

DAFTAR LAMPIRAN

- 1. KODE PROGRAM**
- 2. PENGUJIAN BLACK BOX TESTING**

1. Kode Program

a. splashscreenactivity.kt

```
package com.example.bookingruang

import android.annotation.SuppressLint
import android.content.Intent
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import androidx.appcompat.app.AppCompatActivity

@SuppressLint("SplashScreen")
class SplashScreen : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_splash_screen)

        //mengecek aplikasi sudah pernah dibuka sebelumnya
        menggunakan sharedPreferences
        val preferences = getSharedPreferences("APP_PREFS",
        MODE_PRIVATE)
        val isFirstRun =
        preferences.getBoolean("FIRST_RUN", true)

        if (isFirstRun) {
            //set bahwa aplikasi sudah dibuka sebelumnya
            preferences.edit().putBoolean("FIRST_RUN",
            false).apply()
        }

        //delay menuju main activity 3detik
        Handler(Looper.getMainLooper()).postDelayed({
            //menentukan waktu splash screen 3 detik menuju
            page selanjutnya
            startActivity(Intent(this,
            MasuksebagaiActivity::class.java))
            finish()
        }, 3000)
    }
}
```

b. activity_splashscreen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#E4E4E4">

    <ImageView
        android:id="@+id/logo"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:src="@drawable/logoo1"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

c. masuksebagaiactivity.kt

```
package com.example.bookingruang

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity

class MasuksebagaiActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_masuk_sebagai) //
        Pastikan nama file XML-nya benar

        val btnDosen = findViewById<Button>(R.id.btn1)
        val btnMahasiswa = findViewById<Button>(R.id.btn2)
        val btnAdmin = findViewById<Button>(R.id.btn3)

        btnDosen.setOnClickListener {
            val intent = Intent(this,
LoginActivity::class.java)
            intent.putExtra("role", "DOSEN")
            startActivity(intent)
            finish()
        }

        btnMahasiswa.setOnClickListener {
            val intent = Intent(this,
LoginActivity::class.java)
            intent.putExtra("role", "MAHASISWA")
            startActivity(intent)
            finish()
        }

        btnAdmin.setOnClickListener {
            val intent = Intent(this,
LoginActivity::class.java)
            intent.putExtra("role", "ADMIN")
            startActivity(intent)
            finish()
        }
    }
}
```

d. activity_masuk_sebagai.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:background="#E4E4E4">

    <ImageView
        android:id="@+id/img1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="200dp"
        android:src="@drawable/logoo1"/>

    <ImageView
        android:id="@+id/img2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dp"
        android:src="@drawable/masuksebagai"/>

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btn1"
        android:layout_width="200dp"
        android:layout_height="50dp"
        android:layout_marginTop="40dp"
        android:background="@drawable/button_background"
        android:text="DOSEN"
        android:textSize="18sp"/>

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btn2"
        android:layout_width="200dp"
        android:layout_height="50dp"
        android:layout_marginTop="40dp"
        android:background="@drawable/button_background"
        android:text="MAHASISWA"
        android:textSize="18sp"/>

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btn3"
        android:layout_width="200dp"
        android:layout_height="50dp"
        android:layout_marginTop="40dp"
        android:background="@drawable/button_background"
        android:text="ADMIN"
        android:textSize="18sp"/>

</LinearLayout>
```

e. loginactivity.kt

```
package com.example.bookingruang

import android.content.Intent
import android.os.Bundle
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.appcompat.widget.AppCompatButton
import com.google.android.material.textfield.TextInputEditText
import com.google.firebase.firestore.FirebaseFirestore

class LoginActivity : AppCompatActivity() {

    private lateinit var db: FirebaseFirestore
    private lateinit var inputUsername: TextInputEditText
    private lateinit var inputPassword: TextInputEditText
    private lateinit var loginButton: AppCompatButton

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        db = FirebaseFirestore.getInstance()
        inputUsername = findViewById(R.id.hintusername)
        inputPassword = findViewById(R.id.hintpassword)
        loginButton = findViewById(R.id.masuk)

        loginButton.setOnClickListener {
            val username =
                inputUsername.text.toString().trim()
            val password =
                inputPassword.text.toString().trim()

            if (username.isEmpty() || password.isEmpty()) {
                Toast.makeText(this, "Harap isi username
                dan password!", Toast.LENGTH_SHORT).show()
                return@setOnClickListener
            }

            loginUser(username, password)
        }

        private fun loginUser(username: String, password:
        String) {
            db.collection("users")
                .whereEqualTo("username", username)
                .whereEqualTo("password", password)
                .get()
                .addOnSuccessListener { documents ->
                    if (!documents.isEmpty) {
                        val user = documents.documents[0]
                        val role = user.getString("role") ?: ""

                        Toast.makeText(this, "Login berhasil
                        sebagai $role", Toast.LENGTH_SHORT).show()

                        // Kirim username & role ke
```

```

MainActivity.kt
        val intent = Intent(this,
MainActivity::class.java)
        intent.putExtra("username", username)
        intent.putExtra("role",
role.uppercase()) // Pastikan huruf besar
        startActivity(intent)
        finish()
    } else {
        Toast.makeText(this, "Login gagal!
Username atau password salah", Toast.LENGTH_SHORT).show()
    }
}
    .addOnFailureListener { e ->
        Toast.makeText(this, "Login gagal!
${e.message}", Toast.LENGTH_SHORT).show()
    }
}
}

```

f. activity_login.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#E4E4E4">

    <ImageView
        android:id="@+id/img1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/atas"
        android:scaleType="fitStart"/>

    <TextView
        android:id="@+id/Silahkanmasuk"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Silahkan Masuk!!"
        android:fontFamily="sans-serif-condensed"
        android:textSize="30dp"
        android:layout_below="@+id/img1"
        android:layout_centerHorizontal="true"
        android:textColor="@color/black"/>

    <TextView
        android:id="@+id/username"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/Silahkanmasuk"
        android:layout_alignParentLeft="true"
        android:layout_marginLeft="68dp"
        android:layout_marginTop="35dp"
        android:fontFamily="times"
        android:text="Username"

```

```

        android:textColor="@color/black"
        android:textSize="18dp" />

<com.google.android.material.textfield.TextInputEditText
    android:id="@+id/hintusername"
    android:hint="@string/hint_username"
    android:textColorHint="#808588"
    android:layout_width="300dp"
    android:layout_height="48dp"
    android:layout_below="@+id/username"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10px"
    android:background="@drawable/custom_login"
    android:fontFamily="times"
    android:paddingLeft="20dp"
    android:textColor="#808588"
    android:textSize="18dp"/>

<TextView
    android:id="@+id/text2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/hintusername"
    android:layout_alignParentLeft="true"
    android:layout_marginLeft="68dp"
    android:layout_marginTop="10dp"
    android:fontFamily="times"
    android:text="Password"
    android:textColor="@color/black"
    android:textSize="18dp" />

<com.google.android.material.textfield.TextInputEditText
    android:id="@+id/hintpassword"
    android:hint="@string/hint_password"
    android:textColorHint="#808688"
    android:layout_width="300dp"
    android:layout_height="48dp"
    android:layout_below="@+id/text2"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10px"
    android:background="@drawable/custom_login"
    android:fontFamily="times"
    android:paddingLeft="20dp"
    android:textColor="#808688"
    android:textSize="18dp"/>

<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/masuk"
    android:layout_width="150dp"
    android:layout_height="150px"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="100px"
    android:layout_below="@+id/hintpassword"
    android:text="MASUK"
    android:background="@drawable/button_masuk"/>

<ImageView
    android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:src="@drawable/bawah"
        android:layout_alignParentBottom="true"
        android:scaleType="fitEnd"/>
</RelativeLayout>

```

g. **mainactivity.kt**

```

package com.example.bookingruang

import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.content.IntentFilter
import android.os.Build
import android.os.Bundle
import android.util.Log
import android.widget.Button
import android.widget.GridLayout
import android.widget.ImageButton
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.content.ContextCompat
import com.google.firebase.firestore.FirebaseFirestore
import java.text.SimpleDateFormat
import java.util.*
import java.util.concurrent.CountDownLatch

class MainActivity : AppCompatActivity() {

    private lateinit var tvCurrentTime: TextView
    private lateinit var tvUsername: TextView
    private lateinit var classroomGrid: GridLayout
    private lateinit var btnPengaturan: ImageButton

    private lateinit var currentLoggedInUsername: String
    private lateinit var currentUserRole: String
    private lateinit var db: FirebaseFirestore

    // Handler untuk update waktu
    private val timeUpdateReceiver = object :
BroadcastReceiver() {
        override fun onReceive(context: Context?, intent:
Intent?) {
            if (intent?.action ==
"com.example.bookingruang.TIME_UPDATE") {
                val currentDateTime =
SimpleDateFormat("HH:mm:ss EEEE, dd MMMM yyyy",
Locale("id", "ID")).format(Date())
                tvCurrentTime.text = currentDateTime
                createDynamicRoomButtons()
            }
        }
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}

```

```

        Log.d("MainActivity", "onCreate: MainActivity
dimulai.")
        db = FirebaseFirestore.getInstance()

        // Inisialisasi View
        try {
            tvCurrentTime =
findViewById(R.id.timeDateTextView)
            tvUsername = findViewById(R.id.tvUsername)
            btnPengaturan =
findViewById(R.id.btnPengaturan)
            classroomGrid =
findViewById(R.id.classroomGrid)
            Log.d("MainActivity", "onCreate: Semua Views
berhasil diinisialisasi.")
        } catch (e: Exception) {
            Log.e("MainActivity", "onCreate: Gagal
menginisialisasi Views.", e)
            Toast.makeText(this, "Terjadi kesalahan saat
memuat tampilan. Aplikasi akan ditutup.",
Toast.LENGTH_LONG).show()
            finish()
            return
        }

        currentLoggedInUsername =
intent.getStringExtra("username") ?: "Pengguna"
        currentUserRole = intent.getStringExtra("role") ?:
""
        Log.d("MainActivity", "onCreate: Pengguna yang
login: $currentLoggedInUsername dengan role:
$currentUserRole")

        tvUsername.text = "Selamat Datang,
$currentLoggedInUsername ($currentUserRole)"

        val filter =
IntentFilter("com.example.bookingruang.TIME_UPDATE")
        if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.TIRAMISU) {
            registerReceiver(timeUpdateReceiver, filter,
Context.RECEIVER_NOT_EXPORTED)
            Log.d("MainActivity", "onCreate:
BroadcastReceiver didaftarkan (API 33+).")
        } else {
            @Suppress("DEPRECATION")
            registerReceiver(timeUpdateReceiver, filter)
            Log.d("MainActivity", "onCreate:
BroadcastReceiver didaftarkan (API < 33).")
        }

        // Set waktu awal
        val initialDateTime = SimpleDateFormat("HH:mm:ss
EEEE, dd MMMM yyyy", Locale("id", "ID")).format(Date())
        tvCurrentTime.text = initialDateTime
        Log.d("MainActivity", "onCreate: Waktu awal diatur:
$initialDateTime")

```

```

        btnPengaturan.setOnClickListener {
            Log.d("MainActivity", "OnClickListener: Tombol
Pengaturan diklik.")
            val intent = Intent(this,
SettingActivity::class.java)
            intent.putExtra("username",
currentLoggedInUsername)
            startActivity(intent)
        }

        Log.d("MainActivity", "onCreate: Proses onCreate
selesai.")
    }

    override fun onResume() {
        super.onResume()
        Log.d("MainActivity", "onResume: MainActivity
kembali aktif. Memperbarui tombol ruangan.")
        createDynamicRoomButtons()
    }

    override fun onDestroy() {
        super.onDestroy()
        unregisterReceiver(timeUpdateReceiver)
        Log.d("MainActivity", "onDestroy: MainActivity
dihancurkan. BroadcastReceiver dibatalkan pendaftarannya.")
    }

    private fun createDynamicRoomButtons() {
        Log.d("MainActivity", "createDynamicRoomButtons:
Memulai pembuatan tombol ruangan dinamis.")
        val roomNames = listOf(
            "Ruang A5", "Ruang A6", "Ruang A7",
            "Ruang A8", "Ruang B1", "Ruang B2",
            "Ruang B3", "Ruang B5", "Ruang B6",
            "Ruang B7", "Ruang B8", "Ruang C1",
            "Ruang C2", "Ruang Aula 1", "Ruang E2",
            "Ruang E3", "Ruang E4", "Ruang F2",
            "Ruang F3", "Ruang Aula 2", "Lab Kom",
            "Lab IoT", "Lab Micro", "Lab Bahasa",
            "Lab Jar"
        )

        val displayMetrics = resources.displayMetrics
        val desiredButtonWidthDp = 110
        val buttonWidthPx = (desiredButtonWidthDp *
displayMetrics.density).toInt()
        val desiredButtonHeightDp = 50
        val buttonHeightPx = (desiredButtonHeightDp *
displayMetrics.density).toInt()
        val marginDp = 4
        val marginPx = (marginDp *
displayMetrics.density).toInt()

        // Hapus semua tombol yang ada
        classroomGrid.removeAllViews()
        Log.d("MainActivity", "createDynamicRoomButtons:
Menghapus semua tombol lama dari GridLayout.")
    }

```

```

        // Map untuk menyimpan status booking setiap
ruangan
        val roomStatusMap = mutableMapOf<String,
Triple<Boolean, String, String>>()

        // CountdownLatch untuk menunggu semua query
selesai
        val latch = CountdownLatch(roomNames.size)
        Log.d("MainActivity", "createDynamicRoomButtons:
Total query Firebase yang ditunggu: ${roomNames.size}.")

        roomNames.forEach { roomName ->
            db.collection("bookings")
                .whereEqualTo("roomName", roomName)
                .whereEqualTo("status", "aktif")
                .get()
                .addOnSuccessListener { documents ->
                    val currentTimeString =
SimpleDateFormat("HH:mm",
Locale.getDefault()).format(Date())
                    val currentTime =
SimpleDateFormat("HH:mm",
Locale.getDefault()).parse(currentTimeString)
                    val todayDate = SimpleDateFormat("yyyy-
MM-dd", Locale.getDefault()).format(Date())

                    var hasActiveBooking = false
                    var lecturerName = ""
                    var subjectName = ""

                    for (document in documents) {
                        val startTimeStr =
document.getString("startTime")
                        val endTimeStr =
document.getString("endTime")
                        val timestamp =
document.getLong("timestamp") ?: 0L
                        val bookingDate =
SimpleDateFormat("yyyy-MM-dd",
Locale.getDefault()).format(Date(timestamp))

                        if (bookingDate == todayDate) {
                            try {
                                val startTime =
SimpleDateFormat("HH:mm",
Locale.getDefault()).parse(startTimeStr ?: "00:00")
                                val endTime =
SimpleDateFormat("HH:mm",
Locale.getDefault()).parse(endTimeStr ?: "00:00")

                                if (currentTime != null &&
startTime != null && endTime != null &&
currentTime.after(startTime) &&
currentTime.before(endTime)) {
                                    hasActiveBooking = true
                                    lecturerName =
document.getString("lecturerName") ?: "Tidak diketahui"
                                    subjectName =

```

```

document.getString("subject") ?: "Tidak diketahui"
        Log.d("MainActivity",
"createDynamicRoomButtons: Ruangn $roomName aktif oleh
$lecturerName - $subjectName.")
        break
    }
    } catch (e: Exception) {
        Log.e("MainActivity",
"Error parsing waktu untuk status ruangan $roomName:
${e.message}", e)
    }
}

roomStatusMap[roomName] =
Triple(hasActiveBooking, lecturerName, subjectName)
latch.countDown()
}
.addOnFailureListener { e ->
    roomStatusMap[roomName] = Triple(false,
"", "")
    latch.countDown()
    Log.e("MainActivity", "Error memuat
status ruangan untuk $roomName: ${e.message}", e)
}
}

// Jalankan di thread terpisah
Thread {
    try {
        latch.await()
        Log.d("MainActivity",
"createDynamicRoomButtons: Semua query Firebase selesai.
Membangun tombol.")

        runOnUiThread {
            roomNames.forEach { roomName ->
                val status =
roomStatusMap[roomName]
                val hasActiveBooking =
status?.first ?: false

                val button = Button(this).apply {
                    val params =
GridLayout.LayoutParams().apply {
                        width = buttonWidthPx
                        height = buttonHeightPx
                        setMargins(marginPx,
marginPx, marginPx, marginPx)
                    }
                    layoutParams = params
                    text = roomName
                    setBackgroundColor(
                        ContextCompat.getColor(
                            this@MainActivity,
                            if (hasActiveBooking)
R.color.red_used else R.color.blue_empty
                        )
                    )
                }
            }
        }
    }
}

```

```

setTextColor (ContextCompat.getColor(this@MainActivity,
R.color.white))
    }

    classroomGrid.addView(button)

    // OnClickListener untuk tombol
    button.setOnClickListener {
        Log.d("MainActivity",
"OnClickListener: Tombol $roomName diklik. Role:
$currentUserRole, Active Booking: $hasActiveBooking")
        handleRoomButtonClick(roomName,
hasActiveBooking)
    }
    }
    Log.d("MainActivity",
"createDynamicRoomButtons: Semua tombol berhasil
ditambahkan ke GridLayout.")
    }
    } catch (e: InterruptedException) {
        Thread.currentThread().interrupt()
        Log.e("MainActivity", "Thread terinterupsi
saat menunggu status ruangan: ${e.message}", e)
        runOnUiThread {
            Toast.makeText(this@MainActivity,
"Gagal memuat status ruangan: ${e.message}",
Toast.LENGTH_LONG).show()
        }
    }
    }.start()
}

private fun handleRoomButtonClick(roomName: String,
hasActiveBooking: Boolean) {
    // PEMBATASAN AKSES UNTUK MAHASISWA
    if (currentUserRole.uppercase() == "MAHASISWA") {
        if (hasActiveBooking) {
            // Mahasiswa hanya boleh melihat status
ruangan yang sedang dipakai (merah)
            showRoomStatusForStudent(roomName)
        } else {
            // Ruangan kosong (biru), mahasiswa tidak
bisa melakukan booking
            Toast.makeText(this, "Maaf, Anda tidak
memiliki izin untuk melakukan booking ruangan.",
Toast.LENGTH_LONG).show()
            Log.d("MainActivity", "Mahasiswa mencoba
booking ruangan kosong. Izin ditolak.")
        }
        return
    }

    // LOGIKA UNTUK DOSEN DAN ADMIN
    checkRoomStatusAndRedirect(roomName)
}

private fun showRoomStatusForStudent(roomName: String)
{

```

```

        db.collection("bookings")
            .whereEqualTo("roomName", roomName)
            .whereEqualTo("status", "aktif")
            .get()
            .addOnSuccessListener { documents ->
                val currentTimeString =
                    SimpleDateFormat("HH:mm",
                        Locale.getDefault()).format(Date())
                val currentTime = SimpleDateFormat("HH:mm",
                    Locale.getDefault()).parse(currentTimeString)
                val todayDate = SimpleDateFormat("yyyy-MM-
                    dd", Locale.getDefault()).format(Date())

                var lecturerName = ""
                var subjectName = ""
                var remainingMinutes = 0

                for (document in documents) {
                    val startTimeStr =
                        document.getString("startTime")
                    val endTimeStr =
                        document.getString("endTime")
                    val timestamp =
                        document.getLong("timestamp") ?: 0L
                    val bookingDate =
                        SimpleDateFormat("yyyy-MM-dd",
                            Locale.getDefault()).format(Date(timestamp))

                    if (bookingDate == todayDate) {
                        try {
                            val startTime =
                                SimpleDateFormat("HH:mm",
                                    Locale.getDefault()).parse(startTimeStr ?: "00:00")
                            val endTime =
                                SimpleDateFormat("HH:mm",
                                    Locale.getDefault()).parse(endTimeStr ?: "00:00")

                            if (currentTime != null &&
                                startTime != null && endTime != null &&
                                currentTime.after(startTime) &&
                                currentTime.before(endTime)) {
                                lecturerName =
                                    document.getString("lecturerName") ?: "Tidak diketahui"
                                subjectName =
                                    document.getString("subject") ?: "Tidak diketahui"
                                remainingMinutes =
                                    ((endTime.time - currentTime.time) / (1000 * 60)).toInt()
                                Log.d("MainActivity",
                                    "Detail booking aktif untuk $roomName (Mahasiswa):
                                    $lecturerName, $subjectName, $remainingMinutes menit.")
                                break
                            }
                        } catch (e: Exception) {
                            Log.e("MainActivity", "Error
                                parsing waktu untuk status ruangan saat klik (Mahasiswa):
                                ${e.message}", e)
                        }
                    }
                }
            }
    }
}

```

```

    }

    val intent = Intent(this@MainActivity,
RoomStatusActivity::class.java)
    intent.putExtra("ROOM_NAME", roomName)
    intent.putExtra("LECTURER_NAME",
lecturerName)
    intent.putExtra("SUBJECT_NAME",
subjectName)
    intent.putExtra("REMAINING_TIME",
remainingMinutes)
    startActivity(intent)
    }
    .addOnFailureListener { e ->
        Toast.makeText(this@MainActivity, "Gagal
memuat detail status ruangan: ${e.message}",
Toast.LENGTH_LONG).show()
        Log.e("MainActivity", "Error mendapatkan
status ruangan untuk RoomStatusActivity (Mahasiswa)", e)
    }
    }

    private fun checkRoomStatusAndRedirect(roomName:
String) {
        db.collection("bookings")
            .whereEqualTo("roomName", roomName)
            .whereEqualTo("status", "aktif")
            .get()
            .addOnSuccessListener { documents ->
                val currentTimeString =
SimpleDateFormat("HH:mm",
Locale.getDefault()).format(Date())
                val currentTime = SimpleDateFormat("HH:mm",
Locale.getDefault()).parse(currentTimeString)
                val todayDate = SimpleDateFormat("yyyy-MM-
dd", Locale.getDefault()).format(Date())

                var hasActiveBooking = false
                var lecturerName = ""
                var subjectName = ""
                var remainingMinutes = 0

                for (document in documents) {
                    val startTimeStr =
document.getString("startTime")
                    val endTimeStr =
document.getString("endTime")
                    val timestamp =
document.getLong("timestamp") ?: 0L
                    val bookingDate =
SimpleDateFormat("yyyy-MM-dd",
Locale.getDefault()).format(Date(timestamp))

                    if (bookingDate == todayDate) {
                        try {
                            val startTime =
SimpleDateFormat("HH:mm",
Locale.getDefault()).parse(startTimeStr ?: "00:00")
                            val endTime =

```

```

SimpleDateFormat("HH:mm",
Locale.getDefault()).parse(endTimeStr ?: "00:00")

        if (currentTime != null &&
startTime != null && endTime != null &&

currentTime.after(startTime) &&
currentTime.before(endTime)) {

                hasActiveBooking = true
                lecturerName =
document.getString("lecturerName") ?: "Tidak diketahui"
                subjectName =
document.getString("subject") ?: "Tidak diketahui"
                remainingMinutes =
((endTime.time - currentTime.time) / (1000 * 60)).toInt()
                Log.d("MainActivity",
"Detail booking aktif untuk $roomName (Dosen/Admin):
$lecturerName, $subjectName, $remainingMinutes menit.")
                break
            }
        } catch (e: Exception) {
            Log.e("MainActivity", "Error
parsing waktu untuk status ruangan saat klik (Dosen/Admin):
${e.message}", e)
        }
    }

    if (hasActiveBooking) {
        // Ruangan sedang dipakai, tampilkan
status
        val intent = Intent(this@MainActivity,
RoomStatusActivity::class.java)
        intent.putExtra("ROOM_NAME", roomName)
        intent.putExtra("LECTURER_NAME",
lecturerName)
        intent.putExtra("SUBJECT_NAME",
subjectName)
        intent.putExtra("REMAINING_TIME",
remainingMinutes)
        startActivity(intent)
    } else {
        // Ruangan kosong, arahkan ke
BookingActivity (hanya untuk DOSEN dan ADMIN)
        val intent = Intent(this@MainActivity,
BookingActivity::class.java)
        intent.putExtra("ROOM_NAME", roomName)
        intent.putExtra("LECTURER_NAME",
currentLoggedInUsername)
        intent.putExtra("CURRENT_USER_ROLE",
currentUserRole)
        startActivity(intent)
    }
}
.addOnFailureListener { e ->
    Toast.makeText(this@MainActivity, "Error
checking room status: ${e.message}",
Toast.LENGTH_LONG).show()
    Log.e("MainActivity", "Error checking room

```

```

status on click", e)
        }
    }
}

```

h. activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#E4E4E4">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <ImageView
            android:id="@+id/img1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/atas"
            android:scaleType="fitStart"/>

        <ImageButton
            android:id="@+id/btnPengaturan"
            android:layout_width="48dp"
            android:layout_height="50dp"
            android:layout_gravity="start|top"
            android:layout_marginTop="55dp"
            android:layout_marginStart="8dp"
            android:background="@android:color/transparent"
            android:contentDescription="Pengaturan"
            android:src="@drawable/baseline_settings_24" />

        <TextView
            android:id="@+id/text1"
            android:layout_width="200dp"
            android:layout_height="50dp"
            android:text="RUANG KELAS"
            android:textColor="@color/white"
            android:layout_marginTop="40dp"
            android:gravity="center"
            android:textSize="24sp"
            android:textStyle="bold"
            android:layout_gravity="center"
            android:padding="8dp"
            android:background="@drawable/button_masuk"/>

        <TextView
            android:id="@+id/timeDateTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:padding="16dp"
            android:layout_marginTop="100dp"
            android:textColor="#00aeb4"
            android:textStyle="bold"

```

```

        android:text="12:15:45\nSenin, 02 Oktober 2024"
        android:textSize="16sp" />

        <TextView
            android:id="@+id/tvUsername"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="top|end"
            android:layout_marginTop="35dp"
            android:layout_marginEnd="8dp"
            android:text="Selamat Datang, Pengguna"
            android:textColor="@color/black"
            android:textSize="14sp" />

    </FrameLayout>

    <GridLayout
        android:id="@+id/classroomGrid"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:columnCount="3"
        android:rowCount="8"
        android:layout_gravity="center"
        android:padding="15dp">
    </GridLayout>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="1dp"
        android:text="* Warna merah menunjukkan ruang kelas
terpakai"
        android:textSize="18sp"
        android:textStyle="italic"
        android:layout_gravity="center"
        android:padding="8dp"
        android:textColor="@color/red_used"/>

</LinearLayout>

```

i. `settingactivity.kt`

```
package com.example.bookingruang

import android.content.Context
import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class SettingActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_setting) //
        Menggunakan layout activity_setting.xml

        val btnEditProfil =
            findViewById<Button>(R.id.btnEditProfil)
        val btnLogout =
            findViewById<Button>(R.id.btnLogout)

        btnEditProfil.setOnClickListener {
            val username =
                intent.getStringExtra("username")
            if (username != null) {
                val intent = Intent(this,
                    EditProfilActivity::class.java)
                intent.putExtra("username", username)
                startActivity(intent)
            } else {
                Toast.makeText(this, "Username tidak
                    ditemukan. Gagal membuka Edit Profil.",
                    Toast.LENGTH_SHORT).show()
            }
        }

        btnLogout.setOnClickListener {
            Toast.makeText(this, "Anda telah logout.",
                Toast.LENGTH_SHORT).show()
            val intent = Intent(this,
                LoginActivity::class.java)
            intent.flags = Intent.FLAG_ACTIVITY_NEW_TASK or
                Intent.FLAG_ACTIVITY_CLEAR_TASK
            startActivity(intent)
            finish()
        }
    }
}
```

j. activity_setting.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/settingLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="24dp"
    android:background="@android:color/white">

    <ImageView
        android:id="@+id/settingTitle"
        android:layout_width="45dp"
        android:layout_height="45dp"
        android:layout_gravity="end"
        android:layout_margin="8dp"
        android:background="@android:color/transparent"
        android:contentDescription="Pengaturan"
        android:src="@drawable/baseline_settings_24" />

    <Button
        android:id="@+id/btnEditProfil"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Edit Profil"
        android:layout_marginBottom="16dp" />

    <Button
        android:id="@+id/btnLogout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Logout"
        android:background="@drawable/button_masuk" />
</LinearLayout>
```

k. editprofilactivity.kt

```
package com.example.bookingruang

import android.app.Activity // Pastikan ini diimport
import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.util.Log
import android.widget.Button
import android.widget.EditText
import android.widget.ImageView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.firestore.FirebaseFirestore
import com.google.firebase.firestore.SetOptions

// Import yang diperlukan untuk ActivityResultLauncher
import androidx.activity.result.ActivityResultLauncher
import androidx.activity.result.contract.ActivityResultContracts

class EditProfilActivity : AppCompatActivity() {
```

```

        // Hapus val PICK_IMAGE_REQUEST = 1 karena tidak lagi
        diperlukan dengan ActivityResultLauncher

        private lateinit var etNama: EditText
        private lateinit var etNIM: EditText
        private lateinit var etProdi: EditText
        private lateinit var etAngkatan: EditText
        private lateinit var btnSimpan: Button
        private lateinit var imgFotoProfil: ImageView

        private lateinit var db: FirebaseFirestore

        private var loggedInUsername: String = ""
        private var currentUserDocId: String? = null

        // Deklarasikan ActivityResultLauncher sebagai member
        dari class
        // Ini harus dideklarasikan di scope class, bukan di
        dalam onCreate
        private lateinit var pickImageLauncher:
        ActivityResultLauncher<Intent>

        override fun onCreate(savedInstanceState: Bundle?) {
            super.onCreate(savedInstanceState)
            setContentView(R.layout.activity_edit_profil)

            db = FirebaseFirestore.getInstance()

            loggedInUsername =
            intent.getStringExtra("username") ?: ""

            if (loggedInUsername.isEmpty()) {
                Toast.makeText(this, "Username tidak ditemukan.
                Gagal memuat profil.", Toast.LENGTH_LONG).show()
                val intent = Intent(this,
                LoginActivity::class.java)
                intent.flags = Intent.FLAG_ACTIVITY_NEW_TASK or
                Intent.FLAG_ACTIVITY_CLEAR_TASK
                startActivity(intent)
                finish()
                return
            }

            etNama = findViewById(R.id.etNama)
            etNIM = findViewById(R.id.etNIM)
            etProdi = findViewById(R.id.etProdi)
            etAngkatan = findViewById(R.id.etAngkatan)
            btnSimpan = findViewById(R.id.btnSimpan)
            imgFotoProfil = findViewById(R.id.imgFotoProfil)

            // Inisialisasi ActivityResultLauncher
            // Lakukan ini di dalam onCreate atau di
            inisialisasi properti langsung
            pickImageLauncher =
            registerForActivityResult(ActivityResultContracts.StartActi
            vityForResult()) { result ->
                if (result.resultCode == Activity.RESULT_OK) {
                    val imageUri = result.data?.data // Data

```

```

yang dikembalikan ada di result.data
        if (imageUri != null) {
            imgFotoProfil.setImageURI(imageUri)
            savePhotoUriToFirestore(imageUri)
        } else {
            Toast.makeText(this, "Tidak ada gambar
yang dipilih.", Toast.LENGTH_SHORT).show()
        }
    }

loadUserProfile(loggedInUsername)

imgFotoProfil.setOnClickListener {
    val intent = Intent(Intent.ACTION_PICK)
    intent.type = "image/*"
    // Panggil .launch() pada
    ActivityResultLauncher yang sudah dideklarasikan
    pickImageLauncher.launch(intent)
}

btnSimpan.setOnClickListener {
    saveUserProfile()
}

private fun loadUserProfile(username: String) {
    db.collection("users")
        .whereEqualTo("username", username)
        .get()
        .addOnSuccessListener { documents ->
            if (!documents.isEmpty) {
                val userDoc = documents.first()
                currentUserDocId = userDoc.id

                val nama = userDoc.getString("nama") ?:
                ""
                val nim = userDoc.getString("nim") ?:
                ""
                val prodi = userDoc.getString("prodi")
                ?: ""
                val angkatan =
                userDoc.getString("angkatan") ?: ""
                val fotoUriString =
                userDoc.getString("fotoUri")

                etNama.setText(nama)
                etNIM.setText(nim)
                etProdi.setText(prodi)
                etAngkatan.setText(angkatan)

                if (!fotoUriString.isNullOrEmpty()) {
                    try {
                        val uri =
                        Uri.parse(fotoUriString)
                        imgFotoProfil.setImageURI(uri)
                    } catch (e: Exception) {
                        Log.e("EditProfilActivity",
                        "Gagal memuat foto dari URI: ${e.message}", e)
                    }
                }
            }
        }
    }
}

```

```

                                Toast.makeText(this, "Gagal
memuat foto profil.", Toast.LENGTH_SHORT).show()
                                }
                                }
                                Log.d("EditProfilActivity", "Data
profil berhasil dimuat untuk username: $username")
                                } else {
                                    Toast.makeText(this, "Profil tidak
ditemukan untuk $username. Silakan isi data Anda.",
Toast.LENGTH_LONG).show()
                                    Log.d("EditProfilActivity", "Dokumen
profil tidak ditemukan untuk username: $username")
                                    }
                                }
                                .addOnFailureListener { e ->
                                    Toast.makeText(this, "Gagal memuat data
profil: ${e.message}", Toast.LENGTH_LONG).show()
                                    Log.e("EditProfilActivity", "Error memuat
data profil", e)
                                }
                                }

                                private fun saveUserProfile() {
                                    val nama = etNama.text.toString().trim()
                                    val nim = etNIM.text.toString().trim()
                                    val prodi = etProdi.text.toString().trim()
                                    val angkatan = etAngkatan.text.toString().trim()

                                    if (nama.isEmpty() || nim.isEmpty() ||
prodi.isEmpty() || angkatan.isEmpty()) {
                                        Toast.makeText(this, "Harap isi semua field!",
Toast.LENGTH_SHORT).show()
                                        return
                                    }

                                    val userData = hashMapOf(
                                        "nama" to nama,
                                        "nim" to nim,
                                        "prodi" to prodi,
                                        "angkatan" to angkatan
                                    )

                                    if (currentUserDocId != null) {
                                        db.collection("users").document(currentUserDocId!!)
                                            .set(userData, SetOptions.merge())
                                            .addOnSuccessListener {
                                                Toast.makeText(this, "Data profil
berhasil disimpan!", Toast.LENGTH_SHORT).show()
                                                Log.d("EditProfilActivity", "Data
profil berhasil disimpan untuk ID dokumen:
$currentUserDocId")
                                                finish()
                                            }
                                            .addOnFailureListener { e ->
                                                Toast.makeText(this, "Gagal menyimpan
data profil: ${e.message}", Toast.LENGTH_LONG).show()
                                                Log.e("EditProfilActivity", "Error
menyimpan data profil", e)

```

```

    }
    } else {
        Toast.makeText(this, "Tidak dapat menemukan
profil untuk diupdate. Coba login ulang.",
Toast.LENGTH_LONG).show()
        Log.e("EditProfilActivity", "currentUserDocId
is null, cannot save profile.")
    }
}

private fun savePhotoUriToFirestore(uri: Uri) {
    if (currentUserDocId != null) {
        val photoData = hashMapOf(
            "fotoUri" to uri.toString()
        )

db.collection("users").document(currentUserDocId!!)
    .set(photoData, SetOptions.merge())
    .addOnSuccessListener {
        Toast.makeText(this, "Foto profil
berhasil diperbarui!", Toast.LENGTH_SHORT).show()
        Log.d("EditProfilActivity", "Foto
profil berhasil disimpan untuk ID dokumen:
$currentUserDocId")
    }
    .addOnFailureListener { e ->
        Toast.makeText(this, "Gagal memperbarui
foto profil: ${e.message}", Toast.LENGTH_LONG).show()
        Log.e("EditProfilActivity", "Error
menyimpan foto profil", e)
    }
} else {
    Toast.makeText(this, "Tidak dapat menemukan
profil untuk memperbarui foto.", Toast.LENGTH_LONG).show()
    Log.e("EditProfilActivity", "currentUserDocId
is null, cannot save photo.")
}
}
}

```

1. activity_edit_profil.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:background="#E0E0E0">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="center_horizontal">

        <!-- Foto Profil -->
        <ImageView
            android:id="@+id/img1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:scaleType="fitStart"
            android:src="@drawable/atas" />

        <ImageView
            android:id="@+id/imgFotoProfil"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:src="@drawable/baseline_person_24"

            android:background="@drawable/circle_background"
            android:scaleType="centerCrop" />

        <!-- Icon Pengaturan (kecil di kanan bawah) -->
        <ImageButton
            android:layout_width="48dp"
            android:layout_height="48dp"
            android:contentDescription="Pengaturan profil"
            android:layout_marginTop="-30dp"
            android:layout_marginStart="60dp"
            android:padding="12dp"
            app:tint="#000000"
            android:src="@drawable/baseline_settings_24" />

        <!-- Card Form -->
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginTop="2dp"
            android:padding="24dp"
            android:background="@drawable/card_background"
            android:elevation="10dp">

            <!-- Nama -->
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal"
                android:layout_marginBottom="12dp">
```

```

        <TextView
            android:layout_width="0dp"
            android:layout_weight="1"
            android:layout_height="wrap_content"
            android:text="Nama"
            android:textColor="@color/black"

android:background="@drawable/input_border"
            android:gravity="center"
            android:padding="12dp"
            android:textStyle="bold"/>

        <EditText
            android:id="@+id/etNama"
            android:layout_width="0dp"
            android:layout_weight="2"
            android:layout_height="wrap_content"

android:background="@drawable/input_border"
            android:padding="12dp"
            android:textColor="@color/black"
            android:text="Cahyaning Yuniar
Pramudita"/>
    </LinearLayout>

    <!-- NIM -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:textColor="@color/black"
        android:layout_marginBottom="12dp">

        <TextView
            android:layout_width="0dp"
            android:layout_weight="1"
            android:layout_height="wrap_content"
            android:text="NIM"

android:background="@drawable/input_border"
            android:gravity="center"
            android:padding="12dp"
            android:textColor="@color/black"
            android:textStyle="bold"/>

        <EditText
            android:id="@+id/etNIM"
            android:layout_width="0dp"
            android:layout_weight="2"
            android:layout_height="wrap_content"

android:background="@drawable/input_border"
            android:padding="12dp"
            android:textColor="@color/black"
            android:text="202221043"/>
    </LinearLayout>

    <!-- Fakultas / Prodi -->

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginBottom="12dp">

    <TextView
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:text="Fak/Prodi"

    android:background="@drawable/input_border"
        android:gravity="center"
        android:padding="12dp"
        android:textColor="@color/black"
        android:textStyle="bold"/>

    <EditText
        android:id="@+id/etProdi"
        android:layout_width="0dp"
        android:layout_weight="2"
        android:layout_height="wrap_content"

    android:background="@drawable/input_border"
        android:padding="12dp"
        android:textColor="@color/black"
        android:text="FMIKOM/ Informatika"/>
</LinearLayout>

<!-- Angkatan -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginBottom="12dp">

    <TextView
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:text="Angkatan"

    android:background="@drawable/input_border"
        android:gravity="center"
        android:padding="12dp"
        android:textColor="@color/black"
        android:textStyle="bold"/>

    <EditText
        android:id="@+id/etAngkatan"
        android:layout_width="0dp"
        android:layout_weight="2"
        android:layout_height="wrap_content"

    android:background="@drawable/input_border"
        android:padding="12dp"
        android:textColor="@color/black"
        android:text="2020"/>

```

```

        </LinearLayout>
    </LinearLayout>

    <!-- Tombol Save -->
    <Button
        android:id="@+id/btnSimpan"
        android:layout_width="120dp"
        android:layout_height="48dp"
        android:text="Save"
        android:textStyle="bold"
        android:layout_marginTop="24dp"
        android:background="@drawable/button_masuk"
        android:textColor="@android:color/black" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/bawah"
        android:layout_alignParentBottom="true"
        android:scaleType="fitEnd"/>
    </LinearLayout>
</ScrollView>

```

m. bookingactivity.kt

```

package com.example.bookingruang

import android.app.TimePickerDialog
import android.content.Intent
import android.os.Bundle
import android.util.Log // Import Log untuk debugging
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.firestore.FirebaseFirestore
import java.text.SimpleDateFormat
import java.util.*

class BookingActivity : AppCompatActivity() {

    private lateinit var startTimeEditText: EditText
    private lateinit var endTimeEditText: EditText
    private lateinit var firestore: FirebaseFirestore

    // Properti untuk menyimpan role pengguna yang login
    (penting!)
    private var currentUserRole: String = ""

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_booking)

        // --- PERBAIKAN UTAMA ---
        // Isi properti kelas 'this.currentUserRole' dari
        Intent
        this.currentUserRole =
        intent.getStringExtra("CURRENT_USER_ROLE") ?: ""
    }

```

```

        // Log ini akan menunjukkan nilai 'role' yang
        diterima dari MainActivity
        Log.d("BookingDebug", "Current User Role Received
(in onCreate): ${this.currentUserRole}")
        // --- AKHIR PERBAIKAN ---

        val lecturerName =
intent.getStringExtra("LECTURER_NAME") ?: "Tidak diketahui"
        val roomName = intent.getStringExtra("ROOM_NAME")

        firestore = FirebaseFirestore.getInstance()

        val roomNameTextView =
findViewById<TextView>(R.id.roomNameTextView)
        startTimeEditText =
findViewById(R.id.startTimeEditText)
        endTimeEditText =
findViewById(R.id.endTimeEditText)
        val courseEditText =
findViewById<EditText>(R.id.courseEditText)
        val submitButton =
findViewById<Button>(R.id.submitButton)

        roomNameTextView.text = roomName

        startTimeEditText.setOnClickListener {
            val calendar = Calendar.getInstance()
            TimePickerDialog(this, { _, hour, minute ->

startTimeEditText.setText(String.format("%02d:%02d", hour,
minute))
            }, calendar.get(Calendar.HOUR_OF_DAY),
calendar.get(Calendar.MINUTE), true).show()
        }

        endTimeEditText.setOnClickListener {
            val calendar = Calendar.getInstance()
            TimePickerDialog(this, { _, hour, minute ->

endTimeEditText.setText(String.format("%02d:%02d", hour,
minute))
            }, calendar.get(Calendar.HOUR_OF_DAY),
calendar.get(Calendar.MINUTE), true).show()
        }

        submitButton.setOnClickListener {
            val startTime =
startTimeEditText.text.toString().trim()
            val endTime =
endTimeEditText.text.toString().trim()
            val course =
courseEditText.text.toString().trim()

            if (startTime.isEmpty() || endTime.isEmpty() ||
course.isEmpty()) {
                Toast.makeText(this, "Harap isi semua
field!", Toast.LENGTH_LONG).show()
                return@setOnClickListener
            }

```

```

        if (roomName.isNullOrEmpty()) {
            Toast.makeText(this, "Nama ruangan tidak
ditemukan!", Toast.LENGTH_LONG).show()
            return@setOnClickListener
        }

        // Validasi roomNames
        val validRooms = listOf("Ruang A5", "Ruang A6",
"Ruang A7", "Ruang A8", "Ruang B1", "Ruang B2",
"Ruang B3", "Ruang B5", "Ruang B6", "Ruang
B7", "Ruang B8", "Ruang C1", "Ruang C2", "Ruang Aula 1",
"Ruang E2", "Ruang E3", "Ruang E4", "Ruang
F2", "Ruang F3", "Ruang Aula 2", "Lab Kom",
"Lab IoT", "Lab Micro", "Lab Bahasa", "Lab
Jar")

        if (roomName !in validRooms) {
            Toast.makeText(this, "Ruangan tidak
tersedia untuk booking!", Toast.LENGTH_LONG).show()
            return@setOnClickListener
        }

        try {
            val sdf = SimpleDateFormat("HH:mm",
Locale.getDefault())
            // Set zona waktu lokal untuk memastikan
konsistensi

            sdf.timeZone = TimeZone.getDefault()

            val startDate = sdf.parse(startTime)
            val endDate = sdf.parse(endTime)

            if (startDate != null && endDate != null &&
startDate.before(endDate)) {
                val remainingMinutes = (endDate.time -
startDate.time) / (1000 * 60)

                val todayDate = SimpleDateFormat("yyyy-
MM-dd", Locale.getDefault()).format(Date())

                firestore.collection("bookings")
                    .whereEqualTo("roomName", roomName)
                    .get()
                    .addOnSuccessListener {
querySnapshot ->
                        val bookingsToday =
querySnapshot.documents.filter { doc ->
                            // Memastikan kita hanya
memeriksa booking untuk hari ini
                                val timestamp =
doc.getLong("timestamp") ?: return@filter false
                                val bookingDate =
SimpleDateFormat("yyyy-MM-dd",
Locale.getDefault()).format(Date(timestamp))
                                    bookingDate == todayDate
                                }

                        val overlappingBooking =

```

```

bookingsToday.any { doc ->
    doc.getString("startTime")
    doc.getString("endTime")

    val existingStart =
    val existingEnd =

    try {
        val existingStartDate =
        val existingEndDate =
        existingStartDate !=
        existingEndDate != null &&

        startDate.before(existingEndDate) &&
        endDate.after(existingStartDate)

    } catch (e: Exception) {
        Log.e("BookingDebug",
            "Error parsing existing booking time", e)
        false
    }

    if (overlappingBooking) {
        Toast.makeText(this, "Waktu
bentrok dengan booking lain!", Toast.LENGTH_LONG).show()
        return@addOnSuccessListener

    }

    // Batas booking
    if (bookingsToday.size >= 25) {
        // Anda mungkin ingin memfilter hanya booking "aktif"
        Toast.makeText(this,
            "Booking penuh, maksimal 25 per hari!",
            Toast.LENGTH_LONG).show()

        return@addOnSuccessListener
    }

    // --- PEMBUATAN DATA BOOKING
    YANG DIKIRIM KE FIRESTORE ---
    val bookingData = hashMapOf(
        "roomName" to roomName,
        "startTime" to startTime,
        "endTime" to endTime,
        "subject" to course,
        "lecturerName" to
        lecturerName,
        "remainingMinutes" to
        remainingMinutes,
        "timestamp" to
        System.currentTimeMillis(),
        "status" to "aktif",
        "role" to
        this.currentUserRole // <-- Pastikan ini MERUJUK PROPERTI
        KELAS!
    )

    // --- LOG INI ADALAH KUNCI
    DIAGNOSA! ---
    // Ini akan mencetak seluruh

```

```

Map data yang akan Anda kirim ke Firestore.
// Periksa di Logcat nilai
'role' di sini.
Log.d("BookingDebug", "Data
yang akan dikirim ke Firestore: $bookingData")

firestore.collection("bookings")
    .add(bookingData)
    .addOnSuccessListener {
        Toast.makeText(this,
"Booking berhasil & tersimpan!", Toast.LENGTH_LONG).show()
        // Kembali ke
MainActivity dan bersihkan stack aktivitas
        val intent =
Intent(this, MainActivity::class.java)

intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP or
Intent.FLAG_ACTIVITY_NEW_TASK)

        startActivity(intent)
        finish()
    }
    .addOnFailureListener { e -
>
        Toast.makeText(this,
"Gagal simpan ke Firestore: ${e.message}",
Toast.LENGTH_LONG).show()
        Log.e("BookingDebug",
"Firestore error saat menyimpan booking", e)
    }
    .addOnFailureListener { e ->
        Toast.makeText(this, "Gagal
memeriksa booking: ${e.message}", Toast.LENGTH_LONG).show()
        Log.e("BookingDebug",
"Firestore error saat memeriksa booking", e)
    }

    } else {
        Toast.makeText(this, "Waktu mulai harus
sebelum waktu selesai!", Toast.LENGTH_LONG).show()
    }
    } catch (e: Exception) {
        Toast.makeText(this, "Format waktu salah!
Gunakan HH:mm", Toast.LENGTH_LONG).show()
        Log.e("BookingDebug", "Error parsing
waktu", e)
    }
    }
}
}

```

n. activity_booking.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

```

```

        android:orientation="vertical"
        android:background="#E4E4E4">

        <ImageView
            android:id="@+id/img1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/atas"
            android:scaleType="fitStart"/>

        <TextView
            android:id="@+id/roomNameTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Ruang A8"
            android:layout_gravity="center"
            android:textSize="20sp"
            android:textStyle="bold"
            android:padding="8dp"
            android:textColor="@color/black"/>

        <EditText
            android:id="@+id/startTimeEditText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="16dp"
            android:layout_marginEnd="16dp"
            android:clickable="true"
            android:cursorVisible="false"
            android:focusable="false"
            android:focusableInTouchMode="false"
            android:hint="Pilih waktu mulai (HH:mm)"
            android:inputType="none"
            android:minHeight="48dp"
            android:padding="8dp"
            android:textColor="@color/black"
            android:textColorHint="@color/black" />

        <EditText
            android:id="@+id/endTimeEditText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="16dp"
            android:layout_marginEnd="16dp"
            android:clickable="true"
            android:cursorVisible="false"
            android:focusable="false"
            android:focusableInTouchMode="false"
            android:hint="Pilih waktu selesai (HH:mm)"
            android:inputType="none"
            android:minHeight="48dp"
            android:padding="8dp"
            android:textColor="@color/black"
            android:textColorHint="@color/black" />

        <EditText
            android:id="@+id/courseEditText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

```

```

        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:hint="Mata Kuliah"
        android:minHeight="48dp"
        android:padding="8dp"
        android:textColor="@color/black"
        android:textColorHint="@color/black" />

<Button
    android:id="@+id/submitButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Submit"
    android:layout_gravity="center"
    android:padding="8dp"
    android:background="@drawable/button_masuk"/>

<ImageView
    android:layout_width="411dp"
    android:layout_height="372dp"
    android:layout_alignParentBottom="true"
    android:scaleType="fitEnd"
    android:src="@drawable/bawah" />
</LinearLayout>

```

o. roomstatusactivity.kt

```

package com.example.bookingruang

import android.os.Bundle
import android.os.CountDownTimer
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.database.FirebaseDatabase
import com.google.firebase.firestore.FirebaseFirestore
import java.text.SimpleDateFormat
import java.util.*
import android.util.Log

class RoomStatusActivity : AppCompatActivity() {

    private lateinit var roomName: String
    private var remainingTime: Int = 0

    private lateinit var tvRoomName: TextView
    private lateinit var tvLecturerName: TextView
    private lateinit var tvSubject: TextView
    private lateinit var tvStudentCount: TextView
    private lateinit var tvRemainingTime: TextView

    private val firestore = FirebaseFirestore.getInstance()
    private val realtimeDb =
        FirebaseDatabase.getInstance().reference

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }

```

```

        setContentView(R.layout.activity_room_status)

        // Ambil data dari intent
        roomName = intent.getStringExtra("ROOM_NAME") ?:
        "Unknown Room"
        remainingTime =
        intent.getIntExtra("REMAINING_TIME", 0)

        // Bind view
        tvRoomName = findViewById(R.id.namaruangan)
        tvLecturerName = findViewById(R.id.namadosen)
        tvSubject = findViewById(R.id.matakuliah)
        tvStudentCount = findViewById(R.id.jmlh_siswa)
        tvRemainingTime = findViewById(R.id.sisawaktu)

        // Tampilkan nama ruangan dan waktu tersisa
        tvRoomName.text = roomName
        tvRemainingTime.text = "Sisa Waktu: $remainingTime
        menit"

        // Load data Firestore dan Realtime DB
        loadBookingDetails()
        loadStudentCount()
    }

    private fun loadBookingDetails() {
        firestore.collection("bookings")
            .whereEqualTo("roomName", roomName)
            .whereEqualTo("status", "aktif")
            .get()
            .addOnSuccessListener { documents ->
                if (!documents.isEmpty) {
                    val booking = documents.first()
                    val lecturer =
                    booking.getString("lecturerName") ?: "-"
                    val subject =
                    booking.getString("subject") ?: "-"
                    val startTimeString =
                    booking.getString("startTime") ?: ""
                    val endTimeString =
                    booking.getString("endTime") ?: ""

                    tvLecturerName.text = "Dosen:
                    $lecturer"
                    tvSubject.text = "Mata Kuliah:
                    $subject"

                    startCountdownTimer(startTimeString,
                    endTimeString)
                } else {
                    tvLecturerName.text = "Dosen: Tidak
                    ditemukan"
                    tvSubject.text = "Mata Kuliah: Tidak
                    ditemukan"
                    tvRemainingTime.text = "Sisa Waktu:
                    Tidak ada booking aktif"
                }
            }
        .addOnFailureListener {

```

```

        dosen"
        tvLecturerName.text = "Gagal memuat data"
        tvSubject.text = "Gagal memuat mata kuliah"
        tvRemainingTime.text = "Sisa Waktu: Error"
    }
}

private fun startCountdownTimer(startTimeStr: String,
endTimeStr: String) {
    val sdf = SimpleDateFormat("HH:mm",
Locale.getDefault())
    sdf.timeZone = TimeZone.getDefault() // gunakan
zona waktu lokal

    try {
        val now = Calendar.getInstance()

        // Set waktu mulai
        val startParts = startTimeStr.split(":").map {
it.toInt() }
        val startTime = Calendar.getInstance().apply {
            set(Calendar.HOUR_OF_DAY, startParts[0])
            set(Calendar.MINUTE, startParts[1])
            set(Calendar.SECOND, 0)
            set(Calendar.MILLISECOND, 0)
        }

        // Set waktu selesai
        val endParts = endTimeStr.split(":").map {
it.toInt() }
        val endTime = Calendar.getInstance().apply {
            set(Calendar.HOUR_OF_DAY, endParts[0])
            set(Calendar.MINUTE, endParts[1])
            set(Calendar.SECOND, 0)
            set(Calendar.MILLISECOND, 0)
        }

        // Debugging (opsional): log waktu
        Log.d("WAKTU", "Sekarang: ${now.time}, Mulai:
${startTime.time}, Selesai: ${endTime.time}")

        when {
            now.before(startTime) -> {
                tvRemainingTime.text = "Belum waktunya
mulai"
            }
            now.after(endTime) -> {
                tvRemainingTime.text = "Sisa waktu
habis"
            }
            else -> {
                val diffInMillis = endTime.timeInMillis
- now.timeInMillis
                object : CountDownTimer(diffInMillis,
1000) {
                    override fun
onTick(millisUntilFinished: Long) {
                        val minutes =
(millisUntilFinished / 1000) / 60

```

```

                                val seconds =
(millisUntilFinished / 1000) % 60
                                tvRemainingTime.text = "Sisa
waktu: $minutes menit $seconds detik"
                                }

                                override fun onFinish() {
                                    tvRemainingTime.text = "Sisa
waktu habis"
                                }
                                }.start()
                            }
                        }
                    } catch (e: Exception) {
                        tvRemainingTime.text = "Format waktu tidak
valid"
                        e.printStackTrace()
                    }
                }

                private fun loadStudentCount() {
                    realtimeDb.child("room_sensors").child(roomName).child("act
ive_students")
                        .get()
                        .addOnSuccessListener { snapshot ->
                            val studentCount =
snapshot.getValue(Int::class.java) ?: 0
                            tvStudentCount.text = "Jumlah Mahasiswa:
$studentCount"
                        }
                        .addOnFailureListener {
                            tvStudentCount.text = "Gagal memuat jumlah
mahasiswa"
                        }
                }
            }
        }
    }
}

```

p. activity_room_status.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:background="#E4E4E4"
    android:gravity="center">

    <ImageView
        android:id="@+id/img1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/atas"
        android:scaleType="fitStart"/>

    <TextView

```

```

        android:id="@+id/namaruangan"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ruang A8"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:padding="8dp" />

<TextView
    android:id="@+id/statusruang"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ruang sedang dipakai"
    android:textSize="16sp"
    android:textColor="@color/black"
    android:padding="8dp" />

<TextView
    android:id="@+id/namadosen"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Nama Dosen: Safiq Rosad"
    android:textSize="16sp"
    android:textColor="@color/black"
    android:padding="8dp" />

<TextView
    android:id="@+id/matakuliah"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Mata Kuliah: Sistem Operasi"
    android:textSize="16sp"
    android:textColor="@color/black"
    android:padding="8dp" />

<TextView
    android:id="@+id/jmlh_siswa"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Jumlah Siswa: 30 orang"
    android:textSize="16sp"
    android:textColor="@color/black"
    android:padding="8dp" />

<TextView
    android:id="@+id/sisawaktu"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Sisa waktu: 30 menit"
    android:textSize="16sp"
    android:textColor="@color/black"
    android:padding="8dp" />

<Button
    android:id="@+id/btn_kembali"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Kembali"

```

```

        android:background="@drawable/button_masuk"/>

        <ImageView
            android:layout_width="411dp"
            android:layout_height="372dp"
            android:layout_alignParentBottom="true"
            android:scaleType="fitEnd"
            android:src="@drawable/bawah" />
    </LinearLayout>

```

q. androidmanifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission
        android:name="android.permission.READ_MEDIA_IMAGES" />

    <application
        android:allowBackup="true"

        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Bookingruang"
        tools:targetApi="31">

        <!-- SplashScreen sebagai LAUNCHER -->
        <activity
            android:name=".SplashScreen"
            android:exported="true"
            android:screenOrientation="portrait"

            android:label="@string/title_activity_splash_screen"
            android:theme="@style/Theme.Bookingruang"

            tools:ignore="DiscouragedApi,LockedOrientationActivity">

            <intent-filter>
                <action
                    android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity
            android:name=".MasuksebagaiActivity"
            android:exported="false"

            android:label="@string/title_activity_masuk_sebagai"
            android:theme="@style/Theme.Bookingruang" />

        <activity

```

```

        android:name=".LoginActivity"
        android:exported="false"
        android:label="@string/title_activity_login"
        android:theme="@style/Theme.Bookingruang" />

<activity
    android:name=".RoomStatusActivity"
    android:exported="false"
    android:label="@string/title_activity_room_status"
    android:theme="@style/Theme.Bookingruang" />

<activity
    android:name=".BookingActivity"
    android:exported="false"
    android:label="@string/title_activity_booking"
    android:theme="@style/Theme.Bookingruang" />

<activity
    android:name=".MainActivity"
    android:exported="true" />

<activity
    android:name=".SettingActivity"
    android:exported="true"
    android:label="@string/title_activity_setting"/>

<activity
    android:name=".EditProfilActivity"
    android:exported="true"
    android:label="@string/title_activity_edit_profil"/>

</application>

</manifest>

```

r. colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFF</color>
    <color name="blue_empty">#00aeb4</color>
    <color name="red_used">#F44336</color>
    <color name="teal_700">#018786</color>
</resources>
```

s. string.xml

```
<resources>
    <string name="app_name">bookingruang</string>
    <string
name="title_activity_booking">BookingActivity</string>
    <string
name="title_activity_room_status">RoomStatusActivity</string>
    <string
name="title_activity_splash_screen">ClassRoom</string>
    <string name="hint_username">Username</string>
    <string name="hint_password">Password</string>
    <string
name="title_activity_login">LoginActivity</string>
    <string
name="title_activity_masuk_sebagai">MasuksebagaiActivity</string>
    <string
name="title_activity_setting">SettingActivity</string>
    <string
name="title_activity_edit_profil">EditProfilActivity</string>
</resources>
```

2. Pengujian Black Box Testing

a. Pengujian pada menu *splashscreen*

Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
Tampilan splash screen	Klik icon aplikasi	Menampilkan logo aplikasi sebagai pembuka aplikasi mulai berjalan	[√] berhasil [] tidak berhasil

b. Pengujian pada *login*

1) Pengujian masuk sebagai

Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
Akses masuk sebagai dosen	Klik <i>button</i> 'dosen'	Menampilkan <i>form username</i> dan <i>password</i>	[√] berhasil [] tidak berhasil
Akses masuk sebagai mahasiswa	Klik <i>button</i> 'mahasiswa'	Menampilkan <i>form username</i> dan <i>password</i>	[√] berhasil [] tidak berhasil
Akses masuk sebagai admin	Klik <i>button</i> 'admin'	Menampilkan <i>form username</i> dan <i>password</i>	[√] berhasil [] tidak berhasil

2) Pengujian pada menu *login*

Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
<i>Inputan username</i>	Klik <i>form username</i>	Menampilkan ketikan <i>username</i>	[√] berhasil [] tidak berhasil
<i>Inputan password</i>	Klik <i>form password</i>	Menampilkan ketikan <i>password</i>	[√] berhasil [] tidak berhasil
Akses <i>button</i> masuk	Klik <i>button</i> 'masuk'	Login berhasil dan mengakses dashboard	[√] berhasil [] tidak berhasil

c. Pengujian pada *Setting*

Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
Tampilan <i>setting</i>	Klik <i>icon setting</i> pada <i>dashboard</i>	Menampilkan pilihan Edit profil dan logput	[☒] berhasil [☐] tidak berhasil
Akses menuju edit profil	Klik edit profil	Menampilkan form nama, NIM, Fak/Prodi, Angkatan, dan save	[☒] berhasil [☐] tidak berhasil
akses <i>button logout</i>	Klik <i>logout</i>	Berhasil <i>logout</i> dan menampilkan <i>login</i>	[☒] berhasil [☐] tidak berhasil

d. Pengujian pada Edit Profil

Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
<i>Icon</i> profil	Klik <i>icon profil</i>	Menuju galeri dan mengganti foto profil	[☒] berhasil [☐] tidak berhasil
<i>Inputan</i> Nama	Klik <i>form</i> nama	Menampilkan ketikan nama	[☒] berhasil [☐] tidak berhasil
<i>Inputan</i> NIM	Klik <i>form</i> NIM	Menampilkan ketikan NIM	[☒] berhasil [☐] tidak berhasil
<i>Inputan</i> Fak/Prodi	Klik <i>form</i> Fak/Prodi	Menampilkan ketikan Fak/Prodi	[☒] berhasil [☐] tidak berhasil
<i>Inputan</i> Angkatan	Klik <i>form</i> Angkatan	Menampilkan ketikan angkatan	[☒] berhasil [☐] tidak berhasil
<i>Button</i> Save	Klik <i>button</i> save	Berhasil mensave inputan	[☒] berhasil [☐] tidak berhasil

e. Pengujian pada *Dashboard*

1) Dosen dan admin

Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
Akses tampilan <i>dashboard</i>	Secara <i>realtime</i> berubah warna jika ruangan di <i>booking</i>	Ruangan berubah warna. Warna merah menandakan ruangan kelas sedang terpakai, dan warna biru menandakan ruang kelas tidak digunakan	[☒] berhasil [☐] tidak berhasil
Akses <i>booking</i> ruang kelas	Klik <i>button</i> nama ruang kelas yang berwarna biru	Menampilkan <i>form booking</i> ruangan	[☒] berhasil [☐] tidak berhasil
Akses status ruang kelas	Klik <i>button</i> nama ruang kelas yang berwarna merah	Menampilkan status ruang kelas	[☒] berhasil [☐] tidak berhasil

2) Mahasiswa

Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
Akses tampilan <i>dashboard</i>	Secara <i>realtime</i> berubah warna jika ruangan di <i>booking</i>	Ruangan berubah warna. Warna merah menandakan ruangan kelas sedang digunakan, dan warna biru menandakan ruang kelas tidak digunakan.	[☒] berhasil [☐] tidak berhasil

Akses status ruang kelas	Klik <i>button</i> nama ruang kelas yang berwarna merah	Menampilkan status ruang kelas	[☒] berhasil [☐] tidak berhasil
--------------------------	---	--------------------------------	---

f. Pengujian *Booking Class*

Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
<i>Input</i> waktu memulai perkuliahan	Klik waktu mulai	Menampilkan <i>time pickers</i>	[☒] berhasil [☐] tidak berhasil
<i>Input</i> waktu perkuliahan berakhir	Klik waktu mulai selesai	Menampilkan <i>time pickers</i>	[☒] berhasil [☐] tidak berhasil
<i>Input</i> nama mata kuliah	Klik <i>form</i> mata kuliah	Mengisi <i>input</i> mata kuliah	[☒] berhasil [☐] tidak berhasil
<i>Submit booking</i> ruangan	Klik <i>submit</i>	Ruang kelas <i>terbooking</i>	[☒] berhasil [☐] tidak berhasil

g. Pengujian *Roomstatus*

Skenario Pengujian	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
Akses status ruang kelas	Klik <i>button</i> ruang kelas yang berwarna merah	Menampilkan status ruang kelas “ruang sedang dipakai” dengan nama dosen, mata kuliah, jumlah siswa, dan sisa waktu perkuliahan	[☒] berhasil [☐] tidak berhasil