

## DAFTAR PUSTAKA

- Amrulloh, M. F., Febriansyah, M. (2021). Aplikasi Pengenalan Alat Fiber Optik Menggunakan *Augmented reality*. <https://jurnal.stmik-yadika.ac.id/index.php/spirit/article/view/219>
- Arkadiantika, I., Ramansyah, W., Effindi, M. A., Dellia, P. (2020). Pengembangan Media Pembelajaran *Virtual Reality* pada Materi Pengenalan Termination dan Splicing *Fiber optic*. Universitas Trunojoyo Madura.
- BYJU'S. (2024). Uses of Rectifier. <https://byjus.com/physics/uses-of-rectifier/>
- Choiri, E. O. (2023, August 11). Apa Itu Kabel Fiber Optik, Fungsi, Jenis dan Kelebihannya. <https://qwords.com/blog/apa-itu-fiber-optik/>
- EverythingRF. (2017, July 26). What are Sector Antennas? <https://www.everythingrf.com/community/what-are-sector-antennas>
- Fibconet. (2023, October 30). Apa Itu Kabel Patch Serat Optik. <https://fibconet.com/id/what-is-fiber-optic-patch-cord/>
- Hidayati, L. (2020). Analisa Kualitas Jaringan 4G LTE untuk Provider H31 Berdasarkan Parameter Drive Test Menggunakan Software Genex Probe 5.1 di Kota Purwokerto. Semarang. Universitas Semarang.
- iFORTE (2023, October 11). Mengenal Apa itu Jaringan Fiber Optik, Fungsi, Kelebihan dan Kekurangan. <https://www.iforte.id/news/detail/jaringan-fiber-optik>
- Kamila, M. (2023). Rancang Bangun Modul Praktikum “Simulasi Instalasi Microwave Link Pada Sistem Seluler” Berbasis Teknologi *Virtual Reality*. Makassar: Universitas Hasanuddin.
- Lowood, H. E. (2024, February 25). *Virtual Reality*. Encyclopaedia Britannica. <https://www.britannica.com/technology/virtual-reality>
- Rachman, A. H. A., Syafruddin, R. (2018). Analisis Total Harmonik Distorsi pada Panel ACPDB Akibat Beban Non Linier. STT MANDALA. Bandung.
- Radiall. (2016, November 11). What is RRU in Telecom? <https://www.radiall.com/fr/insights/what-is-rru-in-telecom>
- Roedavan, R. (2021). Pembuatan Aplikasi Ensiklopedia Digital dan Pemrograman Game. Bandung: Informatika.
- Rouse, Margaret. (2019, August 5). Baseband Unit. Techopedia. <https://www.techopedia.com/definition/33908/baseband-unit-bbu>
- Ruiyu, Z. (2023, July 8). What Is an IPRAN? Huawei. <https://info.support.huawei.com/info-finder/encyclopedia/en/IPRAN.html>
- Sukaridhoto, S., Fajrianti, E., Haz, A., Budiarti, R. P. N., Agustien, L. (2023). Implementation of *Virtual Fiber optic* Module Using *Virtual Reality* for Vocational Telecommunications Students. Retrieved from <https://joiv.org/index.php/joiv/article/view/1361>
- Sulistiawan, Y. Y. (2020). Panduan Aplikasi *Virtual Reality* (VR). Jakarta: PT Elex Media Komputindo.
- UK Air Comms. (2022, July 15). What exactly is inside a Telecoms Site? <https://ukaircomms.co.uk/inside-a-telecoms-site/>
- Uma, Bamai. (2022, December 12). Apa Itu Blender 3D? Ini Kelebihan dan Kekurangannya. Biro Administrasi Mutu Akademik dan Informasi Universitas Medan Area. <https://bamai.uma.ac.id/2022/12/12/apa-itu-blender-3d-ini-kelebihan-dan-kekurangannya/>

- Upadhyay, B., Madathil, K. C., Bertrand, J., Rosopa, E., Ballato, J., Verlage, S. S., Kimerling, L. C., Gramopadhye, A. (2023, November 1). *Virtual Reality* enabled asynchronous learning modules for *fiber optic* preform manufacturing education. <https://pubmed.ncbi.nlm.nih.gov/38037917/>
- Zhou, R. B., Jin, Zhangyun. Design of Virtual Training System for *Virtual Reality* Intelligent Production Line of Optical Fiber Router Based on Industrial Robot. <https://ieeexplore.ieee.org/document/9700038>

## LAMPIRAN

### Lampiran 1 *Script*

#### 1. ObjectActiveAll.cs

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

public class ObjectActiveAll : MonoBehaviour
{
    public GameObject[] objectsToActivate; // Objek yang akan
    diaktifkan
    public GameObject buttonToShow; // Tombol yang akan ditampilkan
    public GameObject infocanva;
    //public GameObject toggleCanva;

    private void Update()
    {
        // Periksa apakah semua objek telah diaktifkan
        bool allActivated = true;
        bool offCanva = false;
        foreach (GameObject obj in objectsToActivate)
        {
            if (!obj.activeSelf)
            {
                allActivated = false;
                //offCanva = true;
                break;
            }
        }

        // Tampilkan tombol jika semua objek telah diaktifkan
        buttonToShow.gameObject.SetActive(allActivated);

        infocanva.gameObject.SetActive(allActivated);

        //toggleCanva.gameObject.SetActive(offCanva);
    }
}
```

#### 2. ManageScene.cs

```
using System.Collections;
```

```

using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Events;
using UnityEngine.SceneManagement;

public class ManageScene : MonoBehaviour
{
    public GameObject objectTrigger;

    public UnityEvent onPressed, onReleased;
    public string namaScene;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }

    public void OnTriggerEnter(Collider other)
    {
        if (other.gameObject == objectTrigger)
        {
            //DontDestroyOnLoad(objectToKeep);
            SceneManager.LoadScene(namaScene);
        }
        //Debug.Log("Pressed Trigger");
    }
    public void OnTriggerExit(Collider other)
    {
        if (other.gameObject == objectTrigger)
        {
            onReleased.Invoke();
            //Debug.Log("Pressed Trigger");
        }
    }
}

```

### 3. DelayStart.cs

```

using System.Collections;
using System.Collections.Generic;

```

```

using UnityEngine;

public class DelayStart : MonoBehaviour
{
    public GameObject delayTimeStart;

    public float timeDelay = 2f;
    // Start is called before the first frame update
    void Start()
    {
        StartCoroutine(PlayShow());
    }
    IEnumerator PlayShow()
    {
        //delayTime.SetActive(false);
        yield return new WaitForSeconds(timeDelay);
        delayTimeStart.SetActive(true);
    }
}

```

#### 4. DelayEnd.cs

```

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

public class DelayEnd : MonoBehaviour
{
    public GameObject delayTime;
    public float timeDelay = 2f;
    // Start is called before the first frame update
    void Start()
    {
        StartCoroutine(PlayShow());
    }

    IEnumerator PlayShow()
    {
        //delayTime.SetActive(false);
        yield return new WaitForSeconds(timeDelay);
        delayTime.SetActive(false);
    }
}

```

#### 5. SnapToLocation.cs

```

using System.Collections;

```

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

public class SnapToLocation : MonoBehaviour
{
    //boolean variable used to determine if the object is currently
    //being held by the player
    private bool grabbed;

    //return true when the object is within the Snapzone radius
    private bool insideSnapZone;

    //Return true when the object has had it's location update
    public bool Snapped;

    //Set the specific part we want to snap to this location
    public GameObject ObjectPart;
    //Reference another object we can use to set rotation
    public GameObject SnapRotationReference;

    //Detects when the RocketPart game object has entered the snap
    //zone radius
    public void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.name == ObjectPart.name)
        {
            insideSnapZone = true;
            Debug.Log("TriggerOn");
        }
    }
    //Detects when the Objectpart game object has left the snap zone
    //radius
    public void OnTriggerExit(Collider other)
    {
        if (other.gameObject.name == ObjectPart.name)
        {
            insideSnapZone = false;
            Debug.Log("TriggerOff");
        }
    }

    void SnapObject()
    {
        if (grabbed == false && insideSnapZone == true)
```

```

        {
            ObjectPart.gameObject.transform.position =
transform.position;
            ObjectPart.gameObject.transform.rotation =
SnapRotationReference.transform.rotation;
            Snapped = true;
        }
    }
    // Update is called once per frame
    void Update()
    {
        //grabbed = ObjectPart.GetComponent<OVRGrabbable>().isGrabbed;

        grabbed =
ObjectPart.GetComponent<Grabbable>().TransferOnSecondSelection;

        SnapObject();
    }
}

```

## 6. SnapPoint.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Oculus.Interaction;

public class SnapPoint : MonoBehaviour
{
    //Reference the snap zone collider we'll be using
    //public GameObject SnapLocation;

    //Reference the game object that the snapped objects will become a
part of
    public GameObject ObjectToSnap;

    //Create a variable that will be used by the Object Lunch script
to determine if all their pieces
    public bool isSnapped;

    //boolean variable used to reference the "Snapped" boolean from
the SnapToLocation script
    private bool objectSnapped;

    //boolean variable used to determine if the object is currently
being held by the player
    private bool grabbed;
}

```

```

// Start is called before the first frame update
void Start()
{

}

// Update is called once per frame
void Update()
{
    //set grabbed to equal the boolean value "isGrabbed" from the
OVRGrabbable script
    //grabbed = GetComponent<OVRGrabbable>().isGrabbed;
    grabbed = GetComponent<Grabbable>().TransferOnSecondSelection;

    //Set objectSnapped equal to the Snapped boolean from
SnapToLocation
    //objectSnapped =
SnapLoction.GetComponent<SnapToLocation>().Snapped;

    if (objectSnapped == true)
    {
        GetComponent<Rigidbody>().isKinematic = true;
        //transform.SetParent(ObjectToSnap.transform);
        isSnapped = true;
        transform.SetParent(ObjectToSnap.transform);
    }

    if (objectSnapped == false && grabbed == true)
    {
        GetComponent<Rigidbody>().isKinematic = true;
        GetComponent<Rigidbody>().useGravity = false;
        transform.SetParent(null);
    }
}
}
}

```

## 7. ObjectPakai.cs

```

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

public class ObjectPakai : MonoBehaviour
{
    public GameObject objectUse;
}

```



```

public GameObject objectTriggerTo;

public void OnTriggerEnter(Collider other)
{
    if (other.gameObject == objectTriggerTo)
    {
        objectUse.SetActive(false);
    }
}
}

```

#### 8. ActiveTriggerPoint.cs

```

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

public class ActiveTriggerPoint : MonoBehaviour
{
    public GameObject pointOnTrigger;
    public bool activePoint;
    public GameObject NextActive;

    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
    }

    public void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.name == pointOnTrigger.name)
        {
            activePoint = true;
            //Debug.Log("Pressed Trigger");
        }
    }

    public void OnTriggerExit(Collider other)
    {
        if (other.gameObject.name == pointOnTrigger.name)
        {

```

```

        activePoint = false;
        //Debug.Log("Pressed Trigger");
    }
}
}

```

## 9. PointOnRunning.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Events;

public class PointOnRunning : MonoBehaviour
{
    public GameObject[] pointOnActive;
    private bool pointAdd;
    private bool running = false;

    public UnityEvent onPressed, onReleased;
    void Update()
    {
        if (componentLaunch() == true && running == false)
        {
            onPressed?.Invoke();
            //Debug.Log(componentLaunch())

            running = true;
        }

        if (componentLaunch() == false && running == true)
        {
            onReleased?.Invoke();
            //Debug.Log(componentLaunch());

            running = false;
        }
    }

    private bool componentLaunch()
    {
        for (int i = 0; i < pointOnActive.Length; i++)
        {

            pointAdd =
pointOnActive[i].GetComponent<ActiveTriggerPoint>().activePoint;

```

```

        if (pointAdd == false)
        {
            return false;
        }
    }
    return true;
}
}

```

#### 10. CableComponent.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEditor;

public class CableComponent : MonoBehaviour
{
    #region Class members

    [SerializeField] private Transform endPoint;
    [SerializeField] private Material cableMaterial;

    // Cable config
    [SerializeField] private float cableLength = 0.5f;
    [SerializeField] private int totalSegments = 5;
    [SerializeField] private float segmentsPerUnit = 2f;
    private int segments = 0;
    [SerializeField] private float cableWidth = 0.1f;

    // Solver config
    [SerializeField] private int verletIterations = 1;
    [SerializeField] private int solverIterations = 1;

    //[Range(0,3)]
    [SerializeField] private float stiffness = 1f;

    private LineRenderer line;
    private CableParticle[] points;

    #endregion
    #region Initial setup

    void Start()
    {
        InitCableParticles();
        InitLineRenderer();
    }
}

```

```

}
/**
 * Init cable particles
 *
 * Creates the cable particles along the cable length
 * and binds the start and end tips to their respective game
objects.
 */
void InitCableParticles()
{
    // Calculate segments to use
    if (totalSegments > 0)
        segments = totalSegments;
    else
        segments = Mathf.CeilToInt (cableLength * segmentsPerUnit);

    Vector3 cableDirection = (endPoint.position -
transform.position).normalized;
    float initialSegmentLength = cableLength / segments;
    points = new CableParticle[segments + 1];

    // Foreach point
    for (int pointIdx = 0; pointIdx <= segments; pointIdx++) {
        // Initial position
        Vector3 initialPosition = transform.position +
(cableDirection * (initialSegmentLength * pointIdx));
        points[pointIdx] = new CableParticle(initialPosition);
    }

    // Bind start and end particles with their respective
gameobjects
    CableParticle start = points[0];
    CableParticle end = points[segments];
    start.Bind(this.transform);
    end.Bind(endPoint.transform);
}

/**
 * Initialized the line renderer
 */
void InitLineRenderer()
{
    line = this.gameObject.AddComponent<LineRenderer>();
    line.SetWidth(cableWidth, cableWidth);
    line.SetVertexCount(segments + 1);
    line.material = cableMaterial;
    line.GetComponent<Renderer>().enabled = true;
}

```

```

}

#endregion

#region Render Pass

void Update()
{
    RenderCable();
}

/**
 * Render Cable
 *
 * Update every particle position in the line renderer.
 */
void RenderCable()
{
    for (int pointIdx = 0; pointIdx < segments + 1; pointIdx++)
    {
        line.SetPosition(pointIdx, points [pointIdx].Position);
    }
}

#endregion

#region Verlet integration & solver pass

void FixedUpdate()
{
    for (int verletIdx = 0; verletIdx < verletIterations;
verletIdx++)
    {
        VerletIntegrate();
        SolveConstraints();
    }
}

/**
 * Verler integration pass
 *
 * In this step every particle updates its position and speed.
 */
void VerletIntegrate()
{

```

```

    Vector3 gravityDisplacement = Time.fixedDeltaTime *
Time.fixedDeltaTime * Physics.gravity;
    foreach (CableParticle particle in points)
    {
        particle.UpdateVerlet(gravityDisplacement);
    }
}

/**
 * Constrains solver pass
 *
 * In this step every constraint is addressed in sequence
 */
void SolveConstraints()
{
    // For each solver iteration..
    for (int iterationIdx = 0; iterationIdx < solverIterations;
iterationIdx++)
    {
        SolveDistanceConstraint();
        SolveStiffnessConstraint();
    }
}

#endregion

#region Solver Constraints

/**
 * Distance constraint for each segment / pair of particles
 */
void SolveDistanceConstraint()
{
    float segmentLength = cableLength / segments;
    for (int SegIdx = 0; SegIdx < segments; SegIdx++)
    {
        CableParticle particleA = points[SegIdx];
        CableParticle particleB = points[SegIdx + 1];

        // Solve for this pair of particles
        SolveDistanceConstraint(particleA, particleB,
segmentLength);
    }
}

/**

```

```

    * Distance Constraint
    *
    * This is the main constrains that keeps the cable particles
    "tied" together.
    */
    void SolveDistanceConstraint(CableParticle particleA,
    CableParticle particleB, float segmentLength)
    {
        // Find current vector between particles
        Vector3 delta = particleB.Position - particleA.Position;
        //
        float currentDistance = delta.magnitude;
        float errorFactor = (currentDistance - segmentLength) /
currentDistance;

        // Only move free particles to satisfy constraints
        if (particleA.IsFree() && particleB.IsFree())
        {
            particleA.Position += errorFactor * 0.5f * delta;
            particleB.Position -= errorFactor * 0.5f * delta;
        }
        else if (particleA.IsFree())
        {
            particleA.Position += errorFactor * delta;
        }
        else if (particleB.IsFree())
        {
            particleB.Position -= errorFactor * delta;
        }
    }

    /**
    * Stiffness constraint
    */
    void SolveStiffnessConstraint()
    {
        float distance = (points[0].Position -
points[segments].Position).magnitude;
        if (distance > cableLength)
        {
            foreach (CableParticle particle in points)
            {
                SolveStiffnessConstraint(particle, distance);
            }
        }
    }
}

```

```

/**
 * TODO: I'll implement this constraint to reinforce cable
stiffness
 *
 * As the system has more particles, the verlet integration
approach
 * may get way too loose cable simulation. This constraint is
intended
 * to reinforce the cable stiffness.
 * // throw new System.NotImplementedException ();
 */
void SolveStiffnessConstraint(CableParticle cableParticle, float
distance)
{

}

#endregion
}

```

#### 11. WireLineRenderer.cs

```

using UnityEngine;

public class WireLineRenderer : MonoBehaviour
{
    [SerializeField] private LineRenderer lineRenderer;
    [SerializeField] private Transform start;
    [SerializeField] private Transform end;

    void Start()
    {
        lineRenderer.positionCount = 2;
    }

    void Update()
    {
        lineRenderer.SetPosition(0, start.position);
        lineRenderer.SetPosition(1, end.position);
    }
}

```

#### 12. CableSpawn.cs

```

using UnityEngine;

public class CableSpawn : MonoBehaviour

```



```
{
    // public GameObject firstObject, secondObject, lineRenderer,
    activatedObject;
    public GameObject firstObject;
    public GameObject secondObject;
    public GameObject lineRenderer;
    public GameObject activatedObject;

    private bool collided = false;
    private bool grabbed;

    void Start()
    {
        if (!collided)
        {
            activatedObject.SetActive(false);
        }
    }

    public void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.name == firstObject.name)
        {
            collided = true;
            // Debug.Log("OnTriggerEnter OK");
        }
    }

    void Update()
    {
        grabbed =
        firstObject.GetComponent<Grabbable>().TransferOnSecondSelection;
        ObjectInstalled();
    }

    void ObjectInstalled()
    {
        if (!grabbed && collided)
        {
            //firstObject.gameObject.transform.position =
            transform.position;
            firstObject.gameObject.transform.rotation =
            secondObject.transform.rotation;
            activatedObject.SetActive(true);
            Destroy(firstObject);
            Destroy(secondObject);
            Destroy(lineRenderer);
        }
    }
}
```

```

        // Debug.Log("ObjectInstalled OK");
    }
}

// [System.Serializable]
// public class ObjectInteraction{
//     public GameObject firstObject;
//     public GameObject secondObject;
//     public GameObject LineRenderer;

//     [Header("Objek yang diaktifkan")]
//     public GameObject activatedObject;
// }
}

```

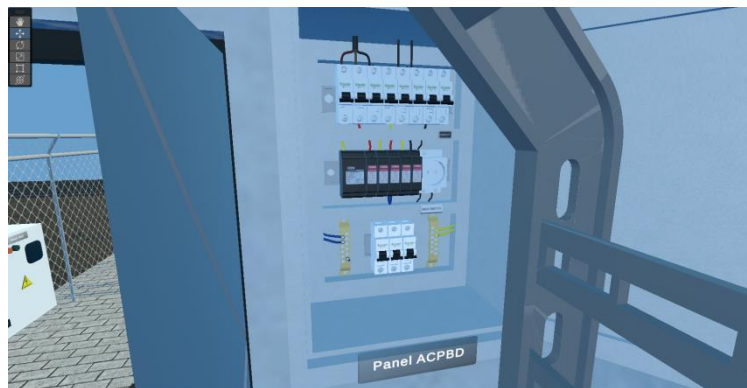
### 13. LiftButton.cs

```

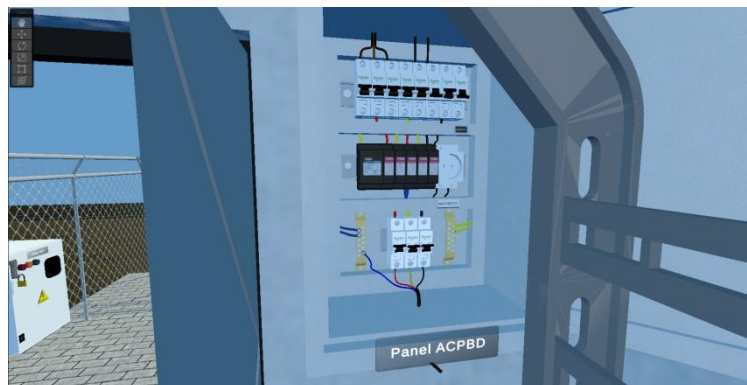
6) using System.Collections;
7) using System.Collections.Generic;
8) using UnityEngine;
9)
10) public class LiftButton : MonoBehaviour
11) {
12)     public GameObject player;
13)     public Vector3 naik;
14)     public Vector3 turun;
15)     public Transform targetPosition;
16)
17)     public void MoveToTargetUp()
18)     {
19)
20)         //transform.position = targetPosition.position;
21)         //player.transform.position = new Vector3(7.961f,
22)         4.398f, 7.158f);
23)         player.transform.position = naik;
24)     }
25)     public void MoveToTargetDown()
26)     {
27)
28)         //transform.position = targetPosition.position;
29)         //player.transform.position = new Vector3(7.02f,
30)         0.22f, 7.34f);
31)         player.transform.position = turun;
32)     }
}

```

## Lampiran 2 Before After Instalasi



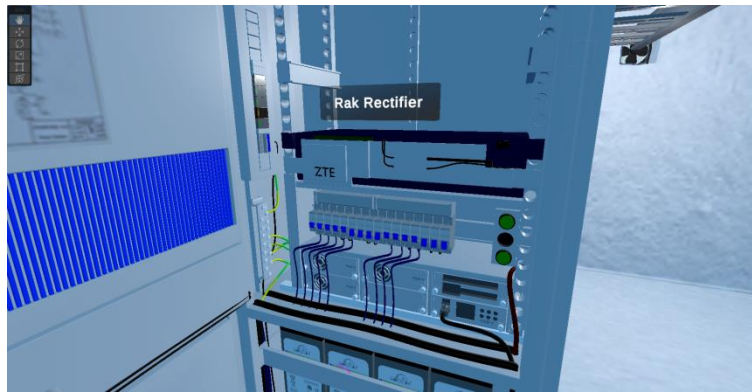
Sebelum instalasi di ACPBD



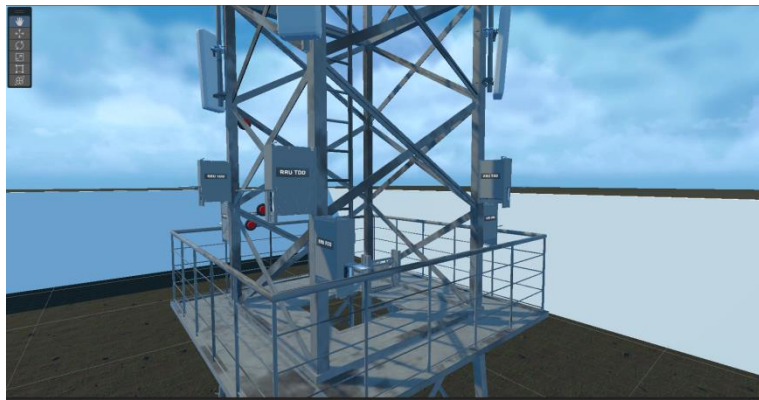
Setelah instalasi di ACPBD



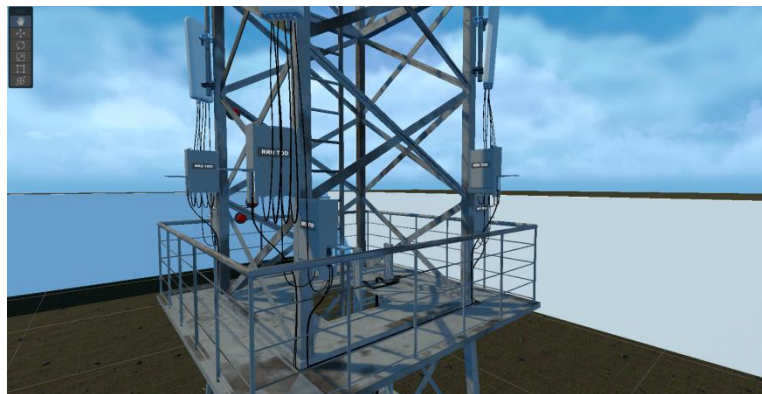
Sebelum instalasi di Rectifier



Setelah instalasi di rectifier




Sebelum instalasi di tower



Setelah instalasi di tower


## Lampiran 3 Hasil Validasi Ahli Materi


BLACK BOX TESTING Report	
Form: Muh Dzulfadli Rusli	
Name	Adil, Setiawan
Pengecekan Komponen dan Peralatan Instalasi	Sesuai
Pemindahan Kabel dari KWH Panel ke ACPDB	Sesuai
Pengerjaan dari ACPDB Sampai Rectifier	Sesuai
Pengerjaan di perangkat BBU	Sesuai
Pengerjaan Kabel dari Shelter Sampai Tower	Sesuai
Pengerjaan di Perangkat Remote Radio Unit (RRU) sampai Antena Sectoral	Sesuai
Pengerjaan "Transmisi Jaringan" di dalam Shelter	Sesuai
Pengerjaan "Transmisi Jaringan" di luar shelter	Sesuai
Rating dari keseluruhan Simulasi Sederhana Ini	5 - ★★★★★
Signature	
Added Time	25-Apr-2024 13:03:10
Referrer Name	
Task Owner	dzulfadlirusi@gmail.com

Hasil Validasi Responden a.n Adil Setiawan

Name	Rizal, Arief
Pengecekan Komponen dan Peralatan Instalasi	Sesuai
Pemindahan Kabel dari KWH Panel ke ACPDB	Sesuai
Pengerjaan dari ACPDB Sampai Rectifier	Sesuai
Pengerjaan di perangkat BBU	Sesuai
Pengerjaan Kabel dari Shelter Sampai Tower	Sesuai
Pengerjaan di Perangkat Remote Radio Unit (RRU) sampai Antena Sectoral	Sesuai
Pengerjaan "Transmisi Jaringan" di dalam Shelter	Sesuai
Pengerjaan "Transmisi Jaringan" di luar shelter	Sesuai
Rating dari keseluruhan Simulasi Sederhana Ini	5 - ★★★★★
Signature	
Added Time	23-Apr-2024 20:58:34
Referrer Name	
Task Owner	dzulfadlirusi@gmail.com

Hasil Validasi Responden a.n Rizal Arief

Name	Mahsyar, Saleh
Pengalokan Komponen dan Peralatan Instalasi	Sesuai
Penarikan Kabel dari KWH Panel ke ACPDB	Sesuai
Pengerjaan dari ACPDB Sampai Rectifier	Sesuai
Pengerjaan di perangkat BBU	Sesuai
Pengerjaan Kabel dari Shelter Sampai Tower	Sesuai
Pengerjaan di Perangkat Remote Radio Unit (RRU) sampai Antena Sectoral	Sesuai
Pengerjaan "Transmisi Jaringan" di dalam Shelter	Sesuai
Pengerjaan "Transmisi Jaringan" di luar shelter	Sesuai
Rating dari keseluruhan Simulasi Sederhana Ini	5 - ★★★★★
Signature	
Added Time	23-Apr-2024 20:28:03
Referrer Name	
Task Owner	dzulfadirusli@gmail.com

Powered by  Forms

Hasil Validasi Responden a.n Mahsyar Saleh

## Lampiran 4 Lembar Hasil Pengujian Black Box

Black Box Testing Report	
Form: Black Box Testing	
Name	Fadhil Muflih H, Firdaus
Splash Screen	Valid
Opening Scene	Valid
Simulation Scene	Valid
Canvas UI Toggle	Valid
Canvas Narasi	Valid
Komponen utama	Valid
Perkakas standar instalasi	Valid
Alat bantu lifting antenna	Valid
Peralatan standar untuk safety	Valid
Komponen pendukung	Valid
Audio Source	Valid
Tanggapan keseluruhan untuk skripsi ini	Semoga FPS simulatornya lebih lancar & lebih realists
Added Time	07-May-2024 13:47:58
Referrer Name	
Task Owner	dzulfadirusli@gmail.com

Name	muh rizaldi, loy
Splash Screen	Valid
Opening Scene	Valid
Simulation Scene	Valid
Canvas UI Toggle	Valid
Canvas Narasi	Valid
Komponen utama	Valid
Perkakas standar instalasi	Valid
Alat bantu lifting antenna	Valid
Peralatan standar untuk safety	Valid
Komponen pendukung	Valid
Audio Source	Valid
Tanggapan keseluruhan untuk skripsi ini	
Added Time	07-May-2024 13:26:46
Referrer Name	
Task Owner	dzulfadirusli@gmail.com

Hasil Pengujian Blackbox a.n Fadhil Mufli Hasil Pengujian Blackbox a.n Muh Rizaldi

Name	Muhammad Fauzan, Fahriandika
Splash Screen	Valid
Opening Scene	Valid
Simulation Scene	Valid
Canvas UI Toggle	Valid
Canvas Narasi	Valid
Komponen utama	Valid
Perkakas standar instalasi	Valid
Alat bantu lifting antenna	Valid
Peralatan standar untuk safety	Valid
Komponen pendukung	Valid
Audio Source	Valid
Tanggapan keseluruhan untuk skripsi ini	Simulasi pemasangan kabel fiber sangat bagus sebagai pembelajaran praktikum, tapi kontrol nya harus lebih responsif, karena setiap menekan sesuatu atau menyalakan sesuatu atau mengambil sesuatu tidak selalu respon dengan baik
Added Time	06-May-2024 16:58:58
Referrer Name	
Task Owner	dzulfadirusli@gmail.com

Hasil Pengujian Blackbox a.n Muhammad Fauzan

Name	Nur, Rahmi
Splash Screen	Valid
Opening Scene	Valid
Simulation Scene	Tidak Valid
Canvas UI Toggle	Valid
Canvas Narasi	Valid
Komponen utama	Valid
Perkakas standar instalasi	Valid
Alat bantu lifting antenna	Valid
Peralatan standar untuk safety	Valid
Komponen pendukung	Valid
Audio Source	Valid
Tanggapan keseluruhan untuk skripsi ini	Mungkin untuk tombol joystick bisa lebih di perlambatkan, lalu mungkin bisa selanjutnya di kembangkan lebih lanjut, agar bisa bekerja dua orang tapi dalam satu proyek yang sama.
Added Time	06-May-2024 16:56:23
Referrer Name	
Task Owner	dzulfadirusli@gmail.com

Hasil Pengujian Blackbox a.n Nur Rahmi

Name	Andi, Indraningsi
Splash Screen	Valid
Opening Scene	Valid
Simulation Scene	Valid
Canvas UI Toggle	Valid
Canvas Narasi	Valid
Komponen utama	Valid
Perkakas standar instalasi	Valid
Alat bantu lifting antenna	Valid
Peralatan standar untuk safety	Valid
Komponen pendukung	Valid
Audio Source	Valid
Tanggapan keseluruhan untuk skripsi ini	teknologi ini sangat bagus karena menciptakan lingkungan simulasi yang dapat dialami pengguna seolah-olah kita berada di dunia digital tersebut, serta dengan teknologi ini kita bisa menggunakannya di berbagai bidang seperti pendidikan, pelatihan maupun hiburan.
Added Time	04-May-2024 16:11:54
Referrer Name	
Task Owner	dzulfadirusli@gmail.com

Hasil Pengujian Blackbox a.n Andi Indraningsih

Name	Sulesti
Splash Screen	Valid
Opening Scene	Valid
Simulation Scene	Valid
Canvas UI Toggle	Valid
Canvas Narasi	Valid
Komponen utama	Valid
Perkakas standar instalasi	Valid
Alat bantu lifting antenna	Valid
Peralatan standar untuk safety	Valid
Komponen pendukung	Valid
Audio Source	Valid
Tanggapan keseluruhan untuk skripsi ini	Tanggapan saya terhadap pengujian VR pada hari ini, menurut saya bagus. Karena VR ini merupakan teknologi yang dapat di kembangkan di masa depan karena merujuk pada tampilan 3D yang menarik, karena menunjukkan kondisi dan situasi asli.
Added Time	04-May-2024 16:11:12
Referrer Name	
Task Owner	dzulfadirusli@gmail.com

Hasil Pengujian Blackbox a.n Sulesti

Name	Nurul, Alifiah
Splash Screen	Valid
Opening Scene	Valid
Simulation Scene	Valid
Canvas UI Toggle	Valid
Canvas Narasi	Valid
Komponen utama	Valid
Perkakas standar instalasi	Valid
Alat bantu lifting antenna	Valid
Peralatan standar untuk safety	Valid
Komponen pendukung	Valid
Audio Source	Valid
Tanggapan keseluruhan untuk skripsi ini	Untuk saat ini, VR bekerja dengan baik dengan sesuai langkah-langkah dan arahan yang telah ditentukan. Tetapi terkadang penampilan dalam VR nya kalau dilihat dari monitor terkadang menampilkan suasana room yang asli. Tanggal butuh penjelasan lebih lanjut terkait setiap langkah-langkah yang dilakukan, seperti untuk apa dan kenapa. Selebihnya sudah baik.
Added Time	04-May-2024 16:07:26
Referrer Name	
Task Owner	dzulfadirusli@gmail.com

Hasil Pengujian Blackbox a.n Nurul Alifiah

Name	Nurfadillah
Splash Screen	Valid
Opening Scene	Valid
Simulation Scene	Valid
Canvas UI Toggle	Valid
Canvas Narasi	Valid
Komponen utama	Valid
Perkakas standar instalasi	Valid
Alat bantu lifting antenna	Valid
Peralatan standar untuk safety	Valid
Komponen pendukung	Valid
Audio Source	Valid
Tanggapan keseluruhan untuk skripsi ini	Menurut saya, skripsi ini sudah sangat bagus karena menggunakan teknologi yang canggih dimana tampilan gambar maupun simulasi yg sedang berlangsung sudah sangat bagus dan jelas, dan sangat membantu untuk dapat melihat keadaan-keadaan maupun situasi yang berada di atas tower yang belum pernah di lihat secara langsung
Added Time	03-May-2024 21:45:08
Referrer Name	
Task Owner	dzulfadirusli@gmail.com

Hasil Pengujian Blackbox a.n Nurfadillah

Name	A. Fatimah, Azzahrah
Splash Screen	Valid
Opening Scene	Valid
Simulation Scene	Valid
Canvas UI Toggle	Valid
Canvas Narasi	Valid
Komponen utama	Valid
Perkakas standar instalasi	Valid
Alat bantu lifting antenna	Valid
Peralatan standar untuk safety	Valid
Komponen pendukung	Valid
Audio Source	Valid
Tanggapan keseluruhan untuk skripsi ini	tampilannya sudah sangat bagus, tapi sensitifitas pergerakan sangat cepat, baik pergerakan maju mundur maupun pergerakan badan sehingga terkadang sangat sulit mengontrol karakter kita dalam simulasi VRnya, terima kasih
Added Time	07-May-2024 14:51:25
Referrer Name	
Task Owner	dzulfadirusli@gmail.com

Hasil Pengujian Blackbox a.n A. Fatimah Azzahrah

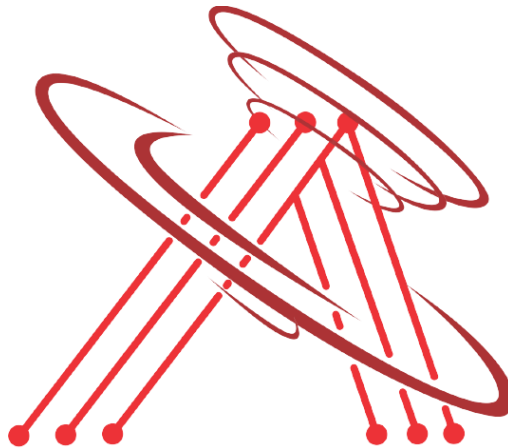
Name	Moh. Dzaugy, Zulfiqar Manaba
Splash Screen	Valid
Opening Scene	Valid
Simulation Scene	Valid
Canvas UI Toggle	Valid
Canvas Narasi	Valid
Komponen utama	Valid
Perkakas standar instalasi	Valid
Alat bantu lifting antenna	Valid
Peralatan standar untuk safety	Valid
Komponen pendukung	Valid
Audio Source	Valid
Tanggapan keseluruhan untuk skripsi ini	Secara keseluruhan pelatihan menggunakan vr ini sudah baik, materi yang di sampaikan sudah bagus dan sangat interaktif. Semoga kedepannya dapat dikembangkan atau ditambahkan beberapa topik training lainnya yang mengenai fiber, antenna, dan telekomunikasi.
Added Time	07-May-2024 14:51:06
Referrer Name	
Task Owner	dzulfadirusli@gmail.com

Hasil Pengujian Blackbox a.n Moh Dzaugy



*Lampiran 5 Manual Operation Modul VR*

**MANUAL OPERATION VR MODUL  
“SIMULASI INSTALASI JARINGAN TRANSMISI  
FIBER OPTIK PADA SISTEM SELULER”**



**LABORATORIUM ANTENA DAN PROPAGASI GELOMBANG  
DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS HASANUDDIN  
2024**

## **1.1 Deskripsi Umum**

Simulasi ini menggunakan teknologi virtual reality (VR) untuk menciptakan lingkungan simulasi jaringan transmisi fiber optik yang realistis dan interaktif. VR memungkinkan pengguna untuk berinteraksi langsung dengan komponen-komponen jaringan dalam lingkungan 3D, memberikan pengalaman belajar yang mendalam dan praktis. Melalui praktikum ini, praktikan akan mempelajari cara instalasi jaringan fiber optik pada sistem seluler. Teknologi ini dirancang untuk menyediakan simulasi yang cepat dan akurat, memungkinkan pengguna untuk mengamati implementasi nyata dan menganalisis kinerja jaringan tanpa harus melakukan eksperimen fisik yang mahal dan memakan waktu.

## **1.2 Teori Dasar**

### 1.3 Tujuan Praktikum

Mahasiswa dapat melakukan simulasi instalasi jaringan transmisi fiber optik pada sistem seluler menggunakan virtual reality.

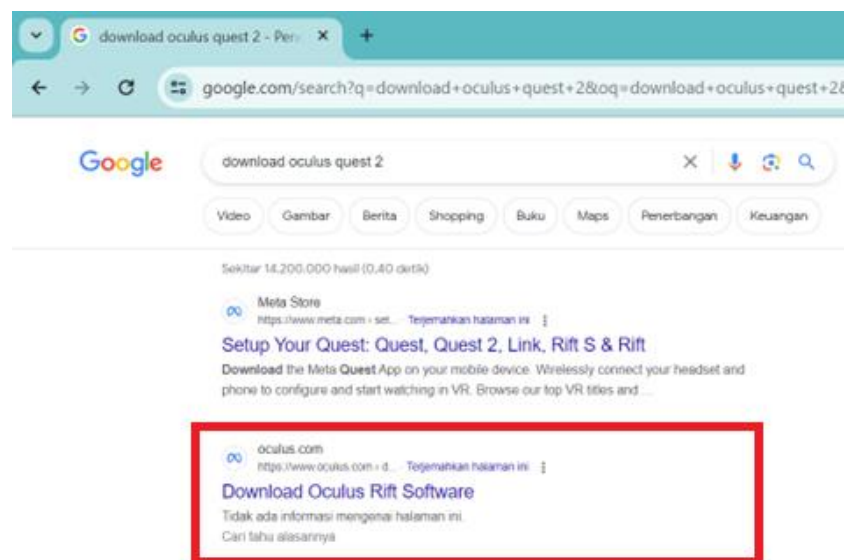
### 1.4 Alat dan Bahan

Laptop

### 1.5 Langkah - langkah Praktikum

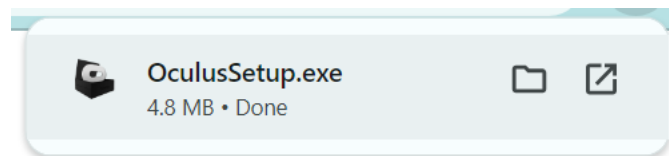
#### A. Install Oculus Quest

1. Download Oculus Software (Oculus Quest 2) pada browser.

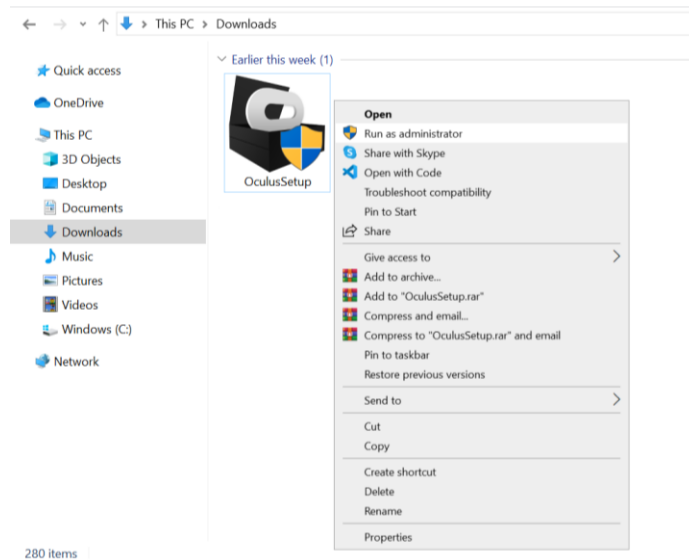


Gambar 1: Halaman pencarian ....

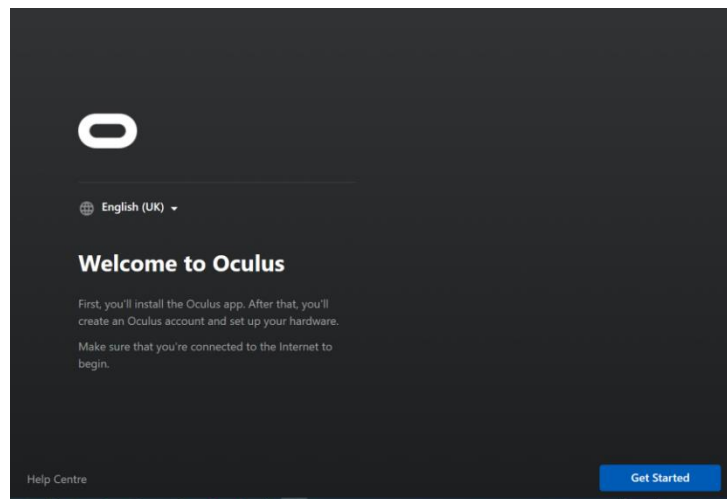
2. Tampilan setelah mengunduh software



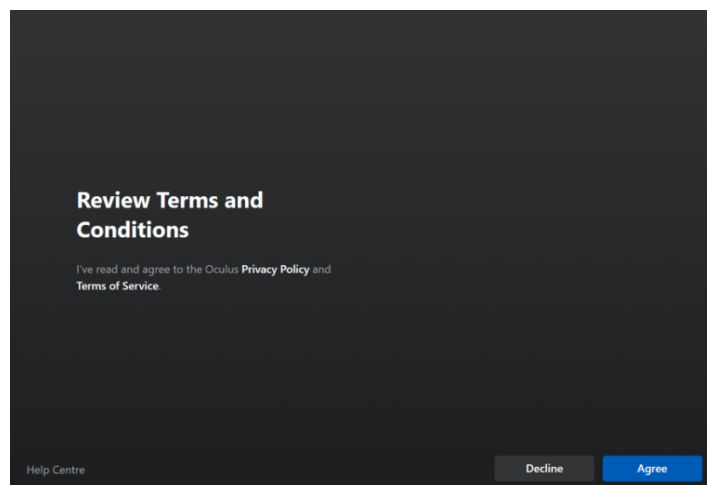
3. Selanjutnya, install software pada laptop.



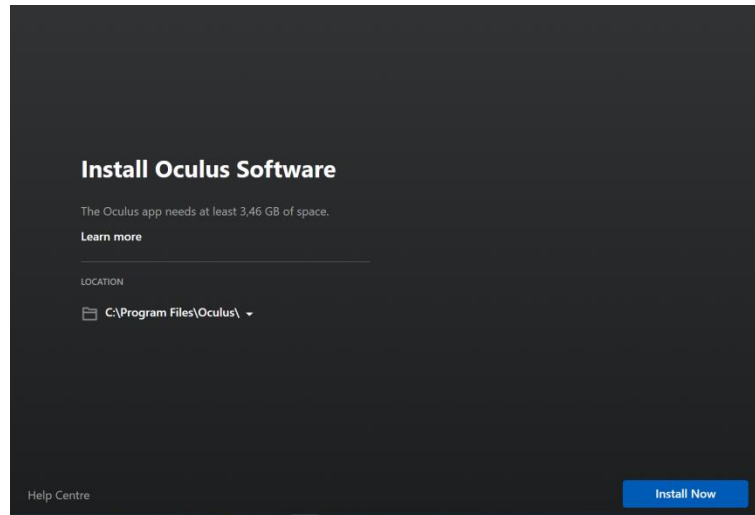
4. Tekan tombol “Get Started”



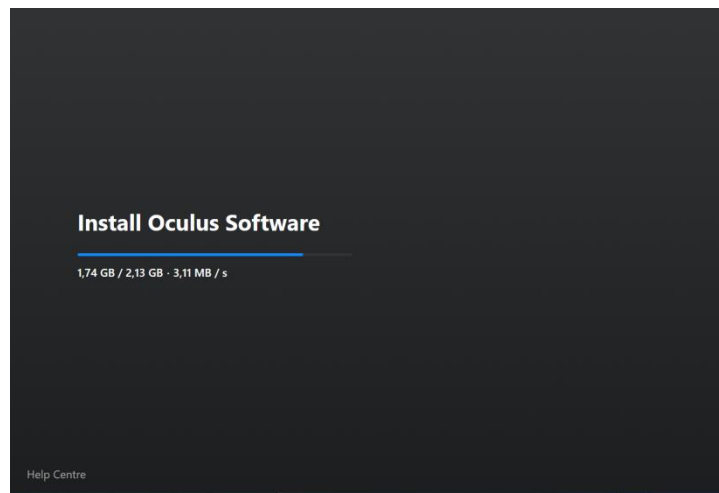
5. Kemudian, pilih “Agree”



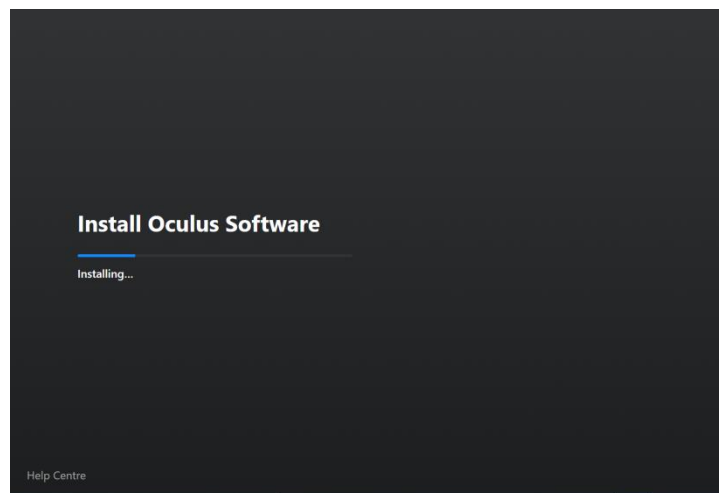
- Selanjutnya, tekan “Install Now” untuk mulai menginstall software oculus



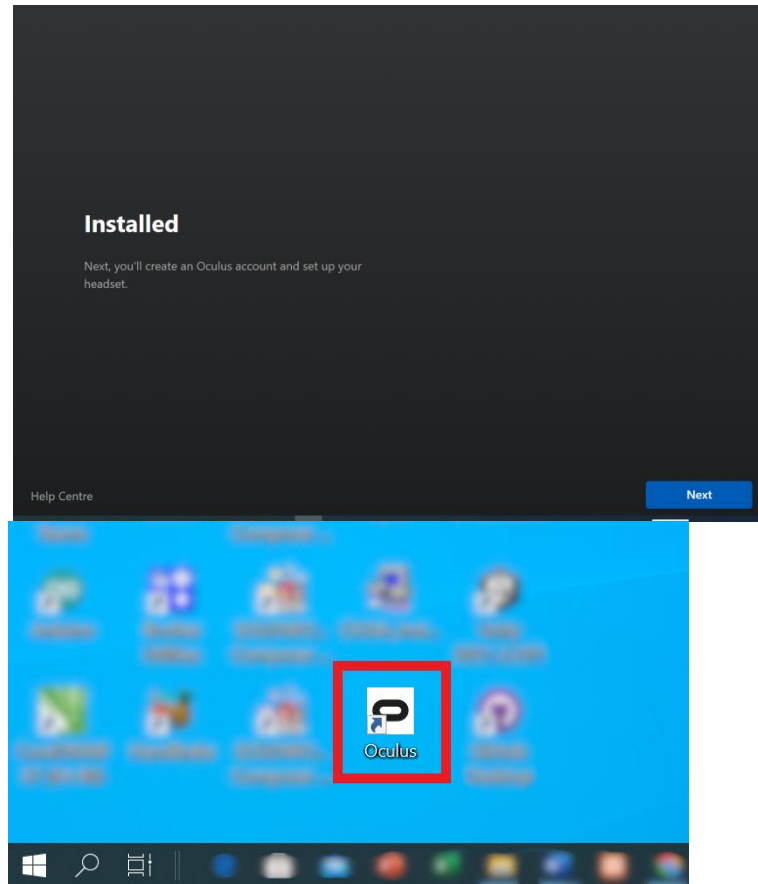
- Tunggu hingga proses install selesai



Ket: Proses install software

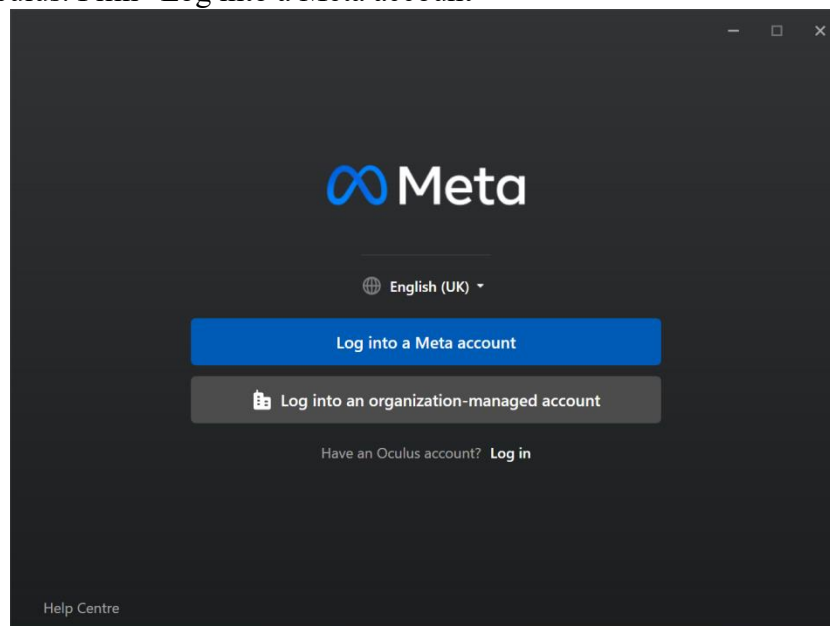


- Proses instalasi software oculus telah selesai.

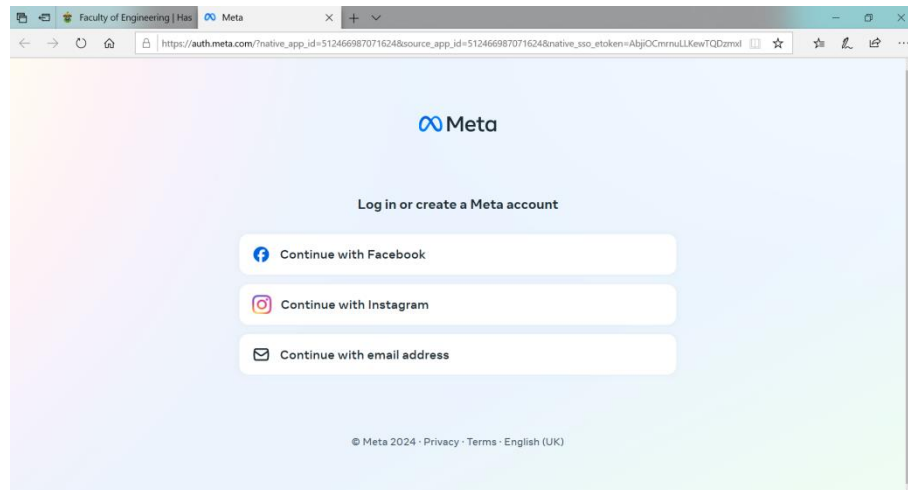


Tampilan software oculus pada dekstop

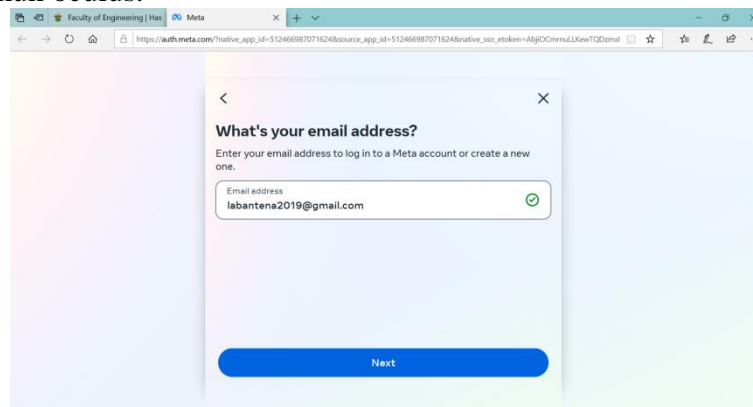
9. Setelah software oculus selesai diinstall, selanjutnya dilakukan proses Login ke akun oculus. Pilih "Log into a Meta account"



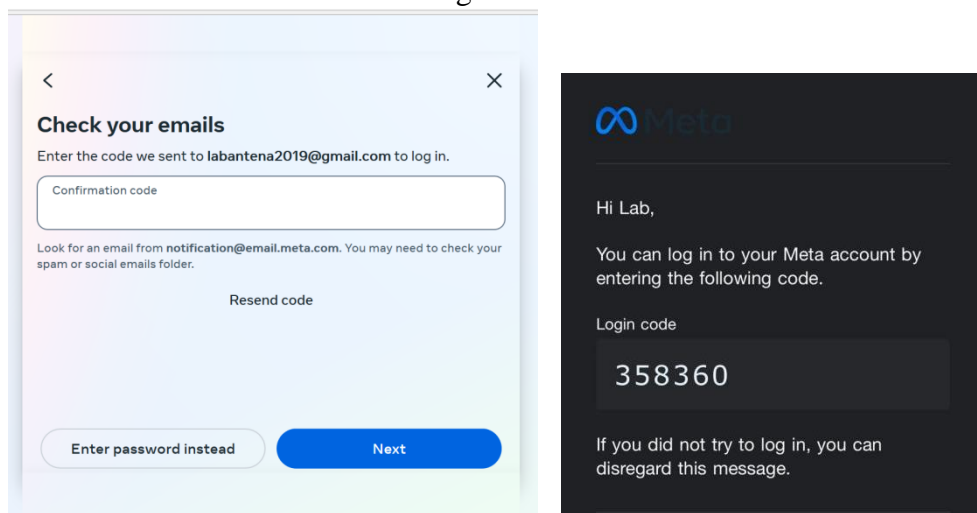
10. Pilih “Continue with email address”



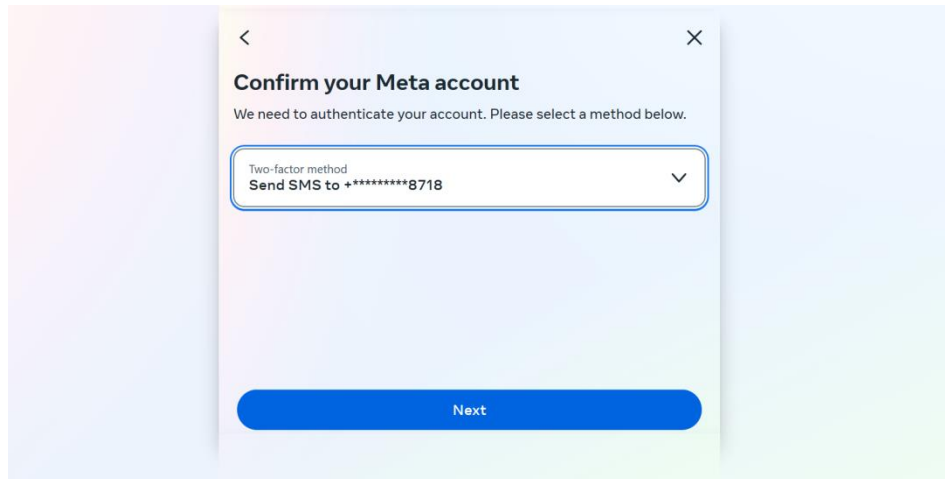
11. Masukkan email address sesuai dengan email yang digunakan pada perangkat Oculus Quest 2. Selanjutnya tekan tombol “Next” untuk melanjutkan proses login ke akun oculus.



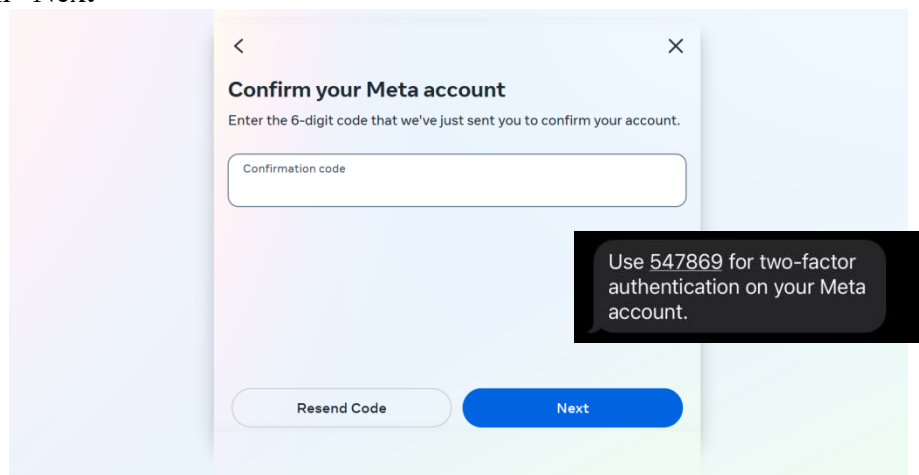
12. Masukkan kode yang terkirim ke Alamat email yang tadi di cantumkan. Kemudian tekan tombol Next untuk login ke akun oculus.



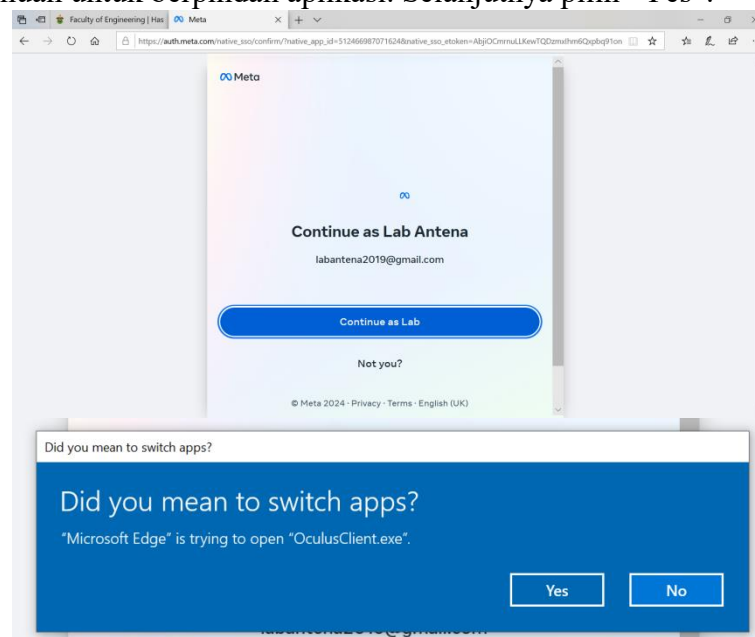
13. Tekan tombol “Next” ntuk mengonfirmasi akun oculus dengan melakukan proses autentikasi



14. Masukkan kode yang masuk sesuai dengan metode yang dipilih, kemudian tekan “Next”



15. Pilih “Continue as Lab” untuk melanjutkan proses login. Maka akan muncul pemberitahuan untuk berpindah aplikasi. Selanjutnya pilih “Yes”.

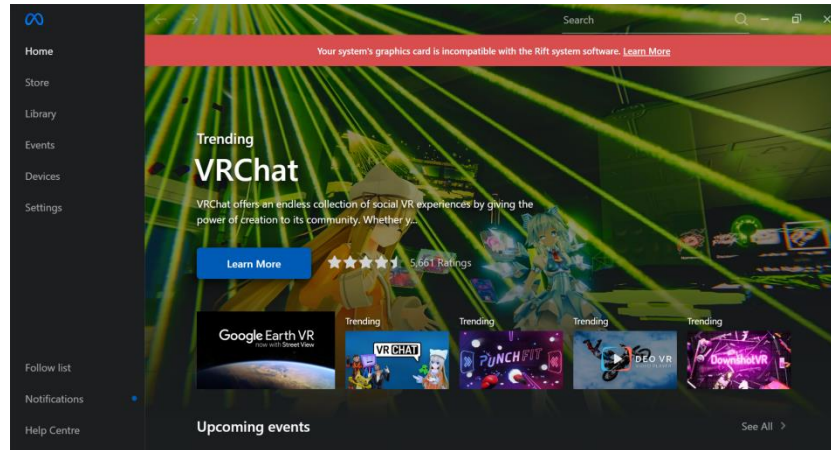


16. Berikut tampilan halamanan pada software oculus setelah dilakukan proses login



## B. Menghubungkan Laptop /PC dengan Perangkat Oculus

1. Pastikan bahwa headset Oculus Quest 2 sudah dinyalakan dan siap digunakan.
2. Buka aplikasi Oculus di Laptop/PC

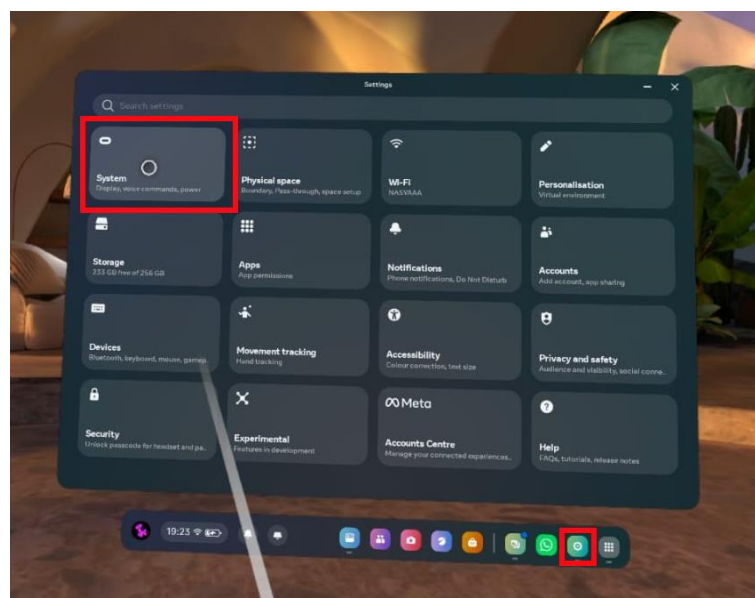


### Metode Akses “Oculus Link” atau “Air Link”:

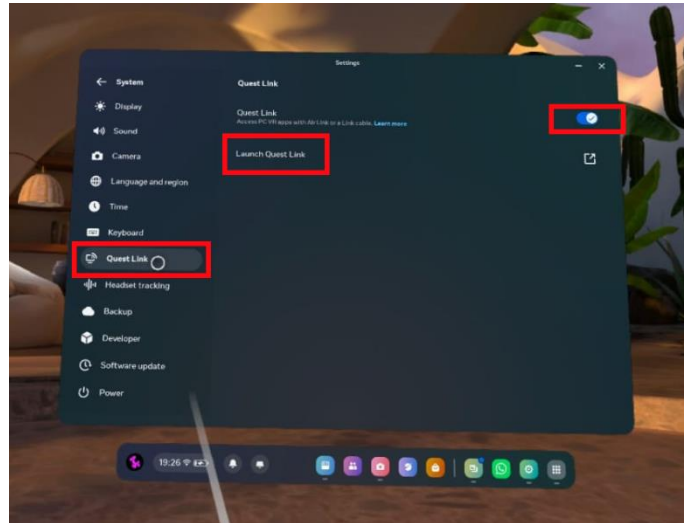
3. Jika Anda ingin menggunakan kabel, pastikan kabel USB-C ke USB 3.0 yang mendukung Oculus Link terhubung antara headset dan komputer.
4. Jika Anda menggunakan Air Link (koneksi nirkabel), pastikan Wi-Fi di komputer dan headset Anda terhubung ke jaringan yang sama.

### Metode Oculus Link:

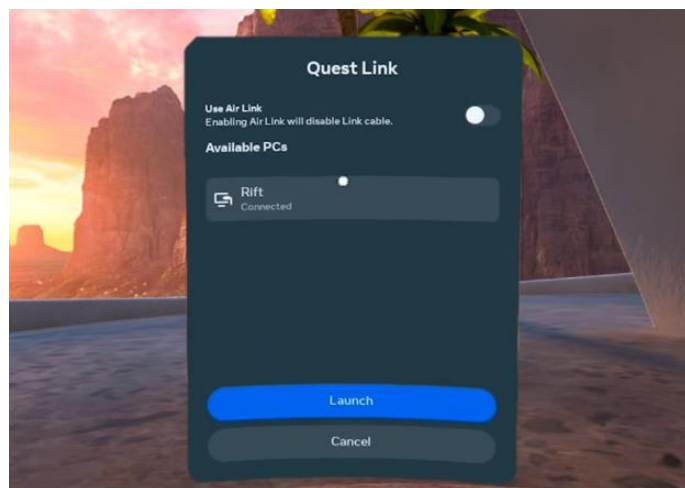
- Colokkan kabel USBC ke Laptop/PC dan ujung lainnya di colokkan pada headset VR
- b. Pada headset VR pilih menu pengaturan, lalu pilih ‘System’



- c. Pilih 'Quest Link' kemudian aktifkan lalu pilih 'Launch Quest Link'. Apabila Launch Quest Link tidak muncul, shutdown headset kemudian nyalakan kembali.



- d. Selanjutnya, pilih 'Launch' untuk mengarahkan ke pop-up selanjutnya

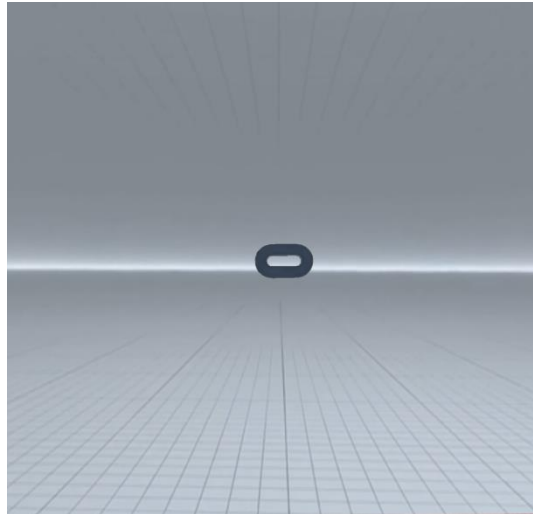


- e. Kemudian pilih 'Enable'



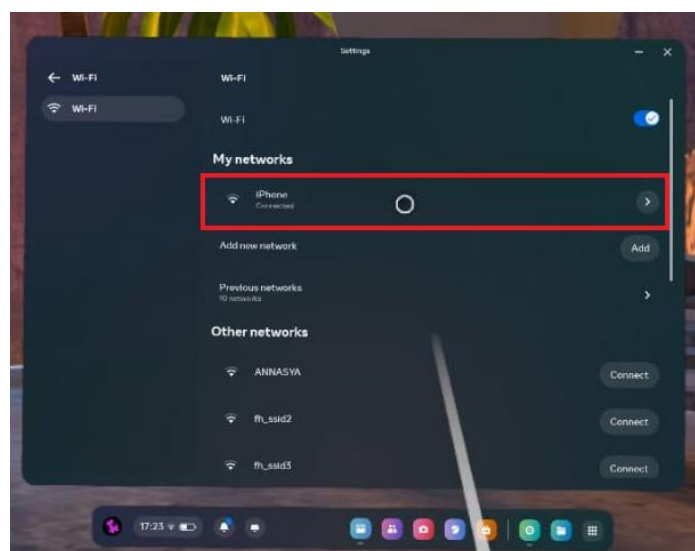
- f. Selanjutnya, akan muncul background putih kotak-kotak yang mendandakan

bahwa perangkat Oculus Quest 2 telah terhubung dengan Laptop/PC

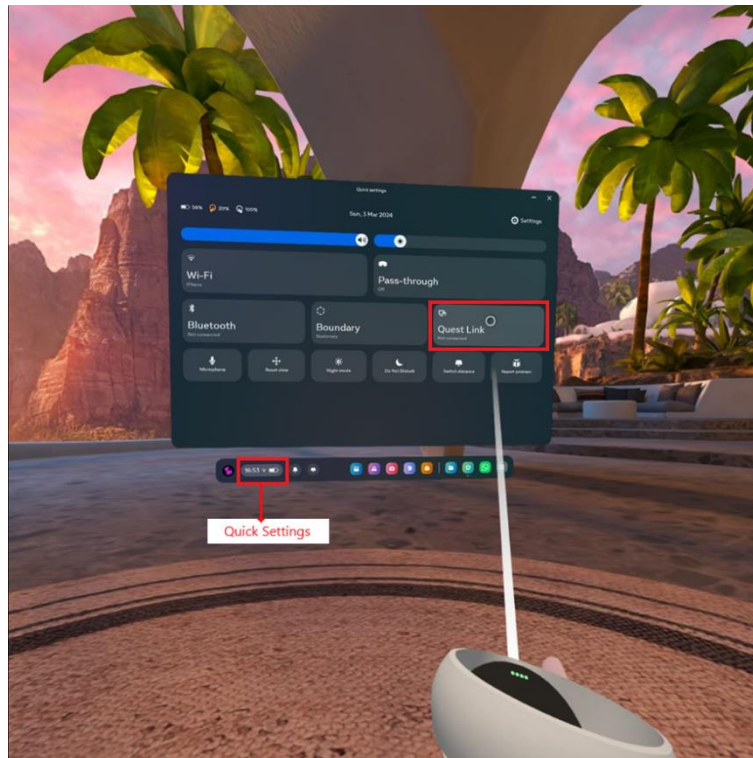


### Metode Air Link:

- Jika Anda menggunakan Air Link, pastikan headset (perangkat Oculus Quest 2) dan Laptop/PC terhubung ke jaringan yang sama.



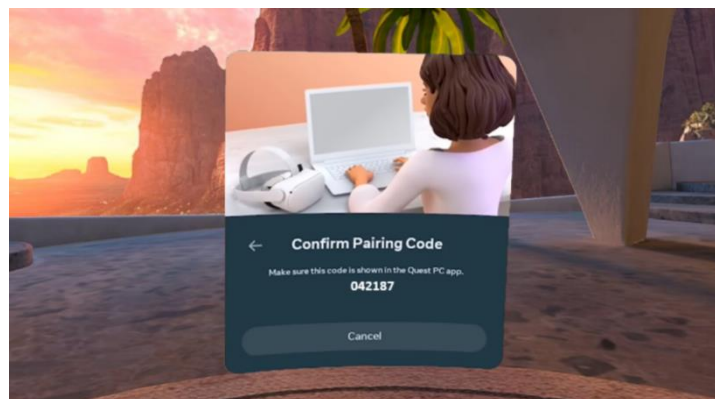
- Setelah perangkat Oculus Quest 2 dan Laptop/PC terhubung ke jaringan yang sama, selanjutnya pilih menu 'Quick Access' pada headseat Oculus Quest 2 kemudian pilih opsi 'Quest Link'.



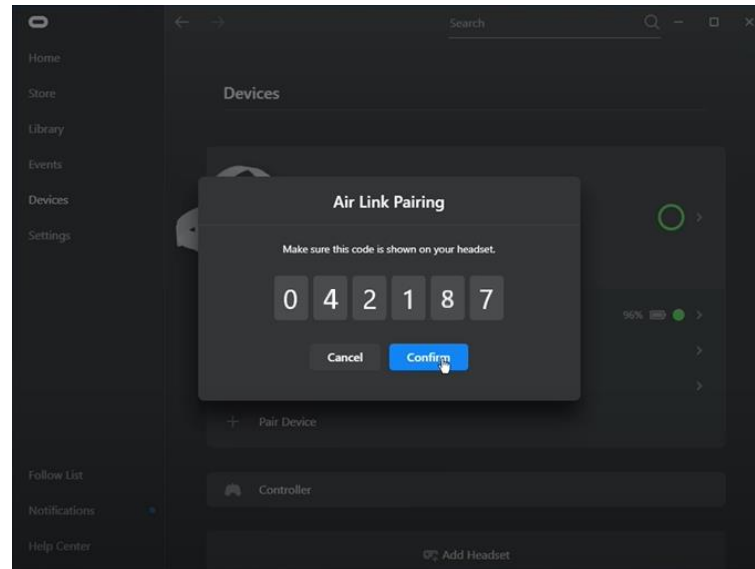
- Pilih 'Use Air Link' (aktifkan). Kemudian pilih 'Pair'



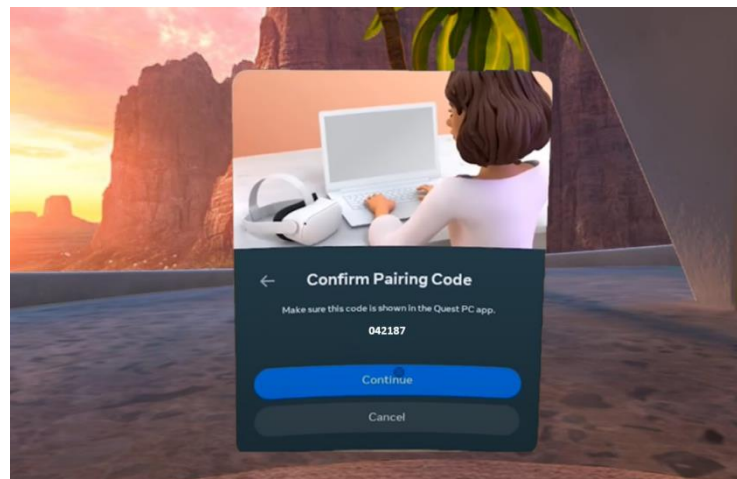
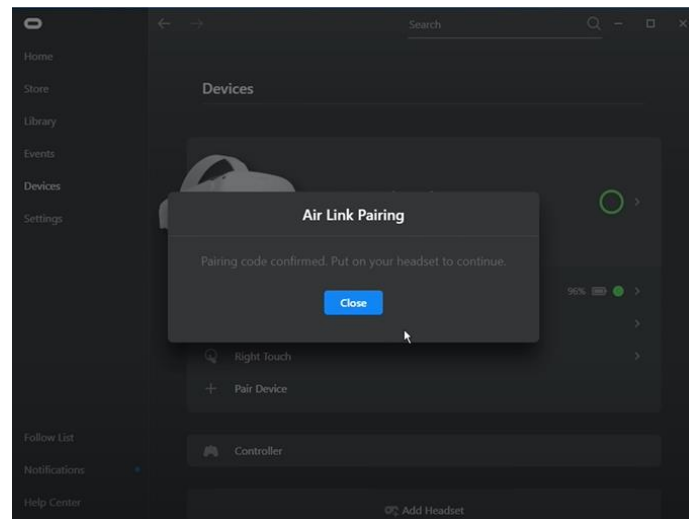
- Konfirmiasi kode pemasangan yang tertampil pada Laptop/PC (lihat tampilan Laptop/PC)



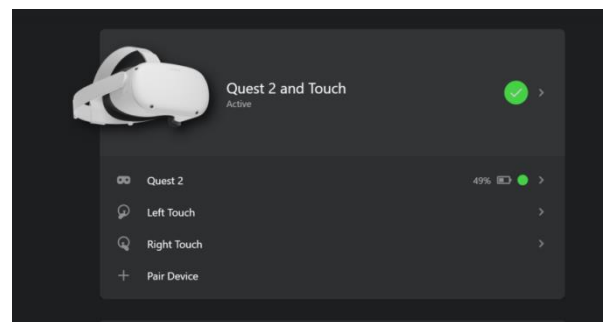
- Pada Laptop/PC tekan 'Confirm'



- Tekan Close. Lalu kembali ke headset Oculus Quest 2 kemudian pilih ‘Continue’



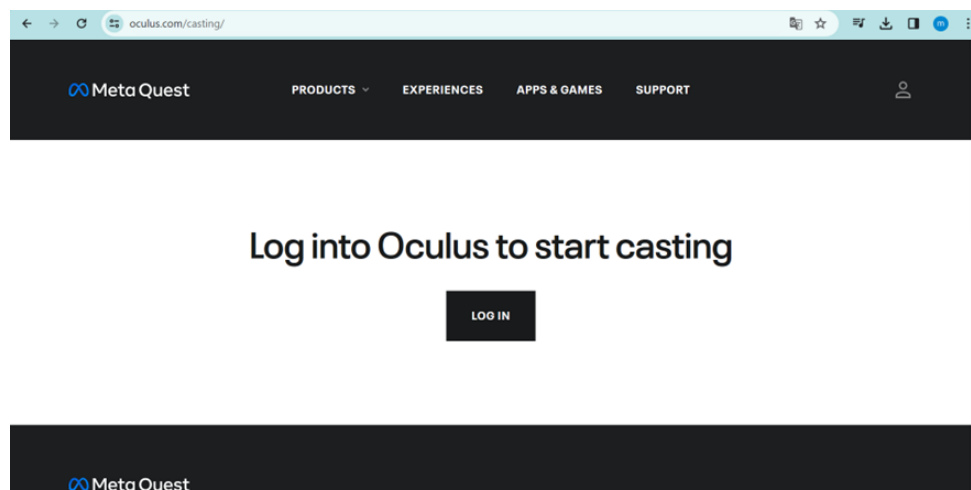
- Selanjutnya tekan ‘Launch’ maka perangkat Oculus Quest 2 telah terhubung ke Laptop/PC.



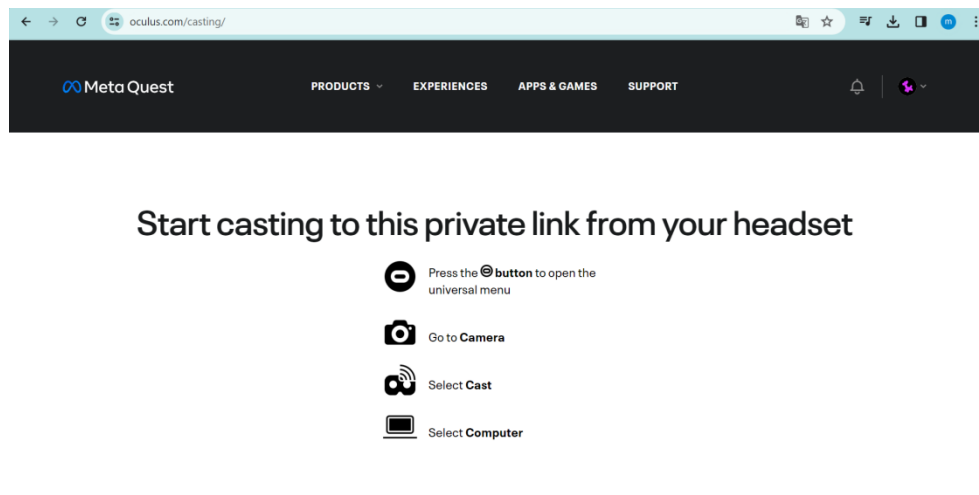
## B. Casting VR

1. Buka tautan <https://www.oculus.com/casting/> pada *browser*

“Tekan tombol ‘LOG IN’”



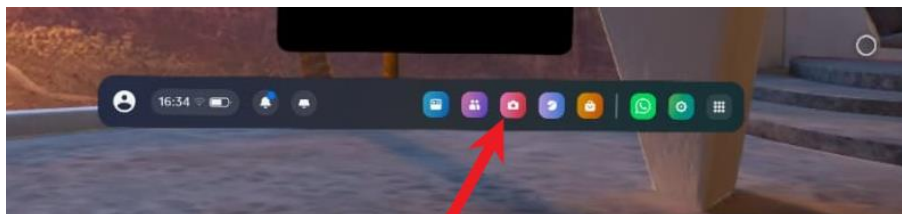
2. Tampilan setelah Login ke akun yang terhubung di perangkat Oculus Quest 2



3. Untuk memulai casting, tekan tombol  pada controller kanan oculus

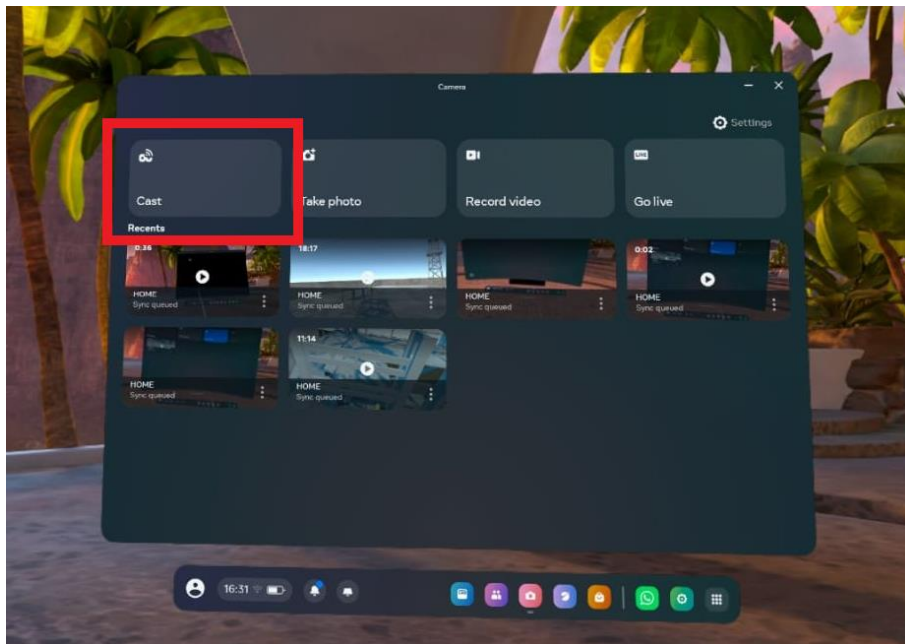


4. Kemudian, pilih menu kamera pada perangkat Oculus Quest 2

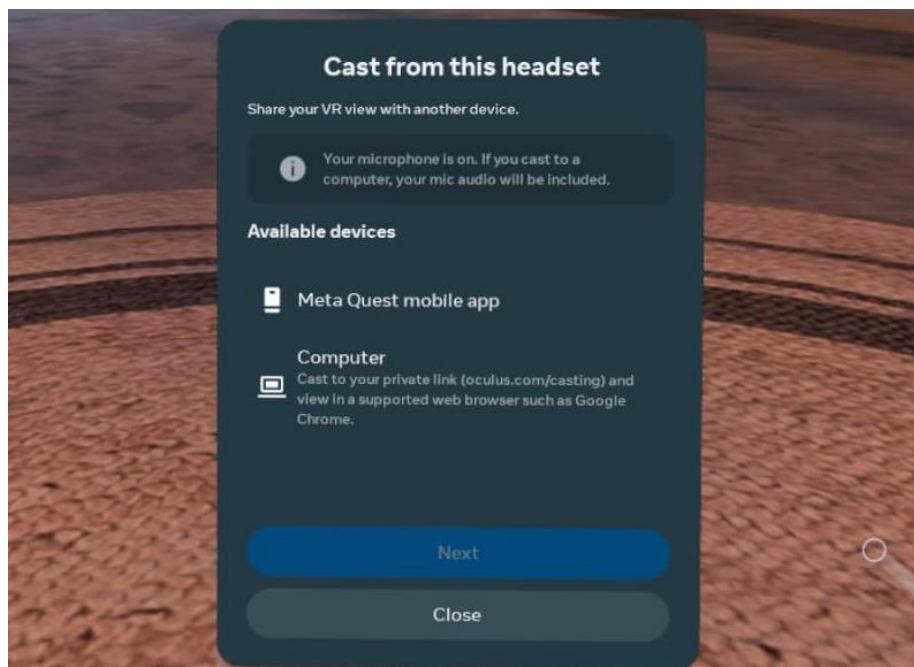




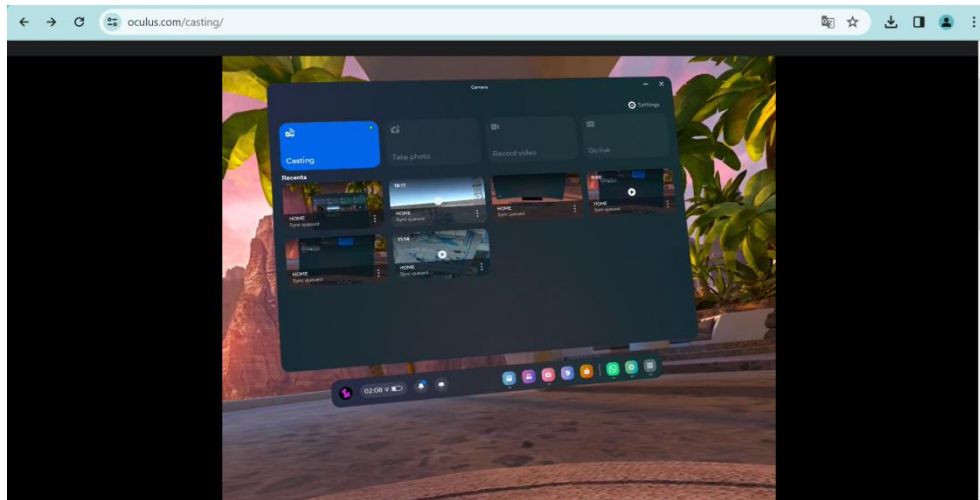
## 5. Pilih Cast



## 6. Pilih komputer kemudian tekan tombol “Next”

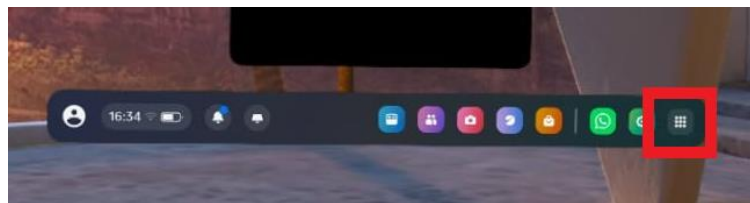


7. Apabila terdapat titik merah pada tampilan pada headset VR, maka proses casting telah dimulai.
8. Selanjutnya pada tampilan PC/Laptop setelah beberapa saat maka akan muncul casting dari headset VR. Berikut tampilan casting pada layar PC/ Laptop.

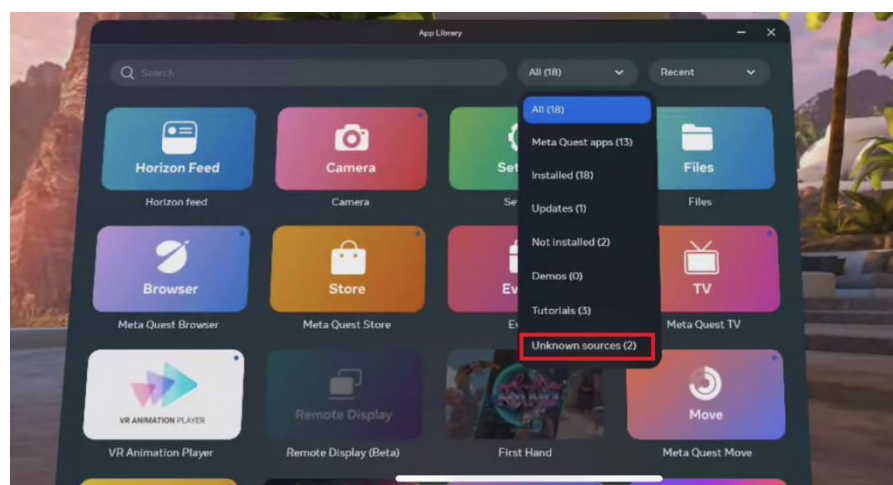


### C. Cara Membuka Aplikasi Simulasi Instalasi pada Headset VR

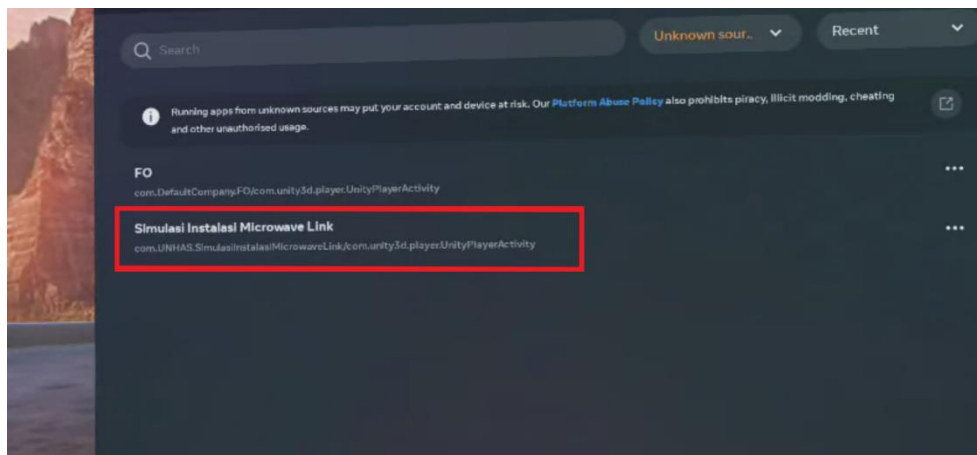
1. Hidupkan perangkat Oculus Quest
2. Pilih menu “App Library”



3. Kemudian, tekan dropdown “All” lalu pilih “Unknown Source”



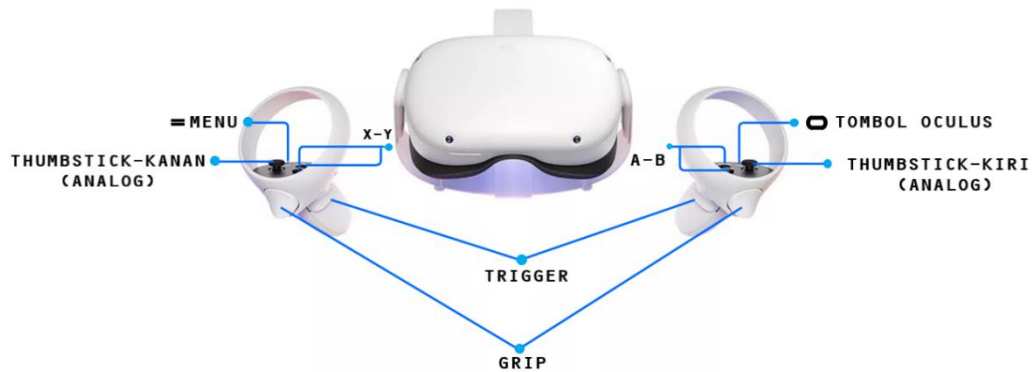
#### 4. Pilih “FO”



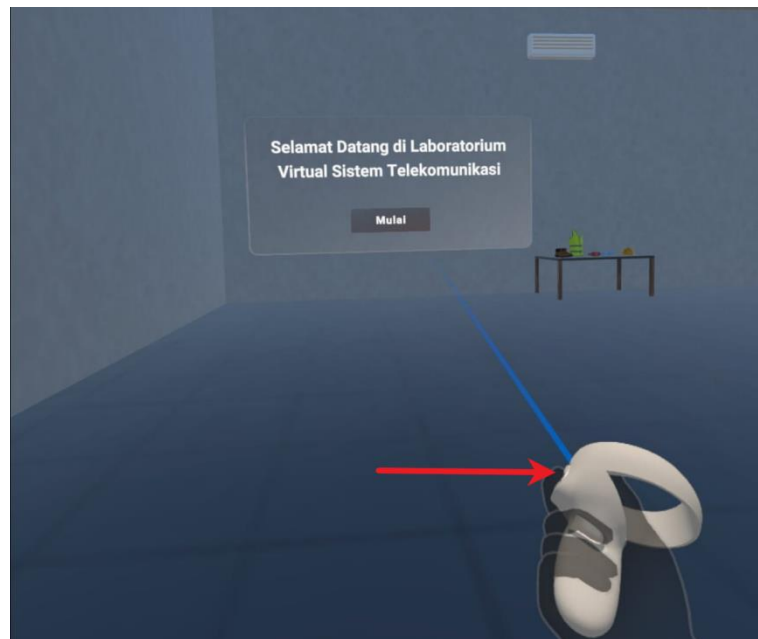
#### 5. Berikut tampilan awal simulasi instalasi setelah dijalankan di headset VR



#### D. Pengenalan Lingkungan Simulasi Instalasi



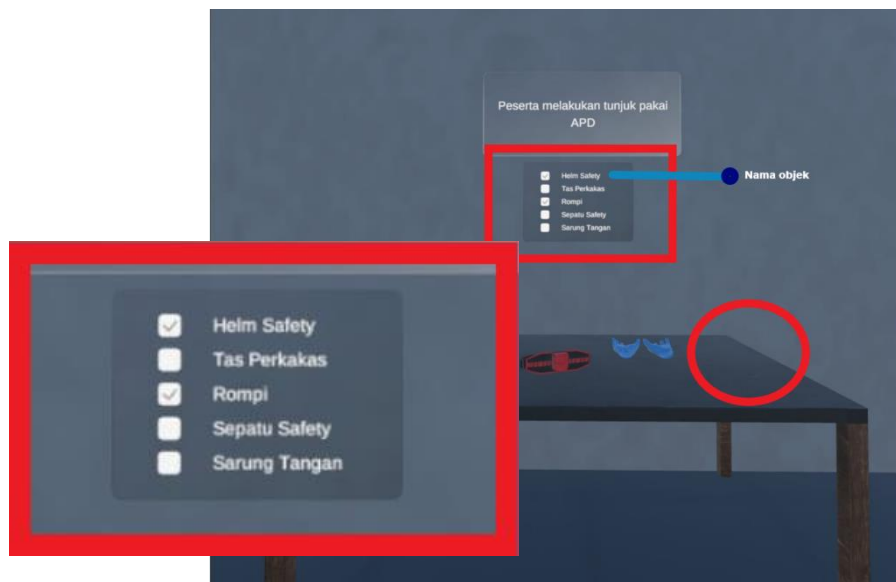
1. Adapun cara untuk berinteraksi dengan canvas yaitu dengan mengarahkan controller VR ke “canvas button” kemudian menekan tombol “grip” maka akan muncul teks selanjutnya.



2. Untuk proses mengenakan APD:
  - Arahkan controller ke setiap objek kemudian tekan tombol “grip” pada controller
  - Setelah itu, Objek yang menjadi target akan hilang kemudian checkbox nama objek yang dituju akan terverifikasi atau terceklis.



User mengarahkan controller ke objek kemudian menekan tombol grip



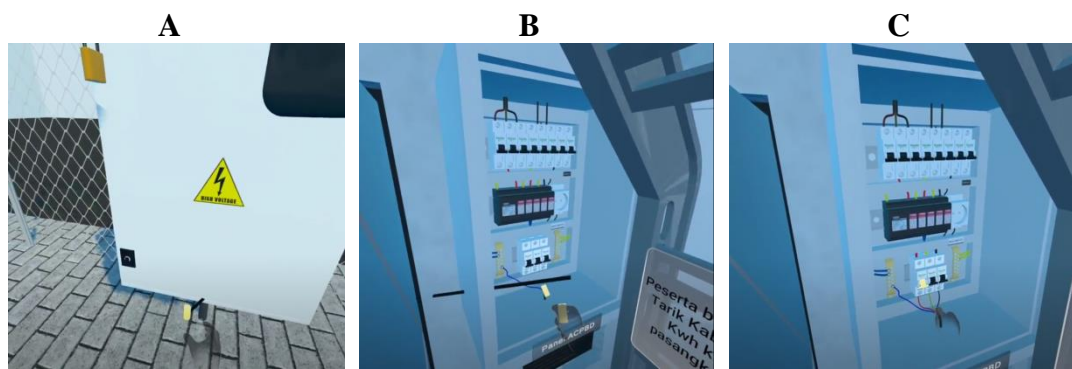
Verifikasi objek

3. Untuk proses perpindahan ke scene berikutnya, player berjalan menuju pintu. Pada saat player menyentuh collider yang ada pada pintu maka sistem secara otomatis akan memproses perpindahan ke scene berikutnya.
4. Lanjutkan tahap selanjutnya dengan mengikuti intruksi dalam simulasi praktikum virtual reality.

## E. Langkah-Langkah Simulasi Aktivasi Site Telekomunikasi

Langkah pertama dalam instalasi aktivasi site adalah pemasangan kabel 3 fasa dari KWH panel ke ACPDB, yang dilakukan untuk memastikan suplai listrik yang stabil dan aman ke perangkat telekomunikasi. Pemasangan ini sangat penting karena kabel 3 fasa mampu mengalirkan daya yang cukup besar, diperlukan untuk mengoperasikan peralatan dengan beban tinggi di site, serta memastikan keberlanjutan operasi site tanpa gangguan listrik.

Berikut adalah langkah-langkah pemasangan kabel 3 *phasa* dari KWH *panel* menuju ACPDB, sesuai dengan gambar di bawah ini:



Gambar A: KWH *Panel* dan Kabel 3 fasa (ditampilkan sebagai simbol tabung yang menyala). Pada tahap ini, kabel dalam posisi siap *grab* dan ditarik menuju ACPDB.

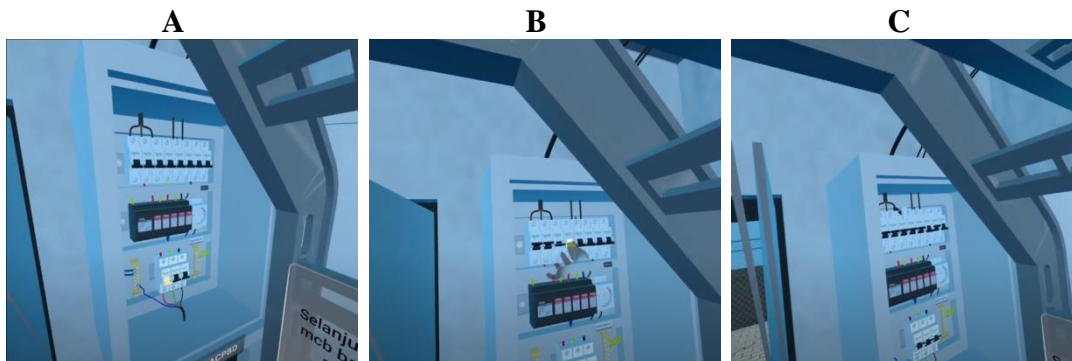
Gambar B: Kabel 3 *phasa* telah di *grab* dan ditarik menuju ACPDB. Pada tahap ini, kabel yang dalam posisi *grab* harus diarahkan dengan tepat hingga menyentuh tujuan pemasangan yang juga ditampilkan dalam bentuk tabung. Saat kabel mengenai tujuan ini maka penyambungan dari KWH *panel* ke ACPDB akan terjadi secara otomatis.

Gambar C: Kabel 3 fasa telah berhasil dipasang dan dihubungkan dengan ACPDB. Pada tahap ini ditampilkan kabel yang telah terhubung dengan ACPDB.

Langkah berikutnya adalah pengaktifan MCB di ACPDB, yang dilakukan untuk mengalirkan listrik ke seluruh rangkaian dan memastikan bahwa semua peralatan di site menerima daya yang cukup. Pengaktifan ini penting karena MCB berfungsi sebagai pelindung sirkuit, mencegah kerusakan akibat arus berlebih, dan

memastikan keamanan operasi site secara keseluruhan.

Berikut adalah langkah-langkah pengaktifan MCB di ACPDB sesuai dengan gambar di bawah ini:



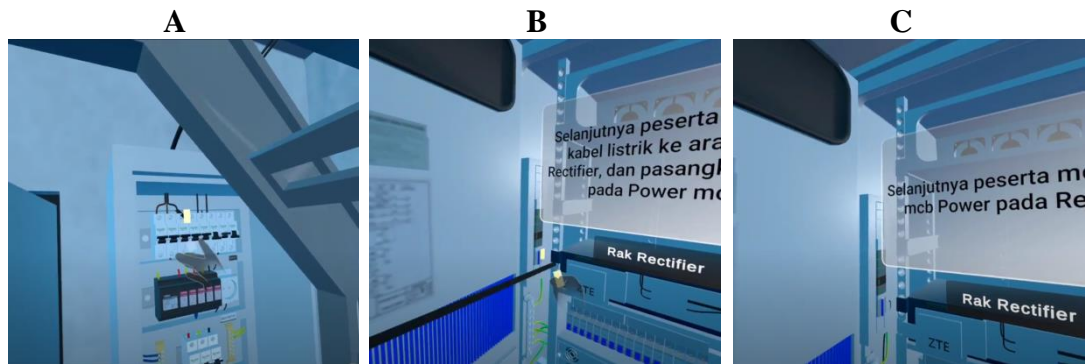
Gambar A: MCB pada ACPDB (ditampilkan dalam posisi OFF). Pada tahap ini, MCB belum diaktifkan, dan rangkaian listrik site masih dalam kondisi tidak teraliri daya. Gambar ini menunjukkan tampilan awal sebelum pengaktifan.

Gambar B: Proses pengaktifan MCB pada ACPDB. Pada tahap ini, MCB sedang dinaikkan atau diaktifkan, yang memungkinkan aliran listrik mengalir ke seluruh rangkaian. Pengaktifan ini adalah langkah penting untuk mulai memberikan daya ke perangkat di site.

Gambar C: MCB pada ACPDB telah berhasil diaktifkan (ditampilkan dalam posisi ON). Pada tahap ini, semua MCB telah diaktifkan, dan aliran listrik sudah tersedia di seluruh site, siap untuk mengoperasikan peralatan telekomunikasi.

Langkah berikutnya adalah pemasangan kabel 3 fasa dari ACPDB ke *rectifier*, yang dilakukan untuk menghubungkan sumber daya utama dengan sistem penyearah yang akan mengubah arus AC menjadi DC. Pemasangan ini sangat penting karena *rectifier* menyediakan daya DC yang stabil dan diperlukan untuk mengoperasikan perangkat telekomunikasi di *site*, sehingga memastikan kinerja yang optimal dan tanpa gangguan.

Berikut adalah langkah-langkah pemasangan kabel 3 *phasa* dari ACPDB ke *rectifier* sesuai dengan gambar di bawah ini:



Gambar A: Kabel 3 fasa (ditampilkan sebagai simbol tabung yang menyala) dalam posisi siap untuk di *grab* dan ditarik menuju *rectifier*. Pada tahap ini, kabel sudah disiapkan dan berada dalam posisi siap dipasang.

Gambar B: Kabel 3 fasa telah di *grab* dan sedang ditarik menuju bagian *power* pada *rectifier*. Pada tahap ini, kabel harus diarahkan dengan tepat hingga menyentuh tujuan pemasangan yang juga ditampilkan dalam bentuk tabung.

Gambar C: Kabel 3 *fasa* telah berhasil terpasang pada *power rectifier*. Pada tahap ini, kabel sudah terhubung dengan *rectifier*, memastikan aliran daya dari ACPDB ke *rectifier* berjalan dengan baik.

Langkah berikutnya adalah mengaktifkan MCB di *rectifier* dan memasang kabel power dari *rectifier* ke bagian *socket power* BBU. Pengaktifan MCB diperlukan untuk mengalirkan daya ke *rectifier*, sementara pemasangan kabel power memastikan bahwa daya yang telah diubah menjadi DC oleh *rectifier* dapat disuplai ke BBU untuk mendukung operasi telekomunikasi secara efektif.

Berikut mcb yang digunakan untuk memasang kabel di *rectifier*. Karena BBU menggunakan 2 power maka peserta memakai dua mcb pertama dari kiri seperti pada gambar dibawah ini.





Selanjutnya langkah untuk memasang kabel power dari rectifier menuju socket power BBU sesuai dengan gambar di bawah ini:



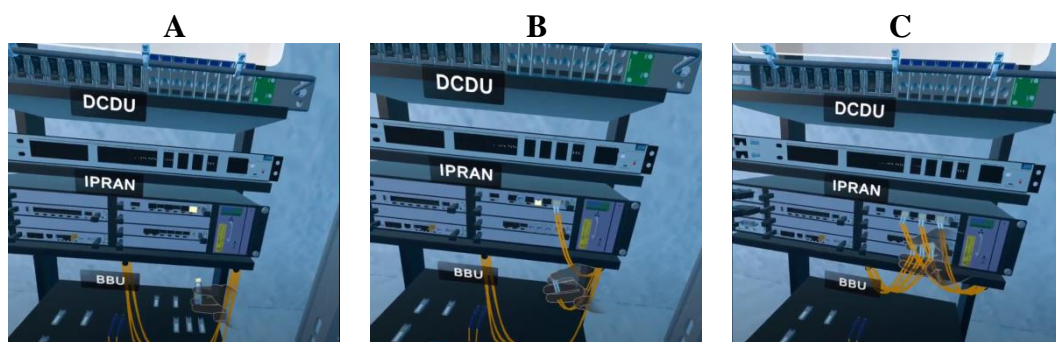
Gambar A: Kabel power (ditampilkan sebagai simbol tabung) berada dalam posisi siap untuk ditarik ke socket power BBU. Pada tahap ini, kabel sudah dipersiapkan untuk dipindahkan ke lokasi pemasangan.

Gambar B: Kabel power sedang dipasang pada socket power BBU. Pada tahap ini, kabel yang di *grab* harus diposisikan dengan tepat dan dipasang ke socket BBU untuk memastikan koneksi yang benar.

Gambar C: Kedua kabel power telah berhasil terpasang pada socket power BBU. Pada tahap ini, kabel sudah terhubung dengan socket, memastikan aliran daya dari rectifier ke BBU siap untuk digunakan.

Langkah berikutnya adalah memasang 6 SFP ke BBU dan menghubungkan 6 kabel patch core ke masing-masing SFP. Pemasangan SFP diperlukan untuk menyediakan konektivitas jaringan yang diperlukan, sementara kabel patch core menghubungkan SFP dengan perangkat jaringan lain, memastikan data dapat ditransfer dengan efektif dan tanpa gangguan.

Berikut pemasangan 6 SFP ke BBU dan pemasangan kabel patch core ke masing masing SFP sesuai dengan gambar di bawah ini:



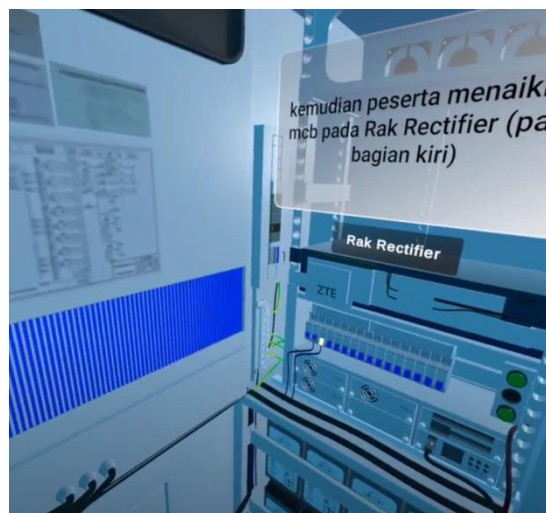
Gambar A: Pemasangan 6 SFP ke socket BBU. Pada tahap ini, SFP sedang dipasang ke slot yang sesuai di BBU, siap untuk menyediakan konektivitas jaringan.

Gambar B: Pemasangan kabel patch core ke masing-masing SFP. Pada tahap ini, kabel patch core dihubungkan ke setiap SFP, memastikan koneksi jaringan yang tepat dan stabil.

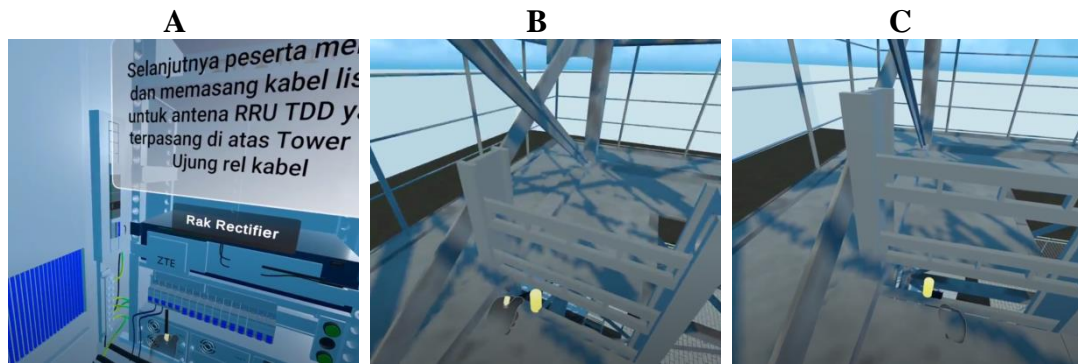
Gambar C: Hasil dari pemasangan kedua SFP dan kabel patch core. Pada tahap ini, semua SFP telah terpasang dengan kabel patch core yang terhubung, siap untuk mendukung transfer data dan operasi jaringan.

Langkah berikutnya adalah mengaktifkan 3 MCB pada rectifier untuk dihubungkan ke power RRU. Pengaktifan MCB diperlukan untuk mengalirkan daya dari rectifier ke RRU, memastikan bahwa RRU mendapatkan daya yang stabil dan memadai untuk berfungsi dengan baik dalam sistem telekomunikasi.

Berikut mcb yang digunakan untuk memasang kabel di *rectifier*. Karena RRU memiliki 3 jenis TDD maka peserta memakai tiga mcb pertama dari kiri seperti pada gambar dibawah ini.



Berikut cara menaikkan kabel power rru menuju ke ujung jalur kabel yang berada di tower sesuai dengan gambar di bawah ini:



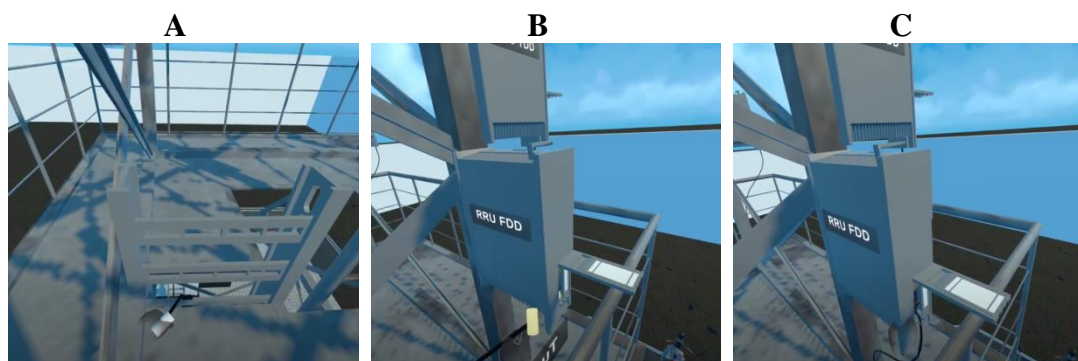
Gambar A: Kabel (ditampilkan sebagai simbol kabel yang di grab dan siap untuk ditarik) berada dalam posisi siap untuk dipindahkan menuju ujung jalur kabel di atas tower. Pada tahap ini, kabel sudah dipersiapkan untuk dipasang.

Gambar B: Kabel yang di grab dan sudah siap ditempatkan di ujung jalur kabel pada tower. Pada tahap ini, kabel diatur dan diposisikan dengan hati-hati di tempat yang telah ditentukan di tower.

Gambar C: Kabel telah berhasil terpasang di ujung jalur kabel pada tower. Pada tahap ini, kabel sudah terpasang dengan benar di lokasi yang diinginkan.

Langkah berikutnya adalah memasang kabel power RRU yang berada pada ujung jalur kabel menuju socket RRU. Pemasangan kabel power ini penting untuk memastikan bahwa RRU menerima pasokan daya yang diperlukan untuk operasionalnya, sehingga dapat berfungsi dengan baik dalam sistem telekomunikasi.

Berikut pemasangan 6 kabel power ke masing masing rru sesuai dengan gambar di bawah ini:



Gambar A: Kabel power RRU (ditampilkan sebagai simbol kabel yang siap untuk ditarik) berada dalam posisi siap untuk dipindahkan menuju masing-masing RRU.

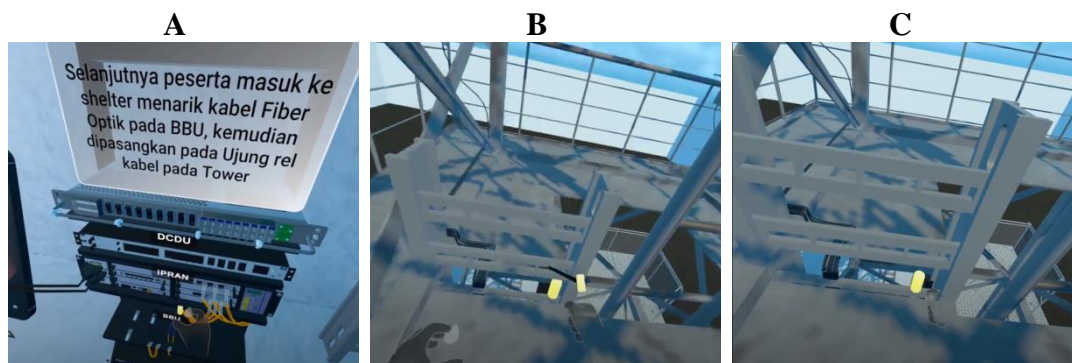
Pada tahap ini, kabel sudah disiapkan untuk dipasang.

Gambar B: Kabel power sedang di *grab* dan siap untuk dipasang di RRU. Pada tahap ini, kabel diposisikan dengan hati-hati agar dapat dihubungkan ke socket RRU dengan benar.

Gambar C: Kabel power telah berhasil terpasang di RRU. Pada tahap ini, kabel sudah terhubung dengan baik ke socket RRU.

Langkah berikutnya adalah menarik 6 kabel patch core menuju ke RRU. Penarikan kabel patch core ini penting untuk menghubungkan RRU dengan sistem jaringan utama, memastikan transfer data yang efisien dan konektivitas yang stabil antara perangkat.

Berikut cara menaikkan kabel patch core rru menuju ke ujung jalur kabel yang berada di tower sesuai dengan gambar di bawah ini:



Gambar A: Kabel patch core (ditampilkan sebagai simbol tabung yang siap untuk ditarik) berada dalam posisi siap untuk dipindahkan menuju ujung jalur kabel. Pada tahap ini, kabel sudah disiapkan untuk ditarik ke lokasi yang ditentukan.

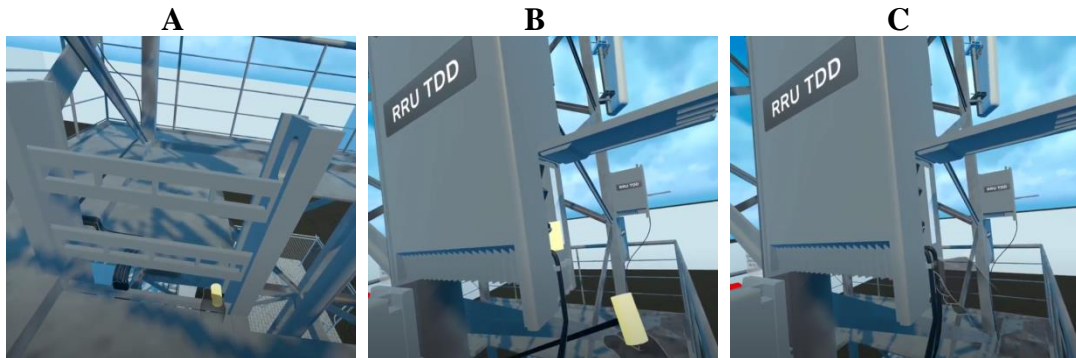
Gambar B: Kabel patch core yang siap ditempatkan di tempat tujuan yang disediakan. Pada tahap ini, kabel diatur dan diposisikan dengan hati-hati di lokasi yang telah disiapkan untuk pemasangan.

Gambar C: Kabel patch core telah berhasil terpasang di ujung jalur kabel. Pada tahap ini, kabel sudah terhubung dengan benar di lokasi yang diinginkan.

Langkah berikutnya adalah memasang kabel patch core ke masing-masing RRU. Pemasangan kabel patch core ini penting untuk menghubungkan RRU dengan sistem jaringan utama, memastikan konektivitas data yang stabil dan efisien antara RRU dan perangkat jaringan lainnya.

Berikut pemasangan kabel patch core menuju rru sesuai dengan gambar di

bawah ini:



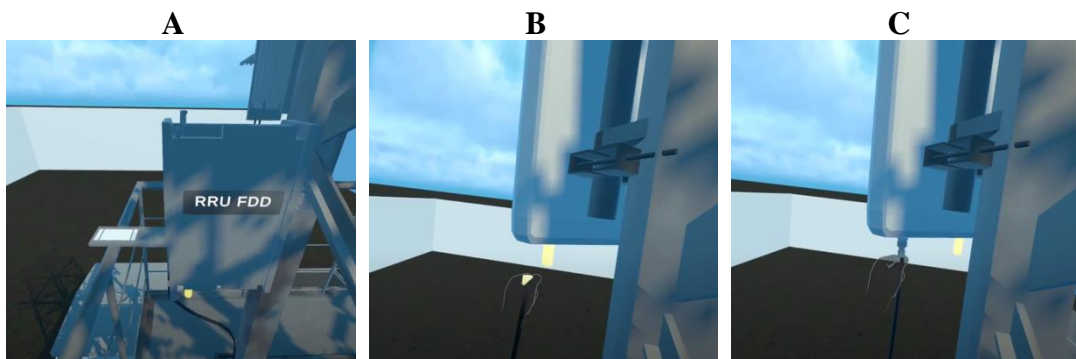
Gambar A: Kabel patch core (ditampilkan sebagai simbol tabung siap untuk ditarik) berada dalam posisi siap untuk dipindahkan menuju RRU. Pada tahap ini, kabel sudah disiapkan untuk dipasang.

Gambar B: Kabel patch core sedang diposisikan dengan hati-hati dan siap untuk dipasang di RRU. Pada tahap ini, kabel dipersiapkan untuk dihubungkan dengan SFP RRU.

Gambar C: Kabel patch core telah berhasil terpasang pada RRU. Pada tahap ini, kabel sudah terhubung dengan baik ke RRU, memastikan konektivitas data yang stabil dan efisien.

Langkah berikutnya adalah memasang semua jumper dari RRU menuju antena panel. Pemasangan jumper ini penting untuk menghubungkan RRU dengan antena panel, memastikan transmisi sinyal yang efektif dan konektivitas jaringan yang optimal.

Berikut pemasangan jumper ke antena panel sesuai dengan gambar dibawah ini:



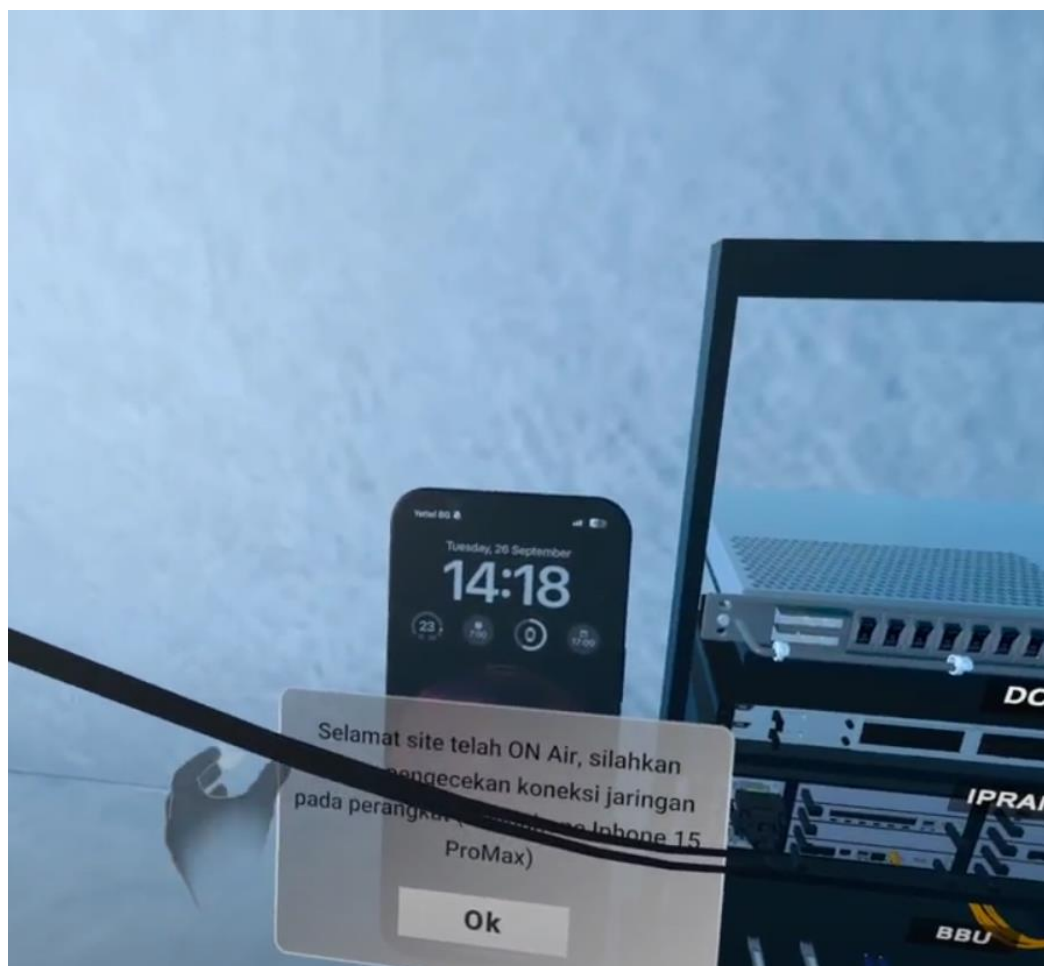
Gambar A:\*\* Jumper (ditampilkan sebagai simbol tabung siap untuk ditarik)

berada dalam posisi siap untuk dipindahkan menuju antenna panel. Pada tahap ini, jumper sudah disiapkan untuk dipasang.

Gambar B: Jumper sedang diposisikan dengan hati-hati dan siap untuk dipasang di antenna panel. Pada tahap ini, jumper dipersiapkan untuk dihubungkan dengan antenna panel.

Gambar C: Jumper telah berhasil terpasang ke antenna panel. Pada tahap ini, jumper sudah terhubung dengan baik ke antenna panel, memastikan konektivitas yang diperlukan untuk transmisi sinyal.

Selanjutnya konfigurasi otomatis berlangsung dan site telah on air yang menyebabkan daerah tersebut sudah memiliki jaringan telepon sesuai dengan gambar dibawah ini



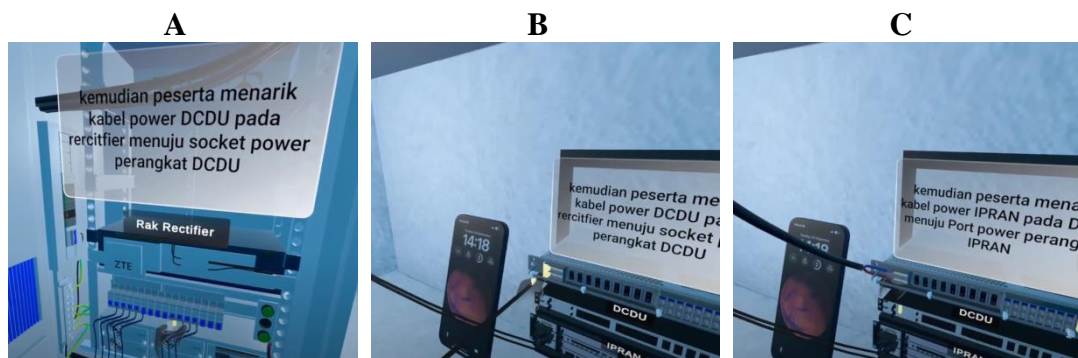
## F. Langkah langkah Interkoneksi Antar Site

Pertama, aktifkan MCB pada rectifier untuk nantinya dihubungkan ke power

DCDU sesuai dengan gambar di bawah ini. Pengaktifan MCB ini diperlukan untuk memulai aliran daya dari rectifier ke DCDCU, memastikan bahwa DCDCU menerima pasokan daya yang stabil dan memadai untuk operasionalnya.



Berikut adalah langkah untuk memasang kabel power dari rectifier menuju power DCDCU sesuai dengan gambar di bawah ini:



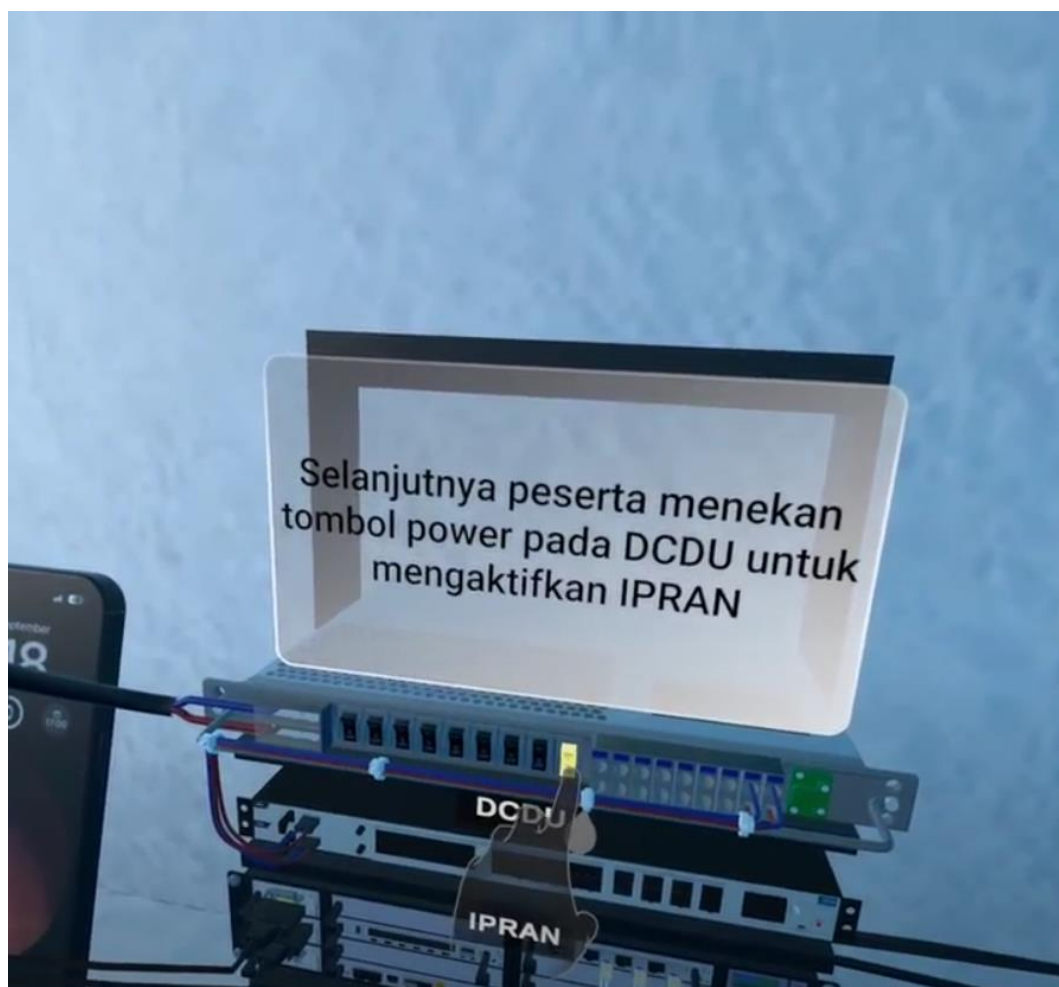
Gambar A: Kabel power (ditampilkan sebagai simbol tabung siap untuk ditarik) berada dalam posisi siap untuk dipindahkan menuju power DCDCU. Pada tahap ini,

kabel sudah disiapkan untuk dipasang.

Gambar B: Kabel power sedang diposisikan dengan hati-hati dan siap untuk dipasang di DCDU. Pada tahap ini, kabel dipersiapkan untuk dihubungkan dengan socket pada DCDU.

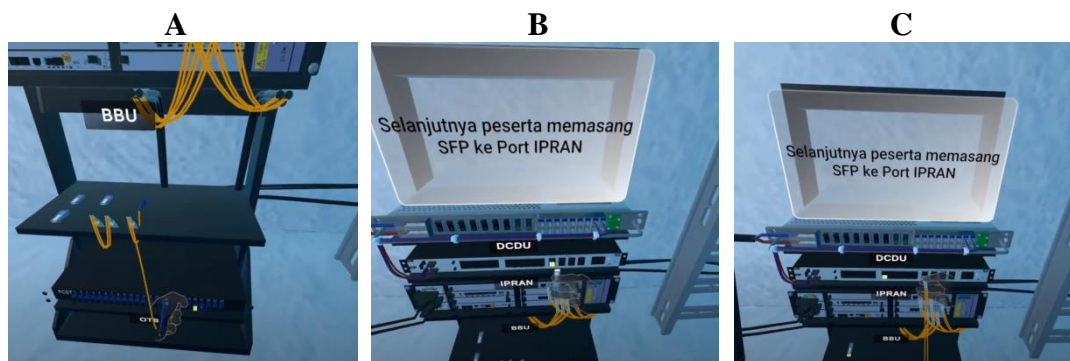
Gambar C: Kabel power telah berhasil terpasang di DCDU. Pada tahap ini, kabel sudah terhubung dengan baik ke DCDU.

Langkah berikutnya adalah mengaktifkan DCDU dengan menekan push button sesuai dengan gambar dibawah ini



langkah berikutnya adalah instalasi untuk interkoneksi nya sesuai dengan gambar dibawah ini:





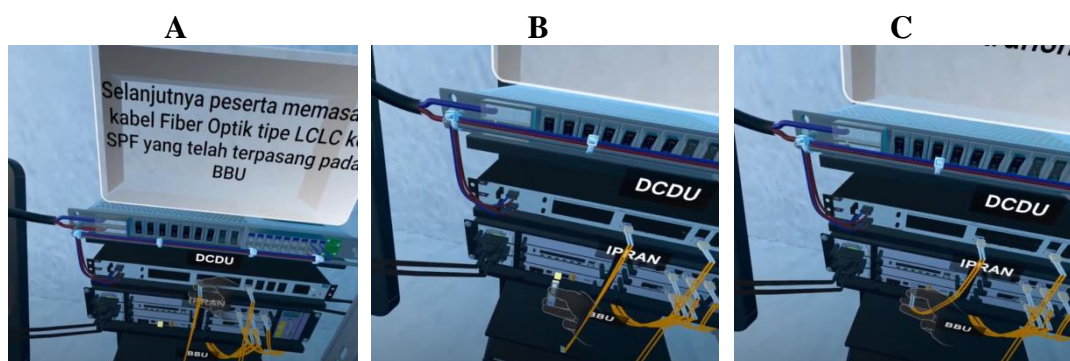
Gambar A: Pemasangan kabel patch core ke OTB (Optical Termination Box). Pada tahap ini, kabel patch core sedang dipasang ke socket di OTB.

Gambar B: Ujung kabel patch core lainnya sedang dipasang ke IPRAN (IP Radio Access Network). Pada tahap ini, kabel diposisikan dengan hati-hati dan terhubung dengan socket di IPRAN.

Gambar C: Kabel patch core SC dan LC/LC telah terpasang dengan baik. Pada tahap ini, kedua jenis kabel patch core sudah terhubung dengan benar di OTB dan IPRAN, memastikan konektivitas jaringan yang optimal.

Langkah berikutnya adalah menghubungkan IPRAN ke BBU menggunakan kabel patch core. Penghubungan ini penting untuk memastikan aliran data yang stabil dan konektivitas yang efisien antara IPRAN dan BBU, mendukung operasional sistem telekomunikasi dengan efektif.

Berikut langkah menghubungkan IPRAN ke BBU menggunakan Patchcore sesuai dengan gambar dibawah ini:



Gambar A: Kabel patch core sedang dipasang dari IPRAN ke BBU. Pada tahap ini, kabel dihubungkan untuk memastikan konektivitas antara kedua perangkat.

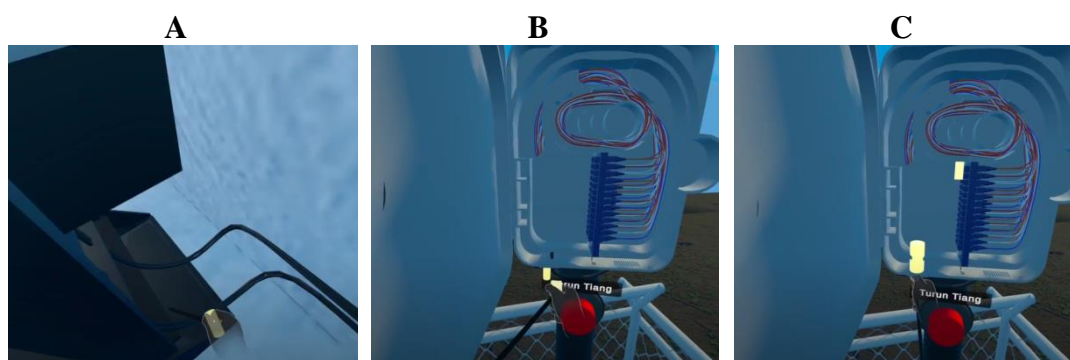
Gambar B: SFP sedang dipasang di BBU. Pada tahap ini, SFP disiapkan untuk

menerima koneksi dari kabel patch core.

Gambar C: Kabel patch core telah berhasil terpasang ke BBU. Pada tahap ini, kabel patch core terhubung dengan SFP di BBU, memastikan aliran data yang stabil antara IPRAN dan BBU.

Langkah berikutnya adalah memasang output OTB menuju ke input ODC. Pemasangan ini penting untuk memastikan aliran sinyal dari OTB ke ODC, memungkinkan data dan sinyal diteruskan dengan efektif dalam sistem jaringan.

Berikut langkah memasang kabel output otb ke input odc sesuai gambar di bawah ini:



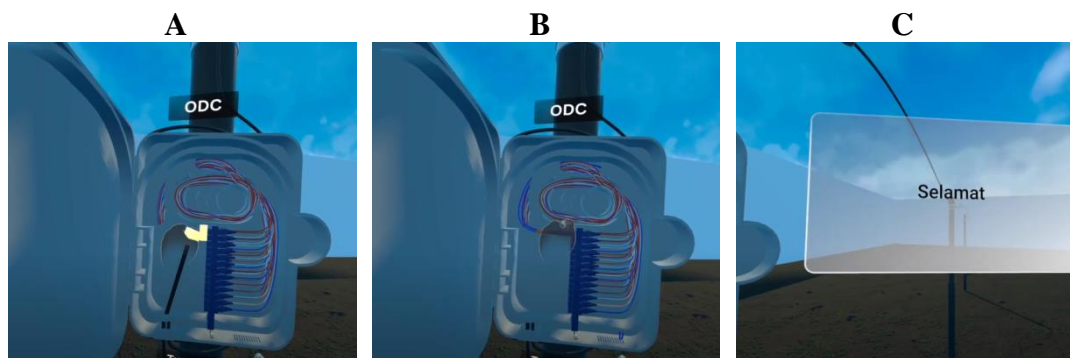
Gambar A: Kabel output OTB (ditampilkan sebagai simbol kabel siap untuk ditarik) berada dalam posisi siap untuk dipindahkan menuju input ODC. Pada tahap ini, kabel sudah disiapkan untuk dipasang.

Gambar B: Kabel output OTB sedang diposisikan dengan hati-hati dan siap untuk dipasang ke input ODC. Pada tahap ini, kabel dipersiapkan untuk dihubungkan dengan socket input di ODC.

Gambar C: Kabel output OTB telah berhasil terpasang di input ODC. Pada tahap ini, kabel sudah terhubung dengan baik, memastikan aliran sinyal dari OTB ke ODC berjalan dengan lancar.

Langkah terakhir adalah memasang output OTB yang berupa patch core ke ODC untuk menghubungkannya dengan site lain. Pemasangan ini penting untuk memastikan konektivitas antar-site, memungkinkan data dan komunikasi berjalan lancar antara lokasi yang berbeda dalam jaringan telekomunikasi.

Berikut merupakan langkah terakhir sesuai dengan gambar di bawah ini:



Gambar A: Kabel patch core (ditampilkan sebagai simbol kabel siap dipasang) berada dalam posisi siap untuk dipindahkan menuju ODC. Pada tahap ini, kabel sudah disiapkan untuk dihubungkan ke ODC.

Gambar B: Kabel patch core telah terpasang dengan baik di ODC. Pada tahap ini, kabel patch core sudah terhubung, memastikan konektivitas antara OTB dan ODC serta hubungan dengan site lain.

Gambar C: Canvas yang menandakan selesainya simulasi. Pada tahap ini, seluruh proses instalasi dan penghubungan telah selesai, menunjukkan bahwa site telah berhasil terhubung dan berfungsi dengan baik dalam jaringan.

## 1.6 Hasil Praktikum

*Jelaskan hasil praktikum berdasarkan perintah pada system simulasi praktikum*

## 1.7 Analisis

*Lakukan hasil praktikum berdasarkan perintah pada sistem simulasi praktikum. Berikan perbandingan antara analisis berdasarkan hasil simulasi dan teori.*

## **1.8 Kesimpulan dan Saran**

*Berikan kesimpulan berdasarkan tujuan percobaan*

**Koordinator Praktikum  
Sistem telekomunikasi**

(  
**D041**)

**Asisten Pembimbing**

(  
**D041**)

## Lampiran 6 Dokumentasi Proses Pengujian



