

Nama : Salwa Tsabitah  
NIM : 1313621033  
Prodi : Ilmu Komputer 2021  
UAS : Mobile Computing

---

## Aplikasi Notes

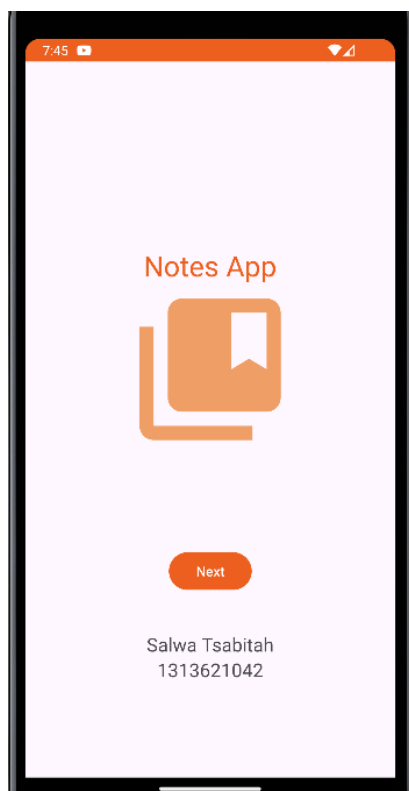
### A. Latar Belakang

Saya memilih membuat aplikasi notes karena kebiasaan saya yang setiap hari selalu membuka aplikasi notes untuk menulis hal-hal dan menjadi todo list yang akan saya lakukan. Aplikasi notes yang sering saya pakai yaitu Google Keep Notes dan Notion. Oleh karena itu, saya berpikir untuk membuat aplikasi notes ini. Selain itu, aplikasi ini termasuk aplikasi yang mudah untuk diimplementasikan menggunakan android studio dan juga terdapat penggunaan database Create, Update, dan Delete.

### B. Penjelasan Fitur

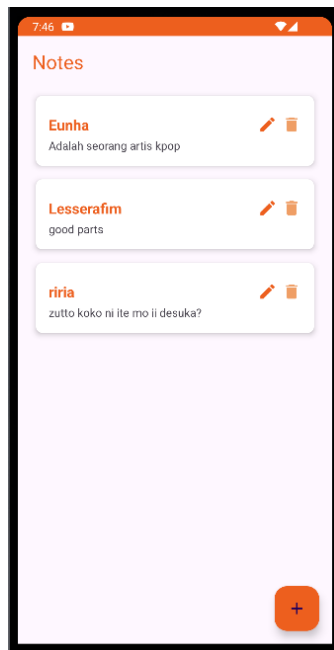
Berikut adalah tampilan dan penjelasan dari fitur-fitur aplikasi notes sederhana

#### a. Halaman pertama untuk nama mahasiswa dan NIM



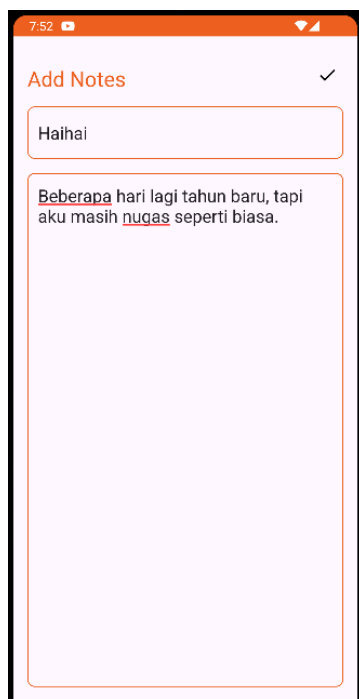
Pengguna dapat menekan button next untuk menampilkan halaman Notes.

**b. Halaman utama yang terdapat catatan yang tersimpan**



Pada halaman ini setiap catatan terdapat button untuk mengedit dan menghapus catatan. Dan jika ingin menambah catatan baru bisa menekan tombol dibawah kanan.

**c. Halaman tambah catatan dan edit catatan.**



Pada halaman ini pengguna dapat menulis judul catatan dan menulis isi kontennya, kemudian meng klik symbol ceklis pada atas kanan untuk menyimpan. Apabila catatan sudah tersimpan akan terlihat dan terdapa toast.

#### d. Implementasi database pada Kotlin

Berikut adalah fungsi-fungsi untuk membuat CRUD.

```
class NoteDatabaseHelper (context: Context): SQLiteOpenHelper(context, DATABASE_NAME, null, DATABASE_VERSION) {

    companion object{
        private const val DATABASE_NAME = "notesapp.com"
        private const val DATABASE_VERSION = 1
        private const val TABLE_NAME = "allnotes"
        private const val COLUMN_ID = "Id"
        private const val COLUMN_CONTENT = "content"
        private const val COLUMN_TITLE = "title"
    }

    override fun onCreate(db: SQLiteDatabase?){
        val createTableQuery = "CREATE TABLE $TABLE_NAME ($COLUMN_ID INTEGER PRIMARY KEY, $COLUMN_TITLE TEXT, $COLUMN_CONTENT TEXT)"
        db?.execSQL(createTableQuery)
    }

    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int){
        val dropTableQuery = "DROP TABLE IF EXISTS $TABLE_NAME"
        db?.execSQL(dropTableQuery)
        onCreate(db)
    }

    fun insertNote(note: Note){
        val db = writableDatabase
        val values = ContentValues().apply { this.put(COLUMN_TITLE, note.title) }
        db.insert(TABLE_NAME, null, values)
        db.close()
    }
}
```

```
fun getAllNotes(): List<Note>{
    val notesList = mutableListOf<Note>()
    val db = readableDatabase
    val query = "SELECT * FROM $TABLE_NAME"
    val cursor = db.rawQuery(query, null)

    while (cursor.moveToNext()){
        val id = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_ID))
        val title = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_TITLE))
        val content = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_CONTENT))

        val note = Note(id, title, content)
        notesList.add(note)
    }
    cursor.close()
    db.close()
    return notesList
}

fun updateNote(note: Note){
    val db = writableDatabase
    val values = ContentValues().apply { this.put(COLUMN_TITLE, note.title) }
    val whereClause = "$COLUMN_ID = ?"
    val whereArgs = arrayOf(note.id.toString())
    db.update(TABLE_NAME, values, whereClause, whereArgs)
    db.close()
}
```

```
fun getNotesByID(noteId: Int): Note{
    val db = readableDatabase
    val query = "SELECT * FROM $TABLE_NAME WHERE $COLUMN_ID = $noteId"
    val cursor = db.rawQuery(query, null)
    cursor.moveToFirst()

    val id = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_ID))
    val title = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_TITLE))
    val content = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_CONTENT))

    cursor.close()
    db.close()
    return Note(id, title, content)
}

fun deleteNote(noteId: Int){
    val db = writableDatabase
    val whereClause = "$COLUMN_ID = ?"
    val whereArgs = arrayOf(noteId.toString())
    db.delete(TABLE_NAME, whereClause, whereArgs)
    db.close()
}
```