

BAB III

Metodologi Penelitian

3.1 Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh beberapa data yang dijadikan sebagai dasar acuan untuk mengembangkan aplikasi yang akan dibuat. Dalam pengumpulan data metode yang dilakukan adalah studi pustaka.

Studi pustaka dilakukan untuk menunjang hasil observasi dan dijadikan sebagai landasan literatur dalam penelitian. Sumber studi pustaka didapatkan dari beberapa buku, artikel, jurnal, maupun informasi dari internet. Sumber utama studi pustaka dalam penelitian ini adalah perhitungan penanggalan kalender islam melalui berbagai metode untuk mendapatkan hasil yang relevan serta aplikasi-aplikasi yang digunakan untuk membangun aplikasi android yang dimaksud. Hasil dari proses studi pustaka ini kumpulan beberapa informasi yang relevan dengan topik bahasan penelitian.

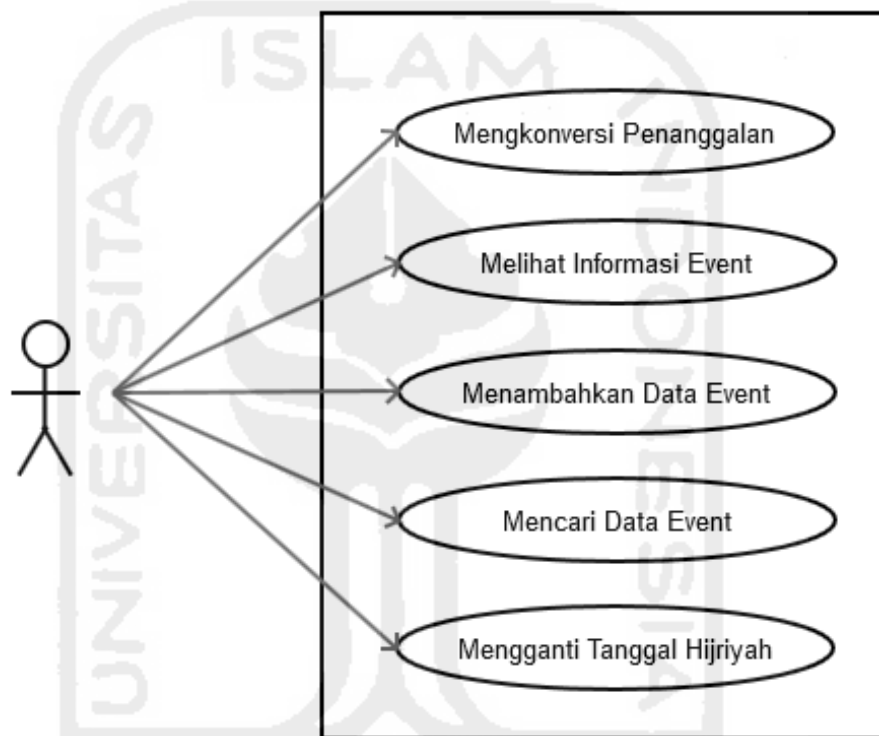
3.2 Perancangan Arsitektur Diagram

Perancangan arsitektur diagram merupakan suatu cara untuk menguraikan perancangan dari sebuah sistem. Tahap ini merupakan tahap penting sebelum membuat sebuah sistem. Pada tahap ini akan ditentukan metode yang akan digunakan dalam pembuatan aplikasi. Sehingga dapat mengevaluasi permasalahan-permasalahan dalam pembuatan aplikasi

Metode yang digunakan adalah metode perancangan berbasis objek dengan pemodelan *Unified Modelling Language* (UML). Bentuk diagram yang akan digunakan untuk pemodelan adalah *use case diagram*.

3.2.1 Use case diagram

Use case diagram menggambarkan seluruh aktivitas yang dapat dilakukan oleh pengguna terhadap sistem yang telah dibuat. *Use case diagram* untuk aplikasi ini dapat dilihat pada gambar 3.7.



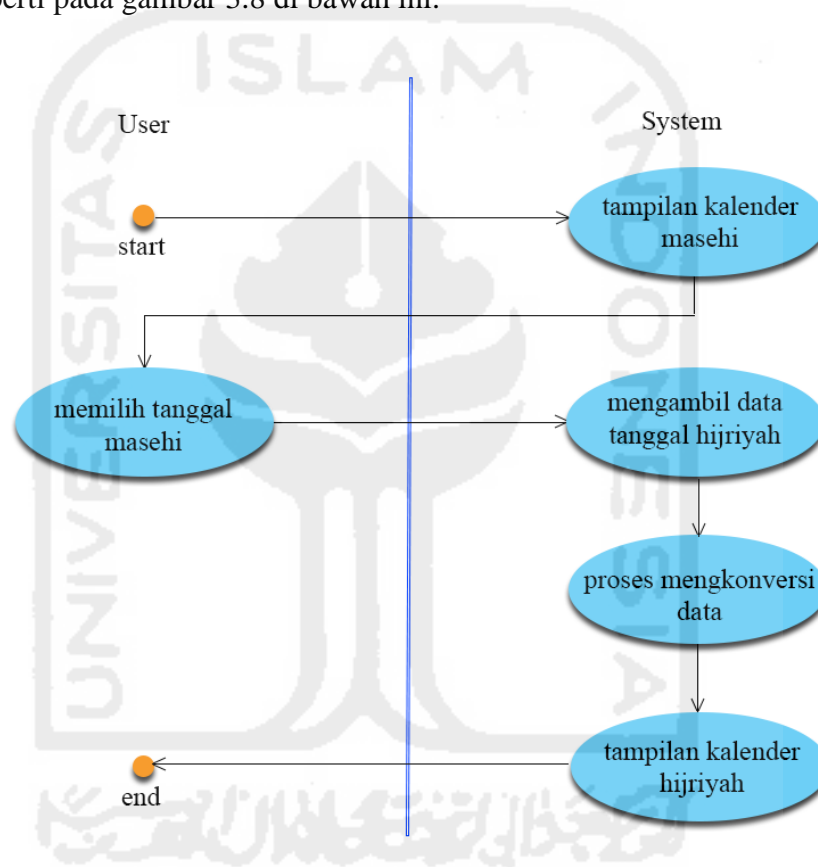
Gambar 3.1 Use case diagram

3.2.2 Activity diagram

Activity diagram merupakan suatu bentuk flow diagram atau diagram alir yang memodelkan alur kerja sebuah proses penggunaan aplikasi serta aktivitas sebuah proses yang dilakukan oleh *user*.

a. *Activity diagram* Konversi Penanggalan

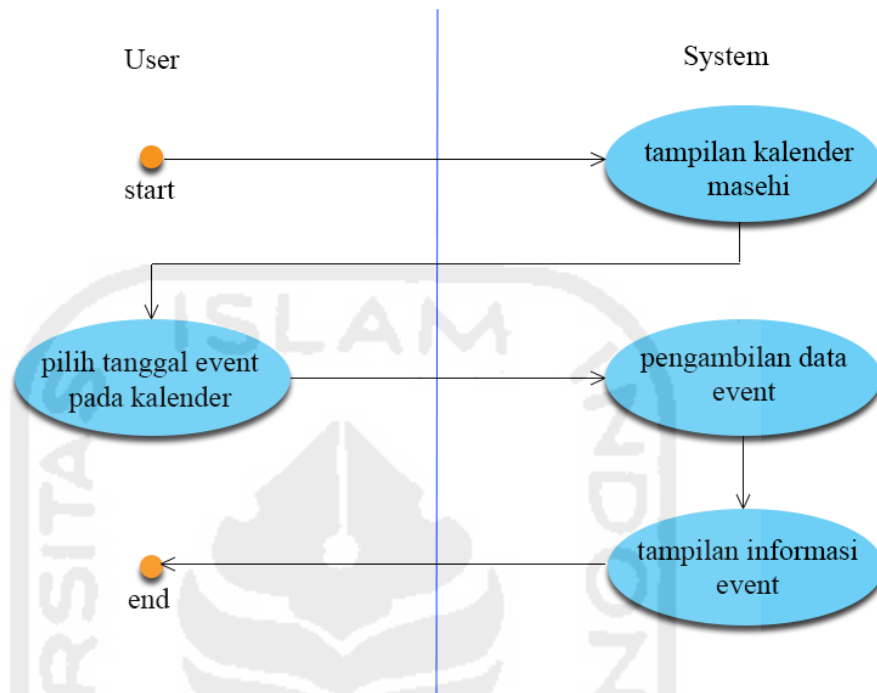
Pada *activity diagram* untuk konversi penanggalan, *user* pertama akan melihat tampilan kalender masehi. Kemudian *user* dapat memilih tanggal masehi untuk di konversi. Setelah memilih tanggal masehi kemudian *user* akan mendapatkan tanggal hijriyah dalam bentuk tampilan kalender hijriyah. Seperti pada gambar 3.8 di bawah ini.



Gambar 3.2 *Activity diagram* Konversi Penanggalan

b. *Activity diagram* Melihat Informasi Event

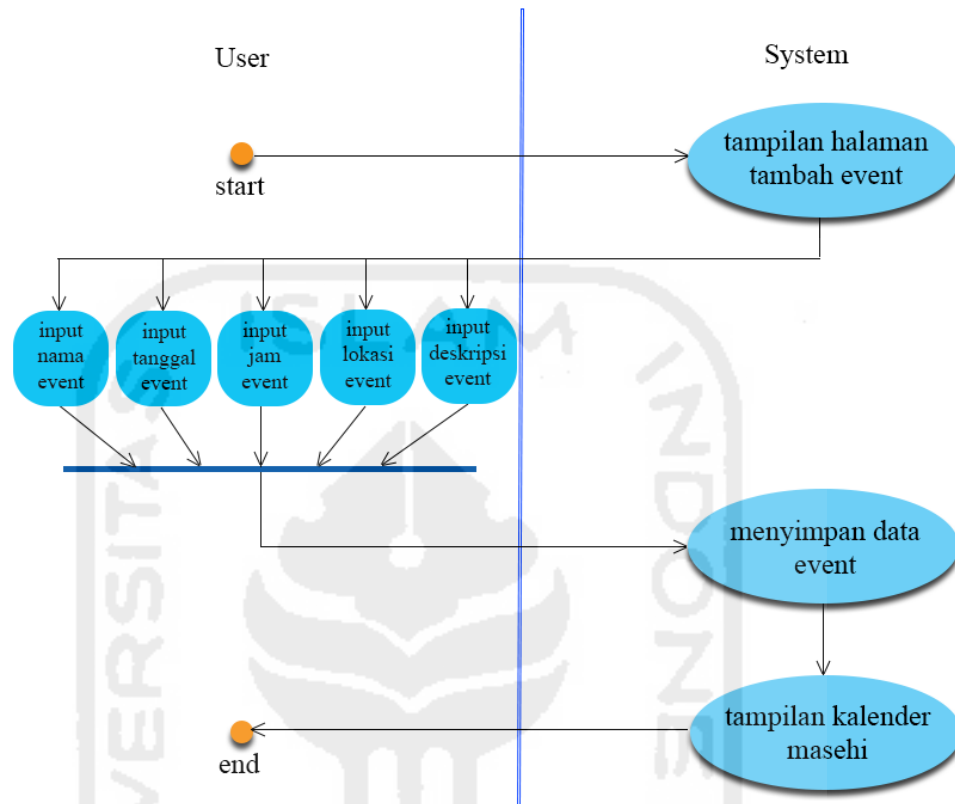
User dapat melihat informasi *event* yang sudah ada untuk melihat *event* secara detail. Untuk melihat informasi *event*, *user* dapat memilih tanggal yang sudah ada *event* didalamnya. untuk lebih jelasnya dapat dilihat pada gambar 3.9 di bawah ini.



Gambar 3.3 Activity diagram Melihat Informasi Event

c. Activity diagram Menambahkan Data Event

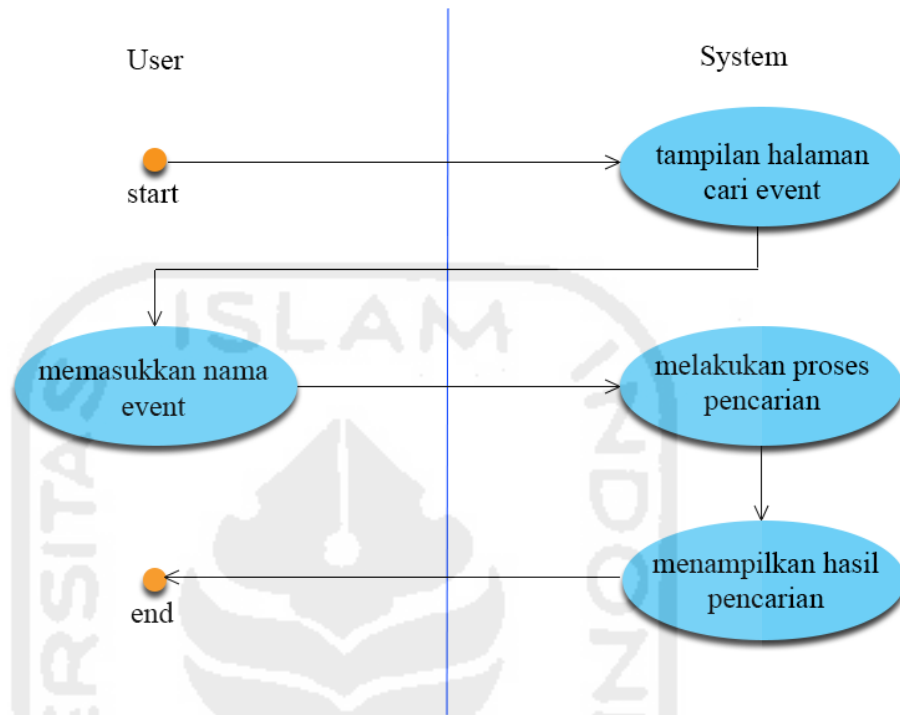
Untuk menambahkan *event*, *user* dapat menambakkannya melalui halaman tambah *event* yang didalamnya berisi detail-detail *event* yang harus diisi oleh *user*. Setelah itu data disimpan dan ditampilkan pada kalender. Agar lebih jelasnya dapat dilihat pada gambar 3.10 berikut.



Gambar 3.4 Activity diagram Menambahkan Data Event

d. Activity diagram Mencari Data Event

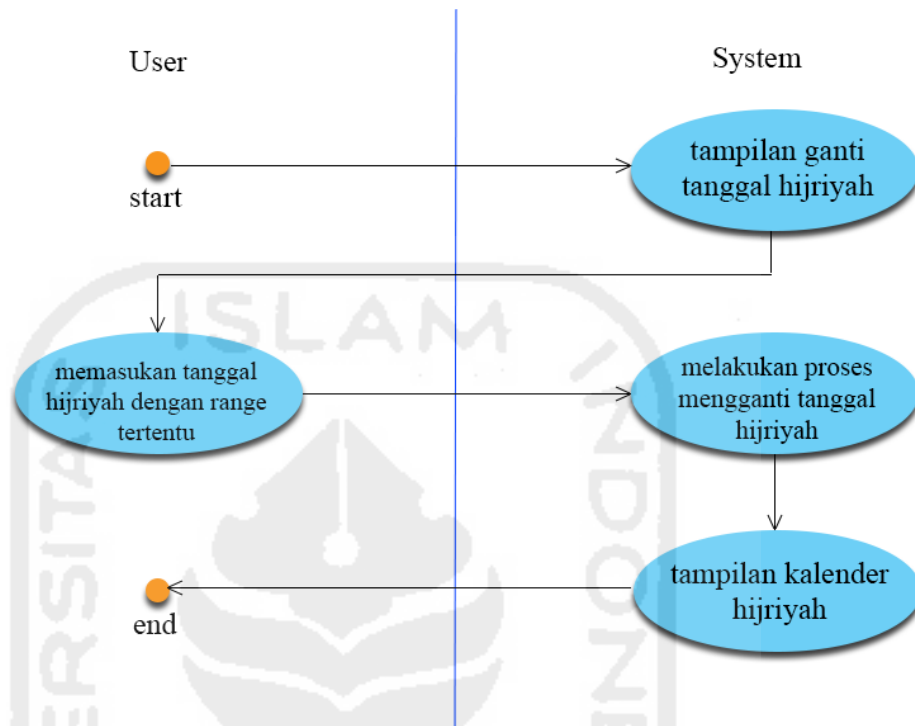
User dapat mencari data event untuk memudahkan user dalam mencari event yang ingin dicari dengan cara pindah ke halaman *search* kemudian memasukkan nama event yang ingin dicari. Seperti yang dijelaskan pada gambar 3.11 di bawah ini.



Gambar 3.5 Activity diagram Mencari Data Event

e. Activity diagram Mengganti Tanggal Hijriyah

User juga dapat mengganti tanggal hijriyah jika ada ketidaksesuaian hari pada penanggalan hijriyah. User tinggal masuk ke menu untuk mengganti tanggal hijriyah kemudian dapat mengubah tanggal berdasarkan range yang telah ditentukan sebelumnya. Untuk activity diagram mengganti tanggal hijriyah dapat dilihat pada gambar 3.12 di bawah ini.



Gambar 3.6 Activity diagram Mengganti Tanggal Hijriyah

3.2.3 Perancangan Tabel Database

a. Tabel Event

Tabel *event* berfungsi untuk menyimpan semua informasi *event* yang ditambahkan oleh *user* pada kalender masehi maupun hijriyah. Struktur dari tabel *event* dapat dilihat pada tabel 3.1.

Tabel 3.1 Tabel Event

Nama Kolom	Tipe	Keterangan
<i>id_event</i>	Int(3)	Primary key
<i>nama_event</i>	Varchar(50)	

tanggal_mulai	Date	
tanggal_selesai	Date	
jam_mulai	Varchar(10)	
jam_selesai	Varchar(10)	
date_type	Boolean	
event_location	Text	
event_description	Text	
id_user	Int(3)	

b. Tabel *user*

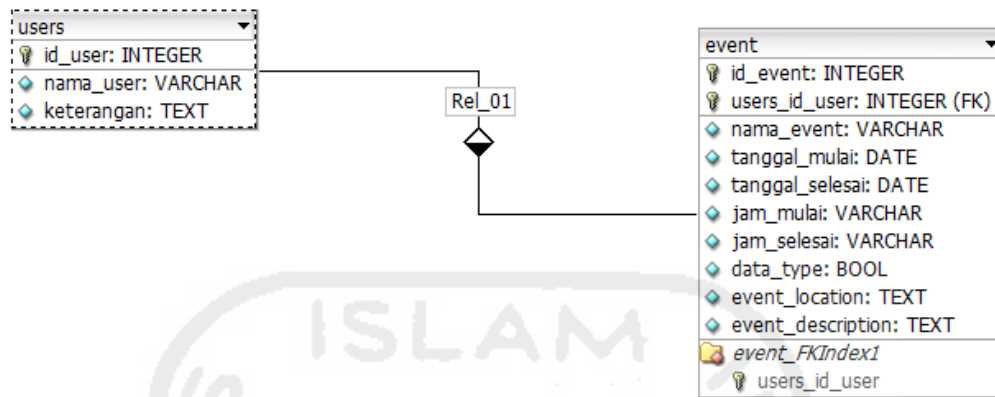
Tabel *user* berfungsi untuk menyimpan semua data *user* yang digunakan pada aplikasi. Struktur dari tabel *user* dapat dilihat pada tabel 3.2.

Tabel 3.2 Tabel *user*

Nama Kolom	Tipe	Keterangan
id_user	Int(3)	Primary key
Nama	Varchar(50)	
Keterangan	text	

3.2.4 Relasi Antar Tabel

Relasi antar tabel dari tabel *event* dan tabel *user* menghubungkan id_user dengan menjadi *foreign key* pada tabel *event* yang dapat dilihat pada gambar 3.13 di bawah ini :



Gambar 3.7 Relasi Antar Tabel

3.3 Identifikasi Kebutuhan Dasar

Sesuai dengan topik dalam penelitian ini yaitu pembuatan aplikasi Qalendary *Switchable Calendar*, maka identifikasi kebutuhan dasar meliputi analisis kebutuhan proses untuk mengetahui proses apa saja yang diperlukan dalam aplikasi serta analisis kebutuhan software untuk mengetahui software apa saja yang digunakan dan saling keterkaitan software satu dengan yang lainnya.

3.3.1 Analisis Kebutuhan Proses

Kebutuhan proses yang dilakukan oleh aplikasi dapat dilihat pada tabel 3.1 di bawah ini.

Tabel 3.1 Analisis Kebutuhan Proses

No	Proses	Keterangan
1	Proses Menambahkan Data Annual Event	User menambahkan data seperti nama event, tipe kalender, tanggal event, lokasi event, deskripsi event yang berulang tiap tahunnya
2	Proses Menambahkan Data	User menambahkan data seperti nama event,

	Nonannual <i>Event</i>	tipe kalender, tanggal mulai <i>event</i> , tanggal selesai <i>event</i> , lokasi <i>event</i> , deskripsi <i>event</i> namun <i>event</i> hanya terjadi sekali
3	Proses Menghapus Data <i>Event</i>	<i>User</i> dapat menghapus data <i>event</i> baik yang annual <i>event</i> maupun yang nonannual <i>event</i>
4	Proses Mengedit Data <i>Event</i>	<i>User</i> dapat mengedit data <i>event</i> untuk mengganti data yang ingin diubah
5	Proses Mencari Data <i>Event</i>	<i>User</i> dapat mencari <i>event</i> untuk dapat melihat informasi <i>event</i> , menghapus, atau pun mengedit <i>event</i>
6	Proses Menampilkan Data <i>Event</i> pada Kalender Masehi	Data <i>event</i> yang telah dimasukkan oleh <i>user</i> kemudian ditampilkan pada kalender masehi jika tipe kalender yang dipilih adalah masehi
7	Proses Menampilkan Data <i>Event</i> pada Kalender Hijriyah	Data <i>event</i> yang telah dimasukkan oleh <i>user</i> kemudian ditampilkan pada kalender hijriyah jika tipe kalender yang dipilih adalah hijriyah
8	Proses Mengkonversi Tanggal dari Kalender Masehi ke Hijriyah	<i>User</i> dapat mengkonversi tanggal dari tanggal yg di pilih pada kalender masehi kemudian ditampilkan dalam kalender hijriyah
9	Proses Mengkonversi Tanggal dari Kalender Hijriyah ke Masehi	<i>User</i> dapat mengkonversi tanggal dari tanggal yg di pilih pada kalender hijriyah kemudian ditampilkan dalam kalender masehi

3.3.2 Analisis Kebutuhan Software

Dalam pembuatan aplikasi ini dibutuhkan software yang dapat digunakan untuk membuat dan mengembangkan aplikasi android. Ada beberapa software yang dapat membuat dan mengembangkan aplikasi android diantaranya Android Studio, Eclipse, App Inventor, App Geysler, dan Ionic Framework.

Dari beberapa software tersebut penulis memilih ionic framework untuk melakukan pembuatan dan pengembangan aplikasi android dengan beberapa pertimbangan. Salah satunya adalah menggunakan HTML5 untuk penulisan *source code* dan memiliki komponen/elemen CSS standar digunakan untuk smartphone, seperti button, list, card, form, range, tabs, grid dan lainnya.

Kelebihan lain Ionic Framework selain yang telah disebutkan di atas dibanding software lainnya adalah Ionic memanfaatkan AngularJS untuk implementasi logiknya. Dengan menggunakan AngularJS performa dan respons lebih cepat sehingga akan seperti menggunakan aplikasi *native* yang pada dasarnya Ionic merupakan pembuat aplikasi *hybrid mobile*.

Ionic Framework membutuhkan aplikasi lain untuk mendukung pembuatan aplikasi android. Ionic membutuhkan Node.js yang digunakan untuk menjalankan perintah-perintah Ionic seperti *SCSS compiling*, *serve app* untuk *testing*, *generate icon resources*, dan banyak lainnya. Karena Ionic menggunakan javascript, maka Ionic membutuhkan Node.js.

Java JDK juga merupakan software yang dibutuhkan oleh ionic. *Java JDK*, software yang digunakan untuk membangun aplikasi-aplikasi *java*. Karena pada dasarnya semua aplikasi android menggunakan java sehingga Ionic membutuhkan Java JDK.

Kemudian software selanjutnya yang dibutuhkan oleh Ionic adalah Apache Ant. Apache Ant adalah software berbasis *Java* yang digunakan untuk keperluan build *tool*. Sebagai build *tool*, tugas utama dari Ant adalah untuk menyediakan sumber daya dan melaksanakan proses yang memungkinkan membangun suatu

software dari bentuk *source code* menjadi aplikasi yang siap didistribusikan atau bahkan telah diinstall pada *remote server*.

Software lainnya yang dibutuhkan oleh Ionic Framework adalah Android SDK. Untuk membuat aplikasi android, instalasi Android SDK juga diperlukan. Android SDK menyediakan API *library* dan alat-alat pengembang yang diperlukan untuk membangun, menguji, dan *debug* aplikasi Android.

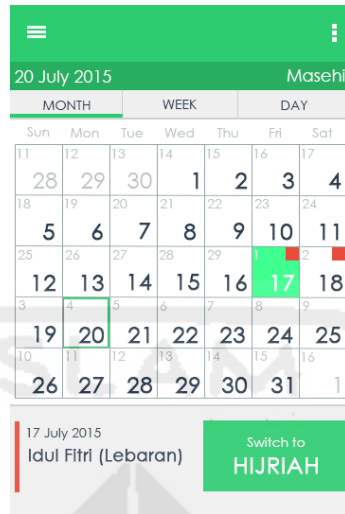
Untuk penyimpanan *database* penulis memilih menggunakan Firebase. Firebase merupakan *database* yang menyediakan API untuk memungkinkan pengembang menyimpan dan sinkron data lewat *multiple client*. Keunggulan Firebase dibanding layanan *database* lainnya adalah *realtime database*. Firebase menyediakan *realtime database* dan *backend* sebagai layanan utama.

3.4 Perancangan Antarmuka

Pada tahap perancangan antarmuka ini akan diberikan gambaran umum tentang perancangan antarmuka aplikasi yang akan ditampilkan pada *user*. Adapun perancangan antarmuka sebagai berikut :

3.4.1 Halaman Kalender Masehi

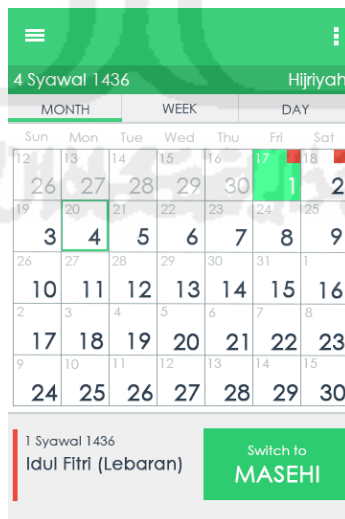
Saat pertama kali aplikasi dijalankan, maka akan muncul halaman yang menampilkan kalender masehi beserta tanggal pada hari itu. Kemudian dibagian bawah terdapat nama *event* untuk tanggal yang dipilih. Di sebelah nama *event* terdapat tombol switch to Hijriyah seperti pada gambar 3.14 di bawah ini



Gambar 3.8 Halaman Kalender Masehi

3.4.2 Halaman Kalender Hijriyah

Pada halaman kalender hijriyah ini akan ditampilkan setelah menekan tombol switch to hijriyah. Halaman ini menampilkan kalender hijriyah secara menyeluruh. Sama seperti kalender masehi, nama *event* tanggal yg dipilih dan tombol *switch to masehi* untuk kembali ke tampilan kalender masehi yang ditunjukkan pada gambar 3.15



Gambar 3.9 Halaman Kalender Hijriyah

3.4.3 Halaman Input *Event*

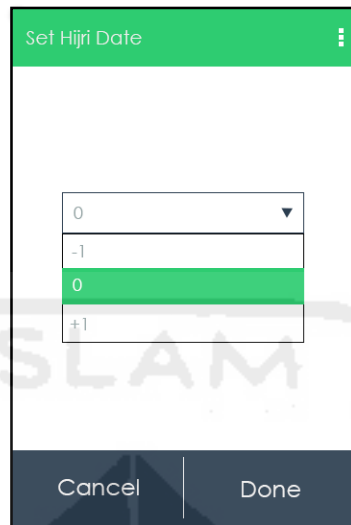
Pada halaman input *event* terdapat beberapa kolom untuk mendeskripsikan *event* yang akan dibuat seperti nama *event*, jam terjadinya *event*, lokasi *event*, deskripsi *event* dan lain-lain. Gambar halaman input *event* dapat dilihat pada gambar 3.16 di bawah ini.



Gambar 3.10 Halaman Input *Event*

3.4.4 Halaman *Set Hijri Date*

Pada halaman ini terdapat 3 pilihan pada kolom untuk mengubah hari pada kalender hijriyah. Menu ini digunakan untuk menyesuaikan kalender hijriyah. Gambar halaman *set hijri date* dapat dilihat pada gambar 3.17 di bawah ini



Gambar 3.11 Halaman *Set Hijri Date*

3.4.5 Halaman *Search Event*

Pada halaman *Search Event*, pengguna dapat mencari *event* yang diperlukan. Di halaman ini terdapat kolom searc untuk *user* mengetikkan nama *event* tersebut. Gambar halaman *search event* dapat dilihat pada gambar 3.18 di bawah ini.



Gambar 3.12 Halaman *Search Event*