public void
PositionDownButtonOff ()
    {
        repeatPositionDown =
false;
        Debug.Log ("Off");
    }
}

```

```
public void ScaleDownButton ()
{
    // transform.localScale +=
new Vector3(-scalingSpeed, -
scalingSpeed, -scalingSpeed);

    GameObject.FindWithTag
("Model").transform.localScale += new
Vector3 (-scalingSpeed, -scalingSpeed, -
scalingSpeed);
}

public void PositionUpButton ()
{

    GameObject.FindWithTag
("Model").transform.Translate (0, 0, -
translationSpeed * Time.deltaTime);
}

public void PositionDownButton
()
{

    GameObject.FindWithTag
("Model").transform.Translate (0, 0,
translationSpeed * Time.deltaTime);
}

public void PositionRightButton
()
{

    GameObject.FindWithTag
("Model").transform.Translate (-
translationSpeed * Time.deltaTime, 0, 0);
}

public void PositionLeftButton ()
{

    GameObject.FindWithTag
("Model").transform.Translate
(translationSpeed * Time.deltaTime, 0, 0);
// backward
}
```

a)

```
public void ChangeScene (string
{
    Application.LoadLevel

}

public void AnyButton ()
{
    Debug.Log ("Any");
}
```

(a);