

3. ANALISIS DAN DESAIN SISTEM

3.1 Analisis Permasalahan

Self Management adalah kemampuan seseorang untuk mengendalikan diri terhadap suatu tindakan. Namun masih banyak orang yang meremehkan hal ini khususnya dalam hal mengatur jadwal kegiatan. Suatu kegiatan pada umumnya dilakukan untuk mencapai sesuatu. Misalnya bekerja untuk membeli sesuatu, atau belajar untuk mendapat nilai bagus dalam ujian. Untuk mencapai tujuan tersebut, orang harus membuat skala prioritas dalam melakukan kegiatan. Namun Orang cenderung hanya mengandalkan ingatan dan asal mengatur hal-hal penting tersebut. Ada juga orang yang mencatatnya di *sticky note* dan ditempelkan di meja kerja atau belajar. Namun orang harus selalu kembali ke meja untuk melihat *sticky note* tersebut.

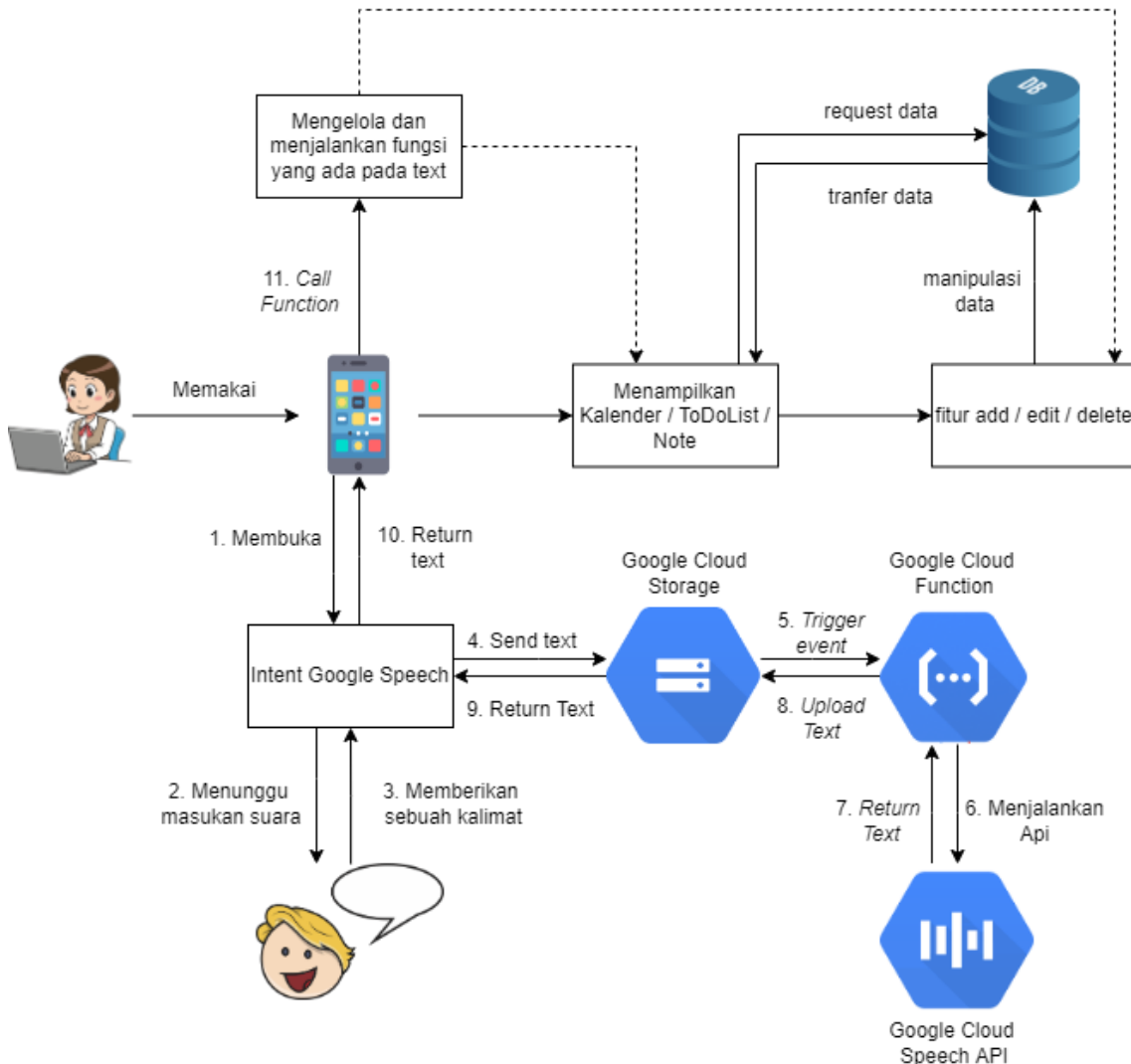
Adapun orang yang sudah melek teknologi dan mengunduh aplikasi untuk mencatat jadwal atau skala prioritas. Namun karena kurangnya fitur dalam satu aplikasi, orang harus mengunduh aplikasi lebih dari satu. Aplikasi yang beredar di pasaran cenderung hanya fokus ke satu hal saja. Contohnya Google Calendar yang hanya menyediakan fitur mencatat jadwal, dan To Do List Apps yang hanya menyediakan fitur mencatat daftar kegiatan. Pada dasarnya, fitur dari dua aplikasi tersebut dibutuhkan untuk *self-management*. Namun karena fitur tersebut tidak ada dalam satu aplikasi, orang harus mengunduh dua aplikasi yang berbeda.

Untuk menjawab permasalahan diatas, dibutuhkannya aplikasi yang memiliki fitur berbeda dalam satu aplikasi. Fitur yang berbeda tersebut bukan berarti fungsinya berbeda. Orang juga tentu membutuhkan aplikasi untuk mencatat yang cepat digunakan. Waktu adalah suatu hal yang sangat penting untuk orang yang memiliki kemampuan *self-management* yang baik. Karena itu, diperlukan fitur *speech to text* yang dikenal mempercepat mencatat dibandingkan pengetikan manual.

3.2 Blok Diagram Aplikasi

Blok Diagram adalah salah satu diagram yang menggambarkan proses berjalannya sistem dalam suatu aplikasi. Proses sistem aplikasi yang penulis buat dapat dilihat pada Gambar 3.1. Diagram diawali dengan seorang pengguna menggunakan aplikasi. Aplikasi menampilkan kalender dan dapat langsung pindah ke ToDoList atau Note. Data yang ditampilkan didapatkan dengan meminta data pada Database dan dikembalikan dalam bentuk data *object*. Pengguna dapat mengubah atau menghapus data tersebut atau menambahkan data baru. Setelah

perintah tambah atau ubah dijalankan, data otomatis masuk kedalam database. Jika perintah hapus dijalankan, data yang berada di Database otomatis terhapus.



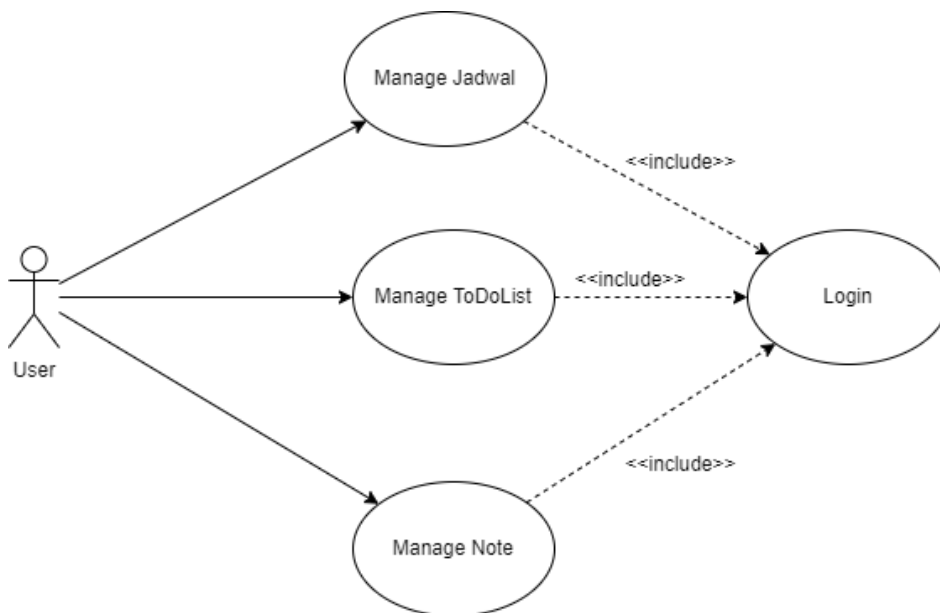
Gambar 3.1 Application Block Diagram

Dalam proses fitur Google Speech API, aplikasi memanggil sebuah *intent*, *Intent* memiliki peran sebagai pembawa informasi dari suatu aktivitas dengan aktivitas lainnya. Dalam aktivitas ini, penggunaan *intent* untuk memanggil halaman Google Speech. Pengguna kemudian dapat mengucapkan sebuah perintah berbentuk kalimat yang tersedia oleh aplikasi. Setelah kalimat selesai diucapkan, *Intent* kemudian men-trigger Google Cloud Storage (GCS). GCS kemudian memanggil Google Cloud Function (GCF) dan coba mengenali *object* yang masuk. Setelah GCF mengenali *object* yang masuk merupakan suara, GCF menjalankan fungsi Google Speech API. Setelah *processing* selesai dilakukan, Google Speech API mengembalikannya data

berupa *text*. Text kemudian dikirim hingga kembali ke Intent, dan kembali ke aplikasi dalam bentuk *object*. Aplikasi kemudian mengelola object tersebut dan menjalankan perintah yang diucapkan sebelumnya. Perintah dapat berupa menampilkan halaman tertentu ataupun menambahkan, mengubah, atau menghapus data.

3.3 Use Case Diagram

Use Case Diagram adalah gambaran dari interaksi antara actor yang merupakan pengguna dengan sistem. Dapat dilihat pada Gambar 3.2, actor yang ada dalam aplikasi ini hanya ada User. Aktivitas yang dapat dilakukan oleh User antara lain mengatur Jadwal, ToDoList, dan Note. Untuk melakukan aktivitas, User dapat melakukan Login atau tidak.



Gambar 3.2 Use Case Diagram Aplikasi

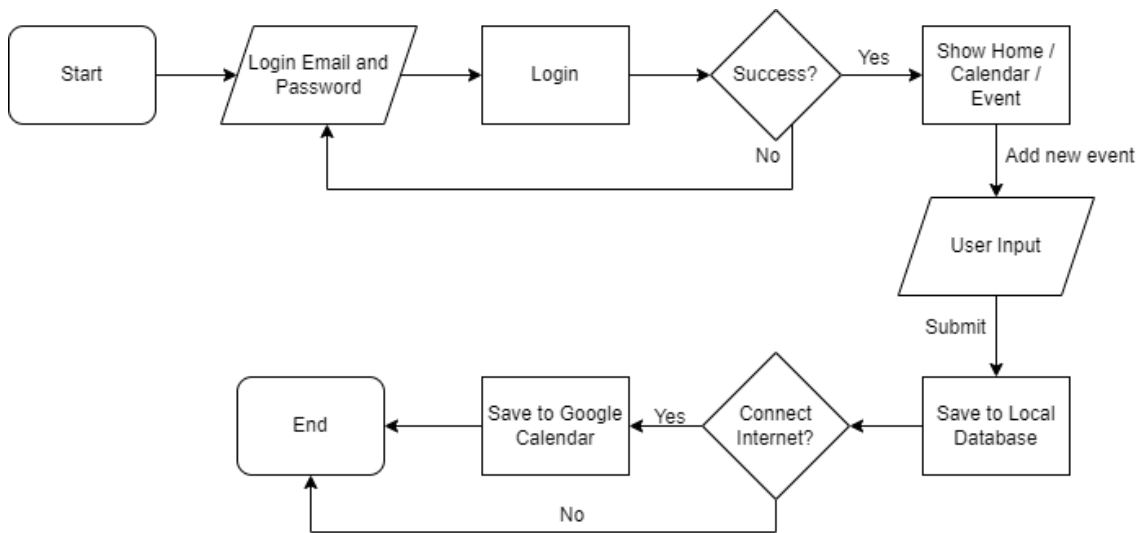
3.4 Flowchart

Flowchart adalah salah satu diagram yang menggambarkan suatu proses kerja suatu sistem. Pada sub bab ini membahas mengenai flowchart dari aktivitas yang disebutkan pada sub bab sebelumnya.

3.4.1 Flowchart Manage Jadwal

Proses kerja untuk mengatur jadwal dapat dilihat pada Gambar 3.3. Proses diawali dengan User melakukan *input email* dan *password* untuk login menggunakan Google. Kemudian aplikasi mengecek apakah proses login yang dilakukan berhasil. Jika iya maka User dialihkan ke

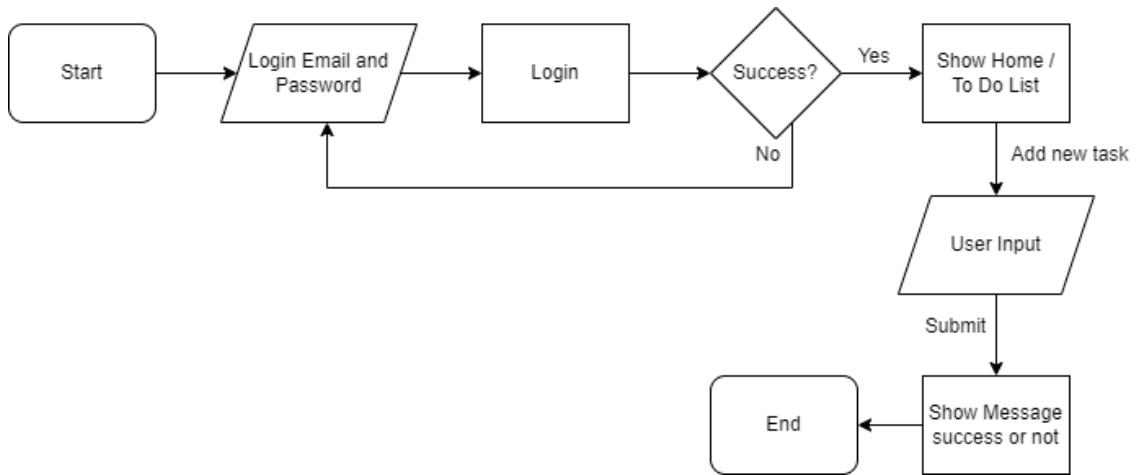
menu utama. Halaman utama menampilkan *calendar* atau *event* yang ada. User kemudian dapat menambahkan *event* baru. Setelah selesai *input* data, User dapat melakukan *submit* yang kemudian data disimpan pada *local database*. Aplikasi kemudian mengecek apakah tersambung dengan jaringan internet, jika iya maka data yang baru saja *di-input* ditambahkan juga ke Google Calendar.



Gambar 3.3 Flowchart Manage Jadwal

3.4.2 Flowchart ToDoList

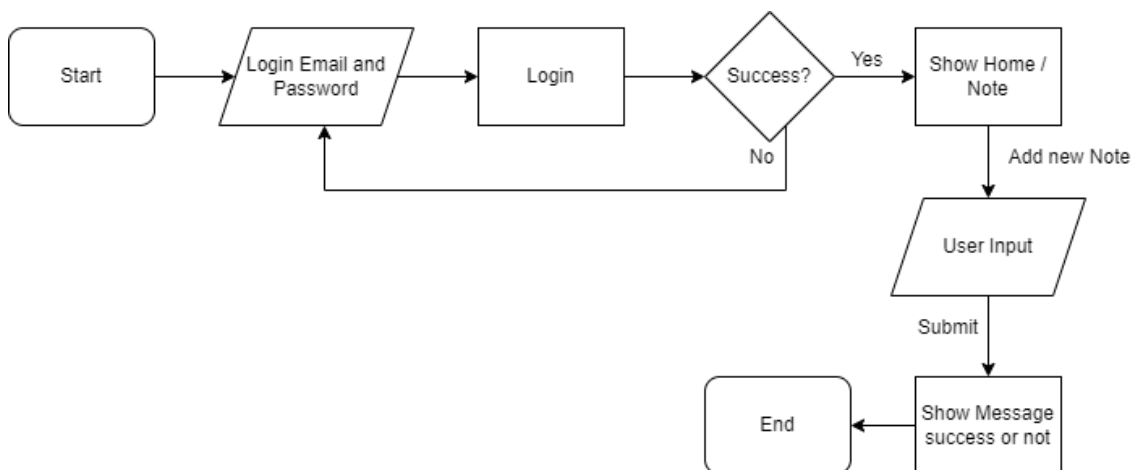
Proses kerja untuk mengatur ToDoList dapat dilihat pada Gambar 3.4. Proses juga diawali dengan melakukan proses *login* dengan *input email* dan *password* Google. Setelah login berhasil, User diarahkan menuju halaman utama (*Home*) dan menampilkan halaman ToDoList. Pada halaman ini, ditampilkan ToDoList yang sudah dibuat sebelumnya. User juga dapat menambahkan data baru dengan *input* sesuai kebutuhan. Setelah *submit* kemudian ditampilkan pesan berhasil atau gagal.



Gambar 3.4 Flowchart ToDoList

3.4.3 Flowchart Note

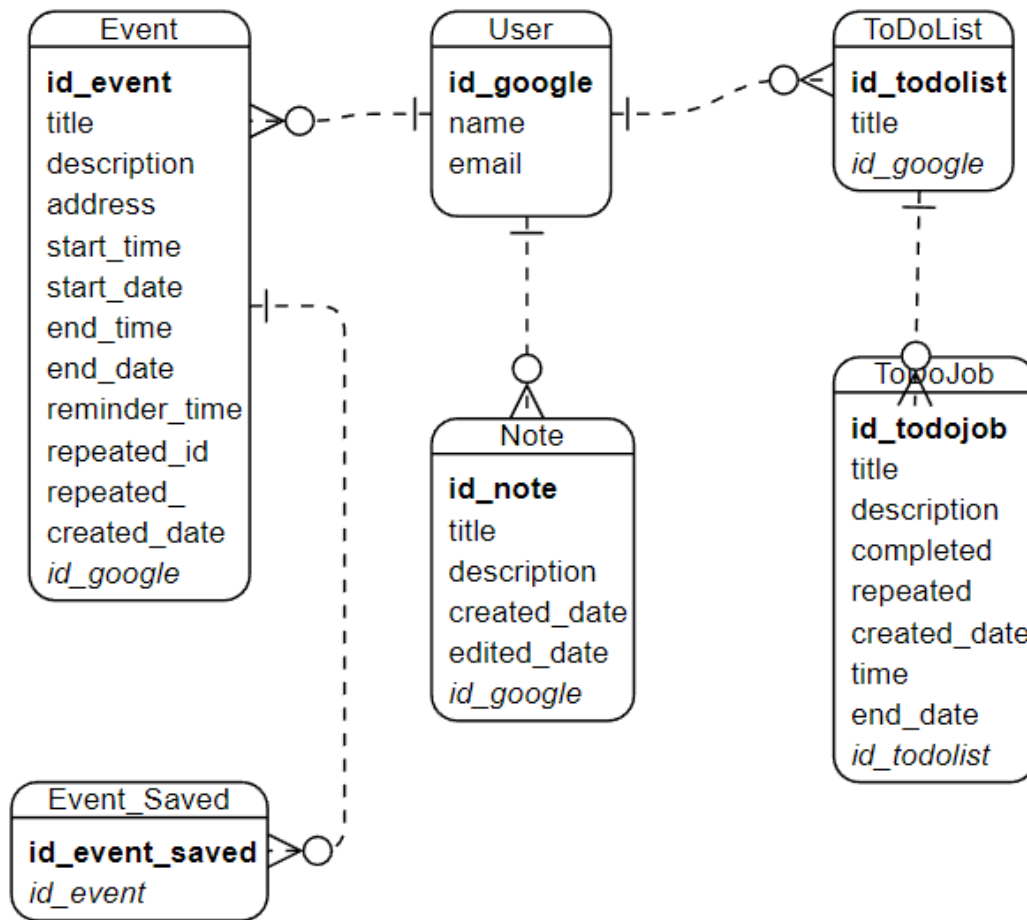
Proses kerja untuk mengatur *note* dapat dilihat pada Gambar 3.5. Diawali juga dengan proses login yang kemudian diarahkan kehalaman utama dan menampilkan halaman *note*. Pada halaman ini, aplikasi menampilkan *note* yang sudah dibuat sebelumnya. User juga dapat membuat *note* baru dengan input data yang dibutuhkan. Setelah *submit*, aplikasi menampilkan sebuah pesan berhasil atau tidak.



Gambar 3.5 Flowchart Note

3.5 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah diagram untuk membuat rancangan database. ERD yang dibuat dapat dilihat pada Gambar 3.6. Penjelasan setiap entitas dilanjutkan pada sub bab berikutnya yang membahas tentang *database*.



Gambar 3.6 Entity Relational Diagram

3.6 Database

Database merupakan kumpulan data atau informasi yang tersimpan secara sistematis. Sub bab ini akan menjelaskan setiap attribute yang ada pada Gambar 3.6.

3.6.1 Tabel User

Tabel *User* berfungsi untuk menyimpan informasi pribadi Google pengguna seputar nama, email, dan id. Informasi pribadi Google digunakan sebagai identifikasi pengguna untuk tabel database lainnya. Detail setiap *attribute* dan kegunaannya dijelaskan pada Tabel 3.1.

Tabel 3.1

Tabel User

No.	Attribute	Key	Tipe Data	Keterangan
1	Id_google	PK	String	Sebagai identifikasi pengguna

2	Name		String	Nama pengguna
3	email		String	Email pengguna

3.6.2 Tabel Event

Tabel *Event* berfungsi untuk menyimpan informasi acara atau kegiatan. Detail setiap *attribute* dan kegunaannya dijelaskan pada Tabel 3.2.

Tabel 3.2

Tabel Event

No.	Attribute	Key	Tipe Data	Keterangan
1	Id_event	PK	String	Sebagai identifikasi acara
2	Title		String	Judul acara
3	Description		String	Deskripsi acara
4	CreatedDate		String	Tanggal dibuat
5	StartDate		String	Tanggal dimulai
6	EndDate		String	Tanggal berakhir
7	TimeStart		String	Waktu dimulai
8	TimeEnd		String	Waktu berakhir
9	Repeated		String	Pengulangan acara seperti harian, mingguan, tahunan, atau tidak diulang
10	RepeatedId		String	Sebagai indentifikasi untuk mengambil acara yang diulang sebagai satu acara
11	ReminderTime		String	Waktu pengingat
12	Address		String	Alamat Acara
13	Id_calendar	FK	String	Sebagai identifikasi pengguna

3.6.3 Tabel Event Saved

Tabel *Event Saved* berfungsi untuk menyimpan informasi acara atau kegiatan yang belum terintegrasi dengan Google Calendar. Detail setiap *attribute* dan kegunaannya dijelaskan pada Tabel 3.3.

Tabel 3.3

Tabel Event Saved

No.	Attribute	Key	Tipe Data	Keterangan
1	IdEventSaved	PK	Int	Sebagai identifikasi event yang belum tersimpan
2	IdEvent	FK	String	Sebagai identifikasi event

3.6.4 Tabel ToDoList

Tabel ToDoList berfungsi untuk menyimpan informasi daftar kegiatan. Detail setiap *attribute* dan kegunaannya dijelaskan pada Tabel 3.4.

Tabel 3.4

Tabel ToDoList

No.	Attribute	Key	Tipe Data	Keterangan
1	Id_todolist	PK	Int	Sebagai identifikasi daftar kegiatan
2	Title		String	Judul daftar kegiatan
3	IdGoogle	FK	String	Sebagai identifikasi pengguna

3.6.5 Tabel ToDoJob

Tabel ToDoJob berfungsi untuk menyimpan kegiatan yang ada didalam daftar kegiatan. Detail setiap *attribute* dan kegunaannya dijelaskan pada Tabel 3.5.

Tabel 3.5

Tabel ToDoJob

No.	Attribute	Key	Tipe Data	Keterangan
1	Id_todojob	PK	Int (10)	Sebagai identifikasi kegiatan
2	Title		String	Judul kegiatan
3	Description		String	Deskripsi kegiatan
4	CreatedDate		String	Tanggal dibuat
5	EndDate		String	Tanggal tenggat kegiatan atau tugas

6	Time		String	Waktu tenggat
7	Repeated		String	Pengaturan pengulangan
8	Completed		Boolean	Apakah kegiatan sudah selesai atau belum
9	IdTodolist	FK	Int	Sebagai identifikasi daftar kegiatan

3.6.6 Tabel Note

Tabel *Note* berfungsi untuk menyimpan informasi catatan. Detail setiap *attribute* dan kegunaannya dijelaskan pada Tabel 3.6.

Tabel 3.6

Tabel Note

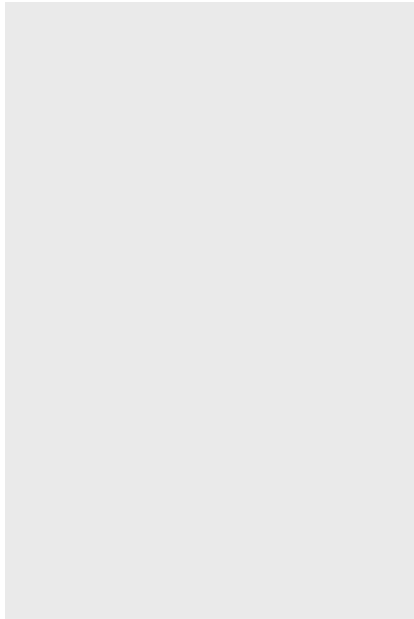
No.	Attribute	Key	Tipe Data	Keterangan
1	Id_note	PK	Int (10)	Sebagai identifikasi catatan
2	Title		String	Judul catatan
4	CreatedDate		String	Tanggal catatan dibuat
5	EditedDate		String	Tanggal catatan diubah
6	Description		String	Isi catatan
7	IdGoogle	FK	String	Email pengguna

3.7 User Interface

User Interface (UI) adalah tampilan yang biasa orang lihat pada suatu aplikasi. Pada sub bab ini di bahas tentang UI aplikasi.

3.7.1 Halaman Login

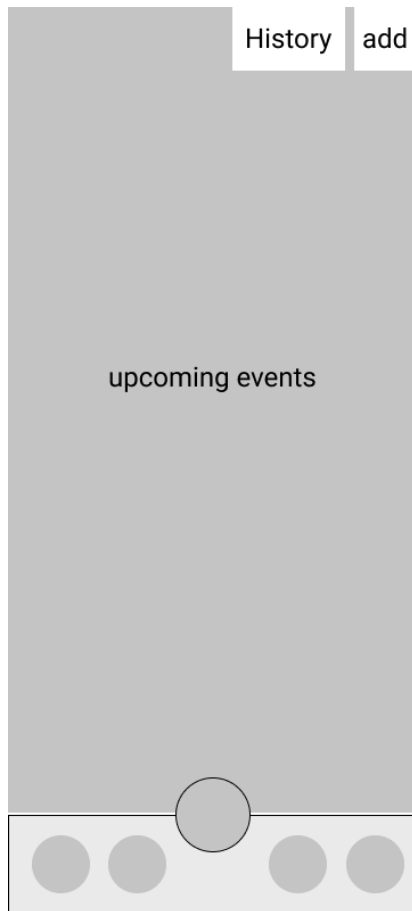
Halaman *Login* berisi tombol yang digunakan untuk *sign in* menggunakan akun Google. Ketika tombol ditekan, aplikasi menampilkan halaman *sign in* Google yang sudah disediakan oleh Google. Sketsa dari halaman *login* dapat dilihat pada Gambar 3.7.



Gambar 3.7 Halaman Login

3.7.2 Halaman *Calendar*

Halaman *Calendar* berisi kalender, daftar acara, tombol tambah, dan tombol riwayat. Kalender menampilkan jadwal atau kegiatan yang belum selesai atau mendatang. Tombol tambah berfungsi untuk berpindah ke halaman tambah jadwal baru. Tombol riwayat berfungsi untuk berpindah ke halaman riwayat jadwal. Setiap jadwal dapat dilihat detailnya dengan menekan jadwal tersebut. Sketsa dari halaman *calendar* dapat dilihat pada Gambar 3.8.



Gambar 3.8 Halaman Utama Calendar

Halaman tambah jadwal baru berisi form untuk menambahkan jadwal baru. *Input* yang harus diisi adalah judul jadwal, tanggal dan waktu. *Input* yang tidak wajib diisi adalah pengulangan jadwal, pengaturan durasi jadwal, deskripsi, dan alamat. Pengulangan yang dimaksud adalah jadwal diulang harian, minggu, bulanan, atau tahunan. Pengaturan durasi kegiatan adalah *input* untuk jam dilaksanakannya jadwal dan jam berakhirnya jadwal. Halaman ini juga berperan sebagai halaman detail jadwal. Sketsa dari halaman tambah dapat dilihat pada Gambar 3.9.

Input event title

Input event date

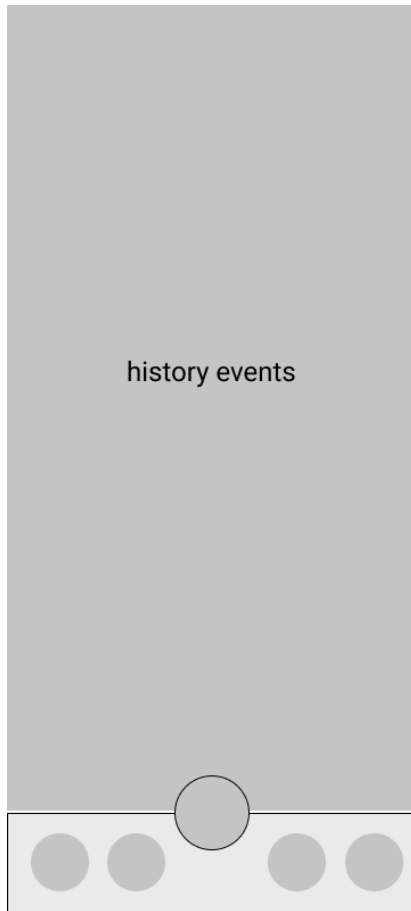
Input event time

Input event additional note

submit

Gambar 3.9 Halaman Tambah Jadwal Calendar

Halaman riwayat kegiatan berisikan daftar kegiatan yang sudah lewat atau lampau. Halaman menampilkan acara dalam periode harian yang diikuti dengan kegiatan pada jam tertentu. Sketsa dari halaman riwayat dapat dilihat pada Gambar 3.10.

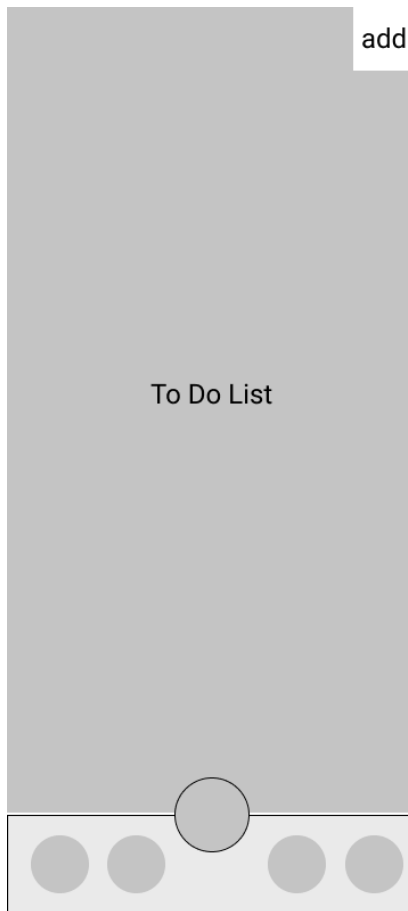


Gambar 3.10 Halaman Riwayat Kegiatan Calendar

3.7.3 Halaman ToDoList

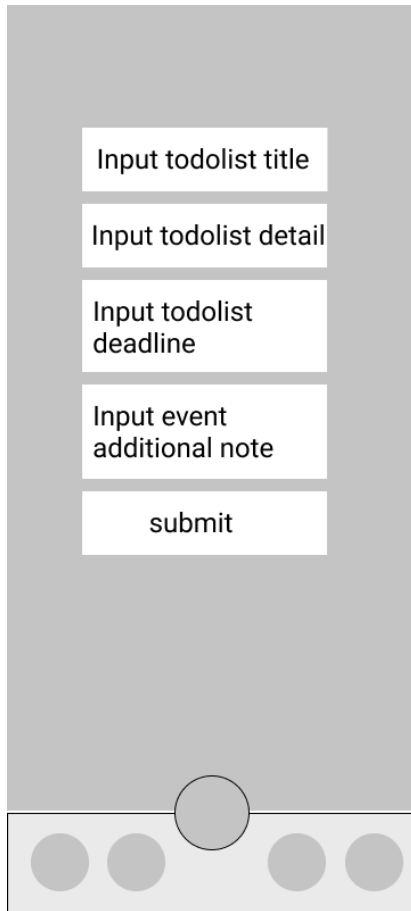
Halaman ToDoList berisikan tentang daftar kegiatan yang sebelumnya sudah ditambahkan. Halaman ini juga menampilkan kegiatan yang sudah dicentang atau kegiatan yang sudah selesai. Pada halaman ini juga berisi tombol tambah (+) yang berguna untuk berpindah ke halaman untuk menambahkan kegiatan baru. Ada juga tombol dengan *icon* lonceng yang menampilkan sebuah tampilan sebagai pengaturan waktu pengingat untuk kegiatan yang memiliki tenggat waktu.

Setiap kegiatan yang sudah dibuat memiliki sebuah *icon* yang dapat ditekan untuk mengubah status kegiatan menjadi selesai. Jika menekan suatu kegiatan hingga lebih dari 2 detik, maka muncul *icon* tempat sampah dan silang. Tempat sampah berfungsi untuk menghapus kegiatan tersebut, dan silang untuk menutup *icon-icon* tersebut. Sketsa dari halaman ToDoList dapat dilihat pada Gambar 3.11.



Gambar 3.11 Halaman Utama ToDoList

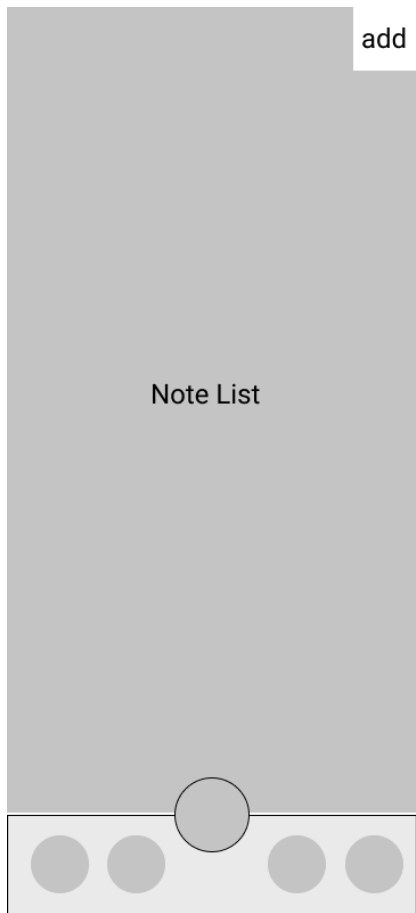
Halaman Tambah ToDoList berisikan form yang berfungsi untuk menambahkan kegiatan baru. *Input* yang harus diisi adalah judul kegiatan, input tambahan dapat berupa catatan tambahan, tenggat waktu, atau pengulangan kegiatan. Pada halaman ini juga dapat menambahkan sebuah daftar baru seperti daftar belanjaan, ataupun daftar buku sekolahan yang harus dibawa. Daftar kegiatan yang selalu ada adalah “Master List” yang bersifat permanen. Terdapat *icon* sampah dan tambah yang berfungsi untuk menampilkan menu untuk menghapus dan menambahkan daftar. Halaman ini juga berfungsi sebagai halaman detail kegiatan. Sketsa dari halaman tambah ToDoList dapat dilihat pada Gambar 3.12.



Gambar 3.12 Halaman Tambah ToDoList

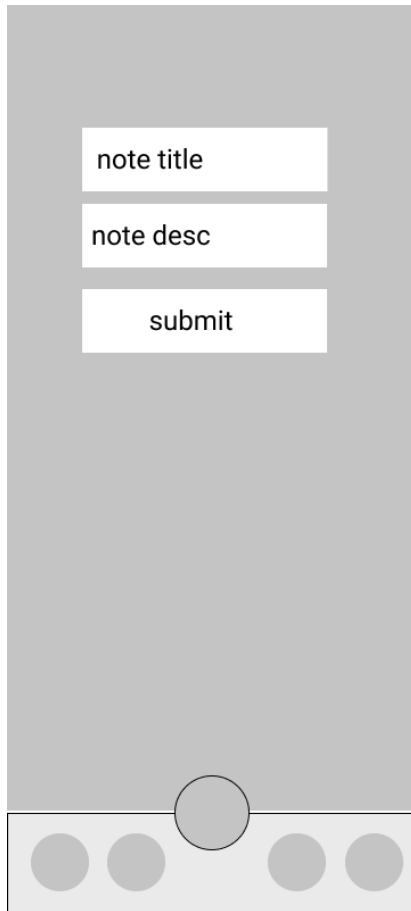
3.7.4 Halaman Note

Halaman Note berisikan daftar catatan yang telah dibuat sebelumnya. Ada juga sebuah tombol untuk membuat catatan baru. Setiap catatan juga menampilkan tanggal terakhir diubah. Sketsa dari halaman Note dapat dilihat pada Gambar 3.13.



Gambar 3.13 Halaman Utama Catatan

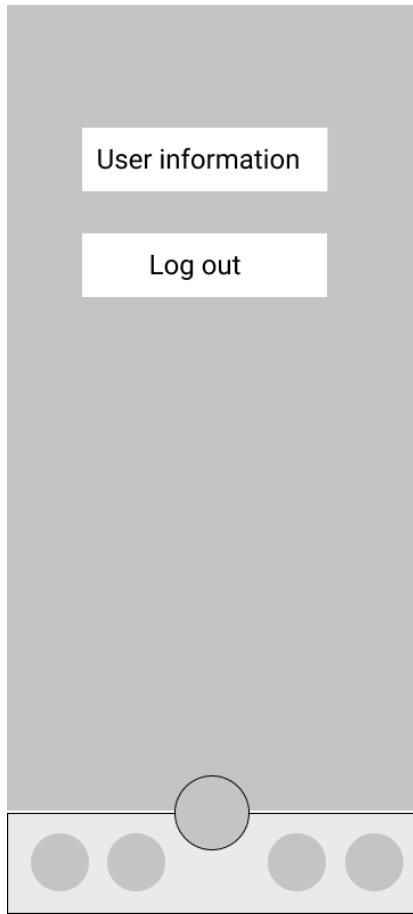
Halaman membuat note berisi form untuk membuat catatan baru. Input yang dapat diisi adalah judul catatan, dan isi catatan. Sketsa dari halaman tambah Ntoe dapat dilihat pada Gambar 3.14. Pengguna juga dapat mengisi note menggunakan *mic* yang berada pada bawah bagian tengah.



Gambar 3.14 Halaman Tambah Catatan

3.7.5 Halaman Profile

Halaman *Profile* berisi informasi pengguna dan halaman untuk melakukan *sign out* atau keluar dari akun. Sketsa halaman *profile* dapat dilihat pada Gambar 3.15.



Gambar 3.15 Halaman Profile