“ UJIAN AKHIR

PEMROGRAMAN PERANGKAT BERGERAK”

Disusun oleh:

Nama :

1. Natalia Vanessa Sailo _225314108
2. Mulyanita Susanti Efendi

PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS SANATA DHARMA

YOGYAKARTA

2025

A. TUJUAN

Membuat aplikasi mobile Android yang dapat digunakan untuk mengatur event atau acara**.**

B. DESAIN

a. Use Case yang Didisain

Pada aplikasi Event Manager, terdapat tiga proses utama yang menjadi basis perancangan sistem, yaitu:

Create Event

➔ Digunakan untuk menambah event baru ke dalam sistem.

Get Event by ID

➔ Digunakan untuk mencari event berdasarkan ID tertentu.

Get Event by Date

➔ Digunakan untuk menampilkan daftar event berdasarkan tanggal tertentu.

Delete

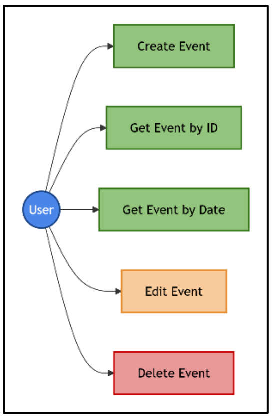
➔ Pengguna dapat menghapus event. Sebelum dihapus, program menampilkan AlertDialog konfirmasi**.**

Edit Event

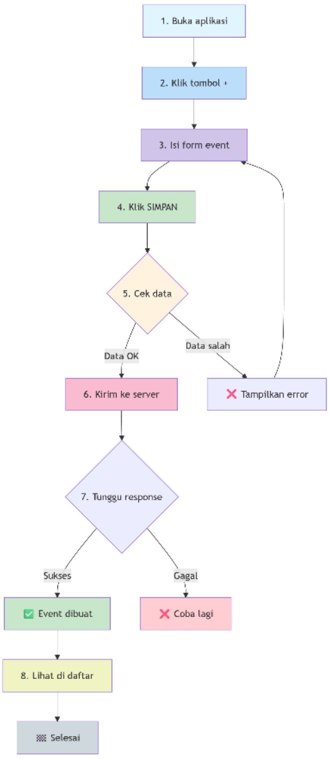
➔ Pengguna dapat memperbarui data event yang sudah ada

b. Diagram Use Case, diagram aktivitas,diagram kelas

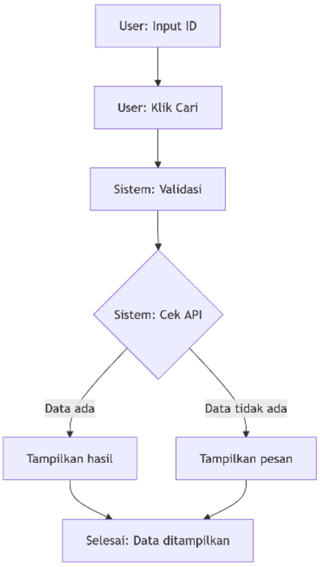
- Diagram use case



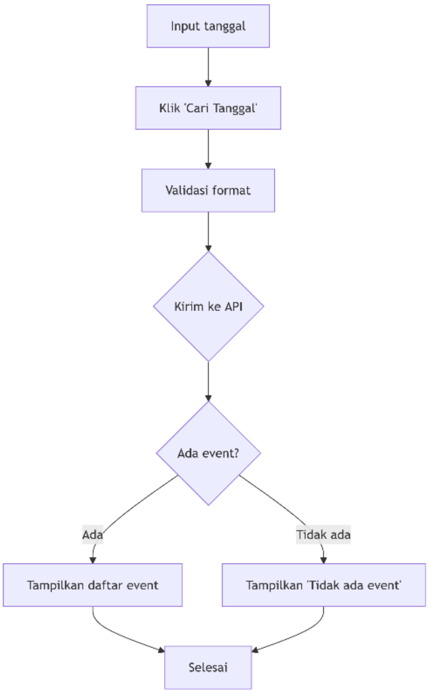
- Diagram aktivitas
  - Create event



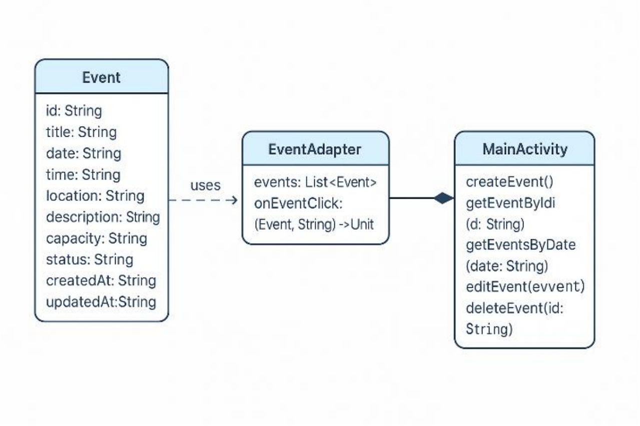
  - Getevent by id



- Getevent by date



- Diagram kelas



c. Implementasi

File UI->kelas Main

- Data Class – Event

```
data class Event(
    val id: String,
    val title: String,
    val date: String,
    val time: String,
    val location: String,
    val description: String,
    val capacity: String,
    val status: String,
    val createdAt: String,
    val updatedAt: String
)
```

- kelas -> EventAdapter

```
class EventAdapter(
    private val items: List<Event>,
    private val action: (Event, String) -> Unit
) : RecyclerView.Adapter<EventAdapter.Holder>() {
override fun getItemCount(): Int = items.size

    override fun onBindViewHolder(h: Holder, pos: Int) {
        val e = items[pos]

        h.tvId.text = "ID: ${e.id}"
        h.tvTitle.text = e.title
        h.tvDate.text = "${e.date} | ${e.time}"
        h.tvLocation.text = e.location
        h.tvStatus.text = e.status

        h.itemView.setOnClickListener { action(e, "view") }
        h.btnEdit.setOnClickListener { action(e, "edit") }
        h.btnDelete.setOnClickListener { action(e, "delete") }
    }
}
```

- Kelas mainAktivity Load Semua Event

```
// ================== LOAD ALL EVENTS
===================
private fun loadEvents() {
    show(true)
    CoroutineScope(Dispatchers.IO).launch {
        try {
            val url = if (filter == "all") API else "$API?status=$filter"
            val json = JSONObject(get(url))

            if (json.getInt("status") == 200) {
                val arr = json.getJSONArray("data")
                val data = parseList(arr)

                withContext(Dispatchers.Main) {
                    list.clear()
                    list.addAll(data)
                    adapter.notifyDataSetChanged()
                    updateEmpty()
                }
            }
        } finally {
            withContext(Dispatchers.Main) { show(false) }
        }
    }
}
```

- Get Event by ID

```
// =================== GET EVENT BY ID
===================
private fun getEventById(id: String) {
    show(true)
    CoroutineScope(Dispatchers.IO).launch {
        try {
            val json = JSONObject(get("$API?id=$id"))
            val event: Event? =
                if (json.getInt("status") == 200) {
                    val data = json.get("data")
                    when (data) {
                        is JSONObject -> parseObj(data)
                        is JSONArray -> parseList(data).firstOrNull()
                        else -> null
                    }
                } else null

            withContext(Dispatchers.Main) {
                list.clear()
                if (event != null) list.add(event)
                adapter.notifyDataSetChanged()
                updateEmpty()
            }

        } finally {
            withContext(Dispatchers.Main) { show(false) }
        }
    }
}
```

- Get Event by Date

```
// =================== GET EVENTS BY DATE ===================
private fun getByDate(date: String) {
    show(true)
    CoroutineScope(Dispatchers.IO).launch {
        try {
            val json = JSONObject(get("$API?date=$date"))
            val events = if (json.getInt("status") == 200)
                parseList(json.getJSONArray("data"))
            else emptyList()

            withContext(Dispatchers.Main) {
                list.clear()
                list.addAll(events)
                adapter.notifyDataSetChanged()
                updateEmpty()
            }

        } finally {
            withContext(Dispatchers.Main) { show(false) }
        }
    }
}
```

- Create Event

```
// =================== CREATE EVENT ===================
private fun showCreateDialog() {
    val v = LayoutInflater.from(this).inflate(R.layout.dialog_event_form, null)

    val dialog = AlertDialog.Builder(this)
        .setTitle("Buat Event Baru")
        .setView(v)
        .setPositiveButton("Simpan", null)
        .setNegativeButton("Batal", null)
        .create()

    dialog.setOnShowListener {
        dialog.getButton(AlertDialog.BUTTON_POSITIVE).setOnClickListener {
            val t = v.findViewById<EditText>(R.id.etTitle).text.toString()
            val d = v.findViewById<EditText>(R.id.etDate).text.toString()
            val tm = v.findViewById<EditText>(R.id.etTime).text.toString()
            val loc = v.findViewById<EditText>(R.id.etLocation).text.toString()
            val desc = v.findViewById<EditText>(R.id.etDescription).text.toString()
            val cap = v.findViewById<EditText>(R.id.etCapacity).text.toString()
            val st = v.findViewById<Spinner>(R.id.spinnerStatus).selectedItem.toString()

            createEvent(t, d, tm, loc, desc, cap, st)
            dialog.dismiss()
        }
    }

    val spinner = v.findViewById<Spinner>(R.id.spinnerStatus)
    spinner.adapter = ArrayAdapter(
        this,
        android.R.layout.simple_spinner_dropdown_item,
        listOf("upcoming", "ongoing", "completed", "cancelled")
    )

    dialog.show()
}

private fun createEvent(t: String, d: String, tm: String, loc: String, desc: String, cap: String, st: String) {
    CoroutineScope(Dispatchers.IO).launch {
        try {
            val json = JSONObject().apply {
                put("title", t)
                put("date", d)
                put("time", tm)
                put("location", loc)
                put("description", desc)
                put("capacity", cap.toIntOrNull() ?: 0)
                put("status", st)
            }

            requestBody(API, "POST", json.toString())
            withContext(Dispatchers.Main) { loadEvents() }
        } catch (e: Exception) {
            withContext(Dispatchers.Main) { toast(e.message!!) }
        }
    }
}
```

- Delete Event

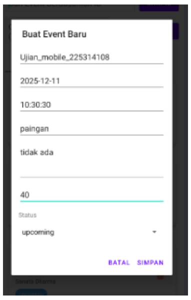
```
// =================== DELETE ===================
private fun confirmDelete(e: Event) {
    AlertDialog.Builder(this)
        .setTitle("Hapus Event")
        .setMessage("Yakin ingin menghapus '${e.title}'?")
        .setPositiveButton("Hapus") { _, _ -> deleteEvent(e.id) }
        .setNegativeButton("Batal", null)
        .show()
}

private fun deleteEvent(id: String) {
    CoroutineScope(Dispatchers.IO).launch {
        try {
            requestBody("$API?id=$id", "DELETE", "")
            withContext(Dispatchers.Main) { loadEvents() }
        } catch (e: Exception) {
            withContext(Dispatchers.Main) { toast(e.message!!) }
        }
    }
}
```

- Edit event

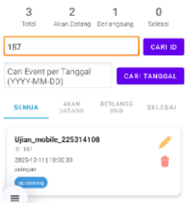
```
// =================== EDIT EVENT ===================
private fun showEditDialog(e: Event) {
    val v = LayoutInflater.from(this).inflate(R.layout.dialog_event_form, null)

    v.findViewById<EditText>(R.id.etTitle).setText(e.title)
    v.findViewById<EditText>(R.id.etDate).setText(e.date)
    v.findViewById<EditText>(R.id.etTime).setText(e.time)
    v.findViewById<EditText>(R.id.etLocation).setText(e.location)
    v.findViewById<EditText>(R.id.etDescription).setText(e.description)
    v.findViewById<EditText>(R.id.etCapacity).setText(e.capacity)

    val statuses = listOf("upcoming", "ongoing", "completed", "cancelled")
    val spinner = v.findViewById<Spinner>(R.id.spinnerStatus)
    spinner.adapter = ArrayAdapter(this, android.R.layout.simple_spinner_dropdown_item, statuses)
    spinner.setSelection(statuses.indexOf(e.status))

    val dialog = AlertDialog.Builder(this)
        .setTitle("Edit Event")
        .setView(v)
        .setPositiveButton("Update", null)
        .setNegativeButton("Batal", null)
        .create()

    dialog.setOnShowListener {
        dialog.getButton(AlertDialog.BUTTON_POSITIVE).setOnClickListener {
            updateEvent(
                e.id,
                v.findViewById<EditText>(R.id.etTitle).text.toString(),
                v.findViewById<EditText>(R.id.etDate).text.toString(),
                v.findViewById<EditText>(R.id.etTime).text.toString(),
                v.findViewById<EditText>(R.id.etLocation).text.toString(),
                v.findViewById<EditText>(R.id.etDescription).text.toString(),
                v.findViewById<EditText>(R.id.etCapacity).text.toString(),
                spinner.selectedItem.toString()
            )
            dialog.dismiss()
        }
    }

    dialog.show()
}

private fun updateEvent(id: String, t: String, d: String, tm: String, loc: String, desc: String,
cap: String, st: String) {
    CoroutineScope(Dispatchers.IO).launch {
        try {
            val json = JSONObject().apply {
                put("title", t)
                put("date", d)
                put("time", tm)
                put("location", loc)
                put("description", desc)
                put("capacity", cap.toIntOrNull() ?: 0)
                put("status", st)
            }

            requestBody("$API?id=$id", "PUT", json.toString())
            withContext(Dispatchers.Main) { loadEvents() }
        } catch (e: Exception) {
            withContext(Dispatchers.Main) { toast(e.message!!) }
        }
    }
}
```

C. PENGUJIAN PROGRAM

- Case : create event

Tgl Uji	Data masukan	Hasil yang diharapkan	Hasil yang didapat	Kesimpulan
11-12-2025	"Ujian_mobile _NIM",tgl:11-12-2025, kampus_paingan, 40 org  Buat Event Baru Ujian_mobile_225314108 2025-12-11 10:30:30 paingan tidak ada 40 Status upcoming 	Data disimpan di tabel	Data masuk database	terima/~~tolak~~

- Case : GetEvet by id

Tgl Uji	Data masukan	Hasil yang diharapkan	Hasil yang didapat	Kesimpulan
11-12-2025	187	Menampilkan event ID 187	Data tampil sesuai  3 Total 2 Akan Datang 1 Berlangsung 0 Selesai 187 Cari Event per Tanggal (YYYY-MM-DD) CARI ID CARI TANGGAL SEMUA AKAN DATANG BERLANGSUNG SELESAI Ujian_mobile_225314108 ID: 187 2025-12-11 | 10:30:30 paingan upcoming	terima

- Case ; GetEvent by date

Tgl Uji	Data masukan	Hasil yang diharapkan	Hasil yang didapat	Kesimpulan
11-12-2025	2025-12-11	Menampilkan event tanggal tersebut	Data tampil sesuai	terima

- Case: edit

Tgl Uji	Data masukan	Hasil yang diharapkan	Hasil yang didapat	Kesimpulan
11-12-2025	Ubah judul event	Data berhasil berubah	Data ter-update	terima

- Case:delete

Tgl Uji	Data masukan	Hasil yang diharapkan	Hasil yang didapat	Kesimpulan
11-12-2025	Hapus ID 187	Event hilang dari list	Berhasil terhapus	terima

D. KESIMPULAN

Dari hasil perancangan, implementasi, dan pengujian aplikasi Event Management, diperoleh kesimpulan sebagai berikut:

1. Aplikasi berhasil dibuat menggunakan Android Studio dengan bahasa pemrograman Kotlin.
2. Aplikasi mampu melakukan operasi dasar CRUD terhadap data event.
3. Aplikasi dapat melakukan pencarian event berdasarkan:
    - ID event

- Tanggal event

4. Program dapat melakukan delete dan edit

E. Lampiran

Daftar pekerjaan setiap anggota dengan link repository (komentar commitnya)

Nama Mahasiswa: Natalia Vanessa sailo_ 225314108

Tanggal	Pekerjaan Yang dilakukan	Link github
9 Desember 2025	First comit	44e21c2d2555caceb156aaa67cac4910e1b51013
9 Desember 2025	Menambahkan semua file project	343c533c2edd6a8fe909ef000a7d641ed7a64444
9 Desember 2025	Menambah vcs.xml	686a71836adfea71ad3954cf11fdd50cab6bbf70
9 Desember 2025	Memperbaiki kelas activity main.xml	
9 Desember 2025	Memperbaiki kelas Mainaktivity.kt	
11 Desember 2025	Melengkapi main	
11 Desember 2025	Merge remote branchf final	
11 Desember 2025	Melengkapi activity main.xml	
11 Desember 2025	Menambah program eventadapter.kt	

Nama Mahasiswa: Mulyanita Susanti Efendi_225314047

Tanggal	Pekerjaan Yang dilakukan	Link github
9 Desember 2025	Memperbaiki kelas dialog	be38929c652271f366af3668d

	event	3be09883cdde5d2
9 Desember 2025	Memperbaiki kelas item_event.xml	de96e05e6edfe03cb44e34c71621c1aff9d29444
9 Desember 2025	Commit oleh santi	2860a43f7c6667ecebfaef125c2aef38864ce9e1
11 Desember 2025	Memperbaiki kelals item event.xml	
11 Desember 2025	Memperbaiki dialog event	

F. Link youtube:
https://youtu.be/0lY-2VC58CM