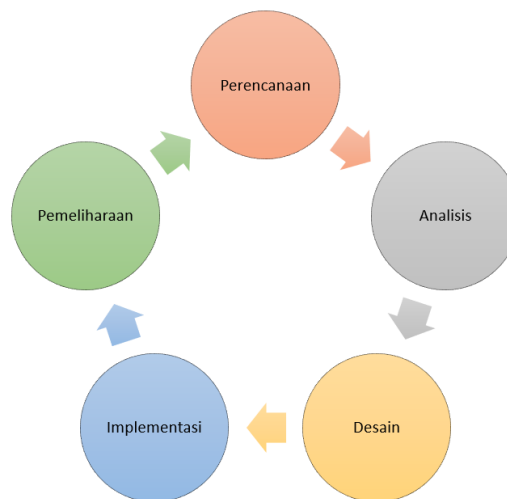


## BAB III METODOLOGI PENELITIAN

### 3.1. Metode Pengembangan Sistem

Dalam proses pengembangan sistem, penelitian ini menggunakan metode SDLC (*Software Development Life Cycle*) untuk siklus pengembangan sistem. Metode SDLC yang digunakan dalam pengembangan ini yaitu *Iterative Model*. Model ini merupakan proses pengulangan dari pengembangan *software*, artinya *software* akan berevolusi selama proses pengembangan sesuai dengan *requirement* yang dibutuhkan sampai sistem selesai. Seperti namanya, model ini mempunyai beberapa fase seperti pada gambar 3.1, yang akan berulang tergantung dibagian mana yang akan dikembangkan.



**Gambar 3.1** Model *Iterative* (Pengulangan)

### 3.2. Alat dan Bahan

Dalam pengembangan aplikasi dibutuhkan beberapa alat dan bahan untuk mendukung berjalannya perancangan dan implementasi aplikasi *mobile learning*.

#### 3.2.1 Alat

Berikut adalah spesifikasi Perangkat Keras (*Hardware*) PC (*Personal Computer*) atau laptop yang digunakan oleh peneliti, yaitu:

- a. *Processor*: Intel(R) Core(TM) i3-4030U CPU @ 1.90GHz, 1900 Mhz, 2 Core(s), 4 Logical Processor(s).
- b. 6,0 GB RAM.
- c. Windows 10 Enterprise.

Berikut adalah spesifikasi Perangkat Keras (*Hardware*) *Smartphone* yang digunakan oleh peneliti, yaitu:

- a. 4,0 GB RAM.
- b. *Eight core Processor*.
- c. OS (*Operating System*) Android.

Berikut adalah spesifikasi Perangkat Lunak (*Software*) yang digunakan oleh peneliti, yaitu:

- a. Microsoft Visio.

Merupakan sebuah aplikasi keluaran Microsoft untuk membuat diagram dan grafik vektor.

- b. CourseLab 3.

Merupakan software untuk membuat konten *e-learning* yang interaktif dan dapat dipublikasikan di internet, *Learning Management System* (LMS), *CD-ROM*, dan perangkat lainnya.

- c. PhoneGap.

Merupakan sebuah kerangka untuk pengembangan software yang dikeluarkan Adobe. PhoneGap akan menghasilkan aplikasi untuk semua platform OS (*Operating System*) yang populer dari hasil *compiler* bahasa pengembangan *web*.

- d. Notepad++

Merupakan sebuah teks dan sumber kode editor untuk mendukung pengeditan *tab* dan memungkinkan bekerja dengan *multiple file* yang dibuka.

- e. Balsamiq Mockup

Merupakan alat *wireframe* untuk membuat rancangan UI (*User Interface*) beserta fungsi dan *widget* dari konten yang dirancang.

### 3.2.2 Bahan

Bahan yang digunakan adalah kumpulan data yang didapat dari melakukan percobaan terhadap aplikasi dan metode umum. Jumlah data tersebut dilakukan terhadap pengguna dari berbagai kalangan. Jumlah pengguna yang melakukan percobaan tersebut sebanyak 30 pengguna.

### 3.3. Desain Arsitektur Sistem

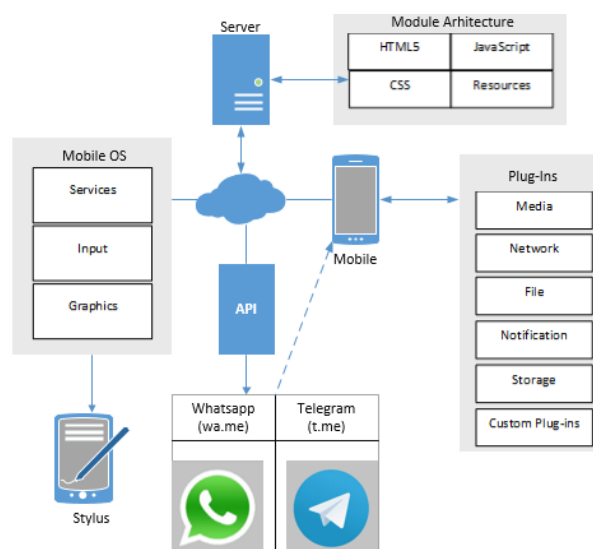
Dalam pengimplementasian, aplikasi ini diberi nama “Stylus-Q” sebagai nama modul *mobile learning*. Tujuan utama mengembangkan aplikasi ini adalah untuk membantu *user* dalam belajar menulis Al-Qur’an sekaligus meningkatkan minat siswa dalam pada pembelajaran Al-Qur’an. Aplikasi ini didesain berdasarkan android karena siswa akan memilih teknologi terkini untuk meningkatkan efisiensi belajar (Tarhini et al, 2016) dan siswa lebihantisipasi untuk mempelajari materi yang diberikan serta dapat dilakukan kapanpun dimanapun (Kurniawati & Priyanto, 2018).

Aplikasi ini dikembangkan dengan menggunakan *software* CourseLab sebagai implementasi media pembelajaran. Kemudian, PhoneGap digunakan untuk membangun aplikasi seluler dengan menggunakan bahasa pemrograman *web* modern, seperti HTML, HTML5, CSS, dan JavaScript daripada menggunakan bahasa lain yang terkenal seperti C# atau Java (Wang & Ma, 2014). Hasil akhir dari kombinasi ini berupa berkas APK (*Application Package File*) yang dapat dijalankan di *smartphone*.

Pada aplikasi ini, langkah pertama yang dilakukan yaitu mendesain arsitektur dari sistem untuk mendeskripsikan struktur, interaksi, dan teknologi dari komponen sistem komputer (Burd, 2010). Setelah itu, langkah berikutnya adalah merancang sistem proses kerja modul siswa untuk memenuhi *requirement* yang dibutuhkan. Pengembangan aplikasi ini juga dirancang dengan menggunakan UML (*Unified Modelling Language*) sebagai standar pemodelan visualisasi, dan dokumentasi sistem (Dharwiyanti & Wahono, 2003) dan menerapkan beberapa prinsip mendesain seperti *visibility* untuk membuat *user*

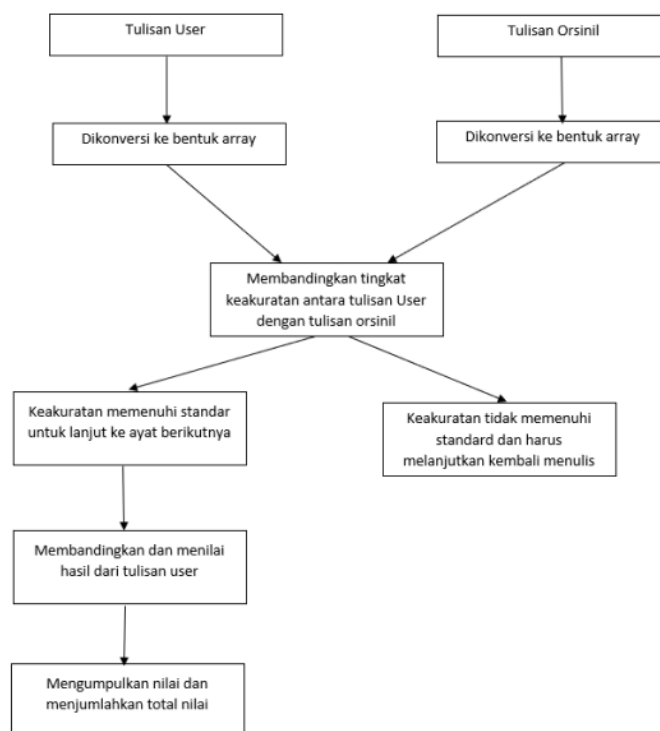
memenuhi tujuan utama dari pembuatan aplikasi, *affordance* untuk penggunaan tombol sesuai fungsinya, dan *feedback* untuk mengirim informasi ke *user* tentang apa yang dilakukan (Crearie, 2013).

Untuk rancangan dari arsitektur aplikasi Stylus-Q, sistem dibuat dengan menggunakan HTML5, JavaScript, CSS dan Resources yang berfungsi untuk memprogram struktur, tampilan, dan logika, beserta interaksi yang terjadi didalam sistem aplikasi. Setelah itu PhoneGap akan menkonverter dan mengkompiler modul dan masuk ke *server*. Untuk modul siswa yang akan menggunakan *stylus* atau sejenis pena yang digunakan untuk layar sentuh, dan modul akan menampilkan data berupa Input, Graphics dan Services. Data yang masuk kategori Input adalah ketika *user* mencoba menulis menggunakan Stylus-Q dan mengirimnya ke *server*, lalu data yang masuk kategori Graphics adalah ketika sistem menampilkan ayat yang akan ditulis beserta hasil tulisan *user*, dan data yang masuk kategori Services adalah semua fitur yang digunakan dalam sistem. Untuk modul guru, modul akan membentuk PhoneGap Native API dengan Plug-Ins berupa media, *network*, *file*, *notification*, *storage*, dan *custom plug-ins*. Untuk keperluan komunikasi dengan siswa, terdapat API (Application Programming Interface) berupa Whatsapp dan Telegram di modul guru. Gambar 3.2 dibawah ini akan mendeskripsikan arsitektur dari sistem ini.



**Gambar 3.2** Arsitektur Sistem

Untuk menunjukkan bagaimana proses yang terjadi dalam perencanaan sistem. Tahap pertama yang sistem lakukan yaitu. baik tulisan dari *user* dan tulisan orsinil dikonvert kedalam bentuk *array*. Tulisan orsinil tersebut berupa ayat yang bersumber dari mushaf Usmani, dan tampilan tulisan berupa *Khat Naskhi* dengan menggunakan *font Traditional Arabic*. Kemudian, sistem akan menggabungkan dan membandingkan tingkat keakuratan *array* dari *user* dengan tulisan orsinil. Perbandingan yang berupa banyak kekuratan penulisan *user* dengan tulisan orsinil akan menentukan sistem untuk mengambil keputusan ke tahap berikutnya apakah dapat lanjut ke ayat berikut atau tidak. Keputusan tersebut didasarkan dengan penilaian berupa tingkat keakuratan tulisan *user* dengan tulisan orsinil. Terakhir, skor didapat dari total nilai keseluruhan dibagi jumlah ayat yang dikerjakan. Berikut gambar 3.3 untuk proses dalam sistem.



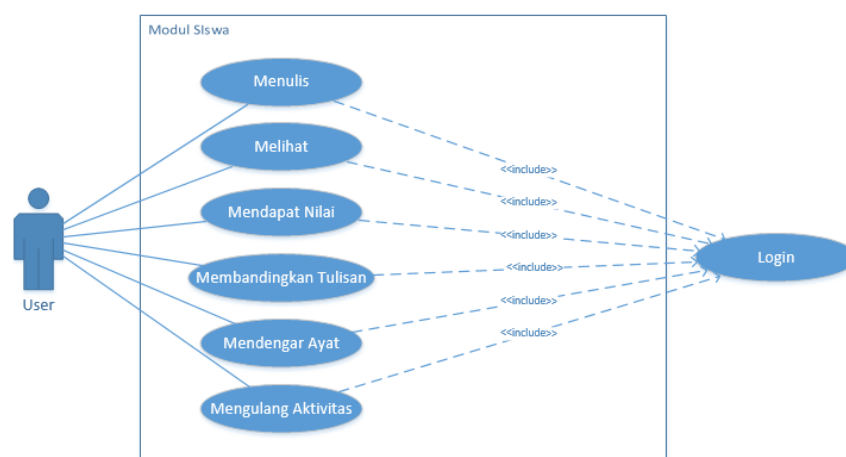
**Gambar 3.3** Proses Sistem

### 3.4. Perancangan Sistem

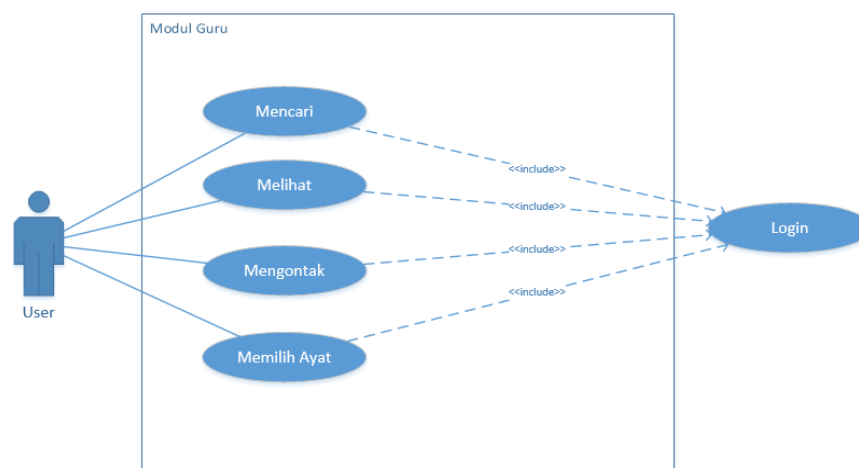
#### 3.4.1. Use Case Diagram

Gambar 3.4 menunjukkan bahwa sistem untuk modul siswa aplikasi mempunyai aktor berupa *user* atau siswa. Siswa dapat menulis, melihat dan

mendengar ayat, membandingkan tulisan, mendapat nilai tulisan, dan mengulang aktivitas sebanyak *user* inginkan. Sebelum dapat mengakses semua itu, *user* terlebih dahulu harus melakukan *login*. Sedangkan untuk gambar 3.5 menunjukkan sistem untuk modul guru yang mempunyai aktor berupa guru. Setelah *login*, guru dapat mencari data mata pelajaran, melihat daftar data siswa, mengontak siswa, dan memilih ayat yang ingin dilihat.



**Gambar 3.4** Use Case Modul Siswa

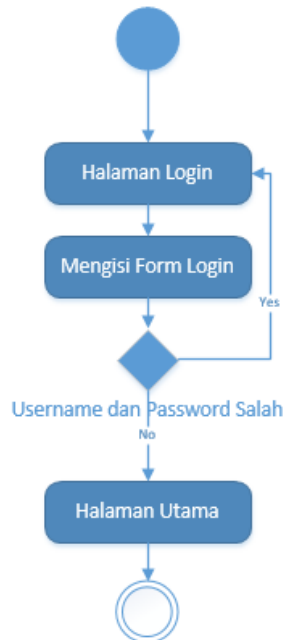


**Gambar 3.5** Use Case Modul Guru

### 3.4.2. Activity Diagram Login

Gambar 3.6 menampilkan bagaimana aktivitas dalam sistem untuk terjadi. Pertama, sistem akan menampilkan halaman yang terdapat *form* untuk diisi berupa *username* dan *password*. Jika data yang dimasukkan salah,

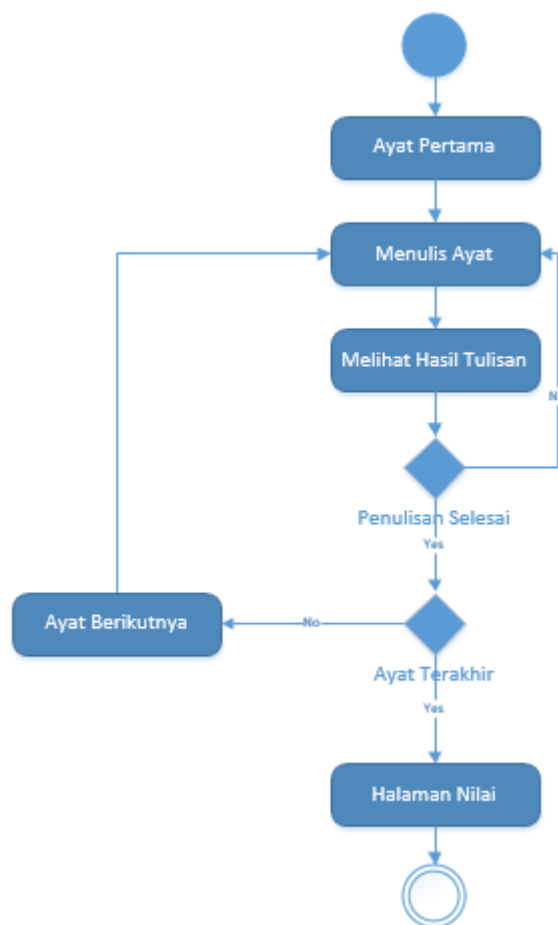
maka sistem akan tetap di halaman *login*. Jika data yang dimasukkan benar, maka sistem akan menuju halaman utama.



**Gambar 3.6** *Activity Login*

### 3.4.3. Activity Diagram Menulis

Gambar 3.7 menampilkan bagaimana aktivitas dalam sistem untuk fitur penulisan terjadi. Pertama, sistem akan menampilkan halaman utama berupa gambar untuk ayat pertama pada surah. Setelah itu, aktivitas menulis dapat dilakukan dan dapat dilihat hasilnya. Jika penulisan belum selesai, maka sistem akan kembali ke aktivitas menulis dan jika penulisan sudah selesai, maka sistem akan memastikan apakah ayat tersebut merupakan ayat terakhir pada surah atau bukan. Jika ayat tersebut bukan ayat terakhir, maka sistem akan mengarah pada ayat berikutnya dan jika ayat tersebut merupakan ayat terakhir maka sistem akan mengarah ke halaman hasil.



**Gambar 3.7** Activity Menulis

#### 3.4.4. Activity Diagram Perbandingan

Gambar 3.8 menampilkan bagaimana aktivitas dalam sistem untuk fitur perbandingan terjadi. Pertama, sistem akan menampilkan ayat asli, kemudian pengguna akan melakukan aktivitas menulis. Setelah itu, saat tombol hasil diklik, sistem akan membandingkan hasil aktivitas menulis dengan ayat asli. Jika perbandingan tersebut menghasilkan berupa aktivitas menulis belum selesai, maka sistem akan kembali kembali ke pekerjaan penulisan sebelumnya dan jika perbandingan tersebut menghasilkan aktivitas penulisan selesai, maka sistem akan lanjut ke ayat berikut.

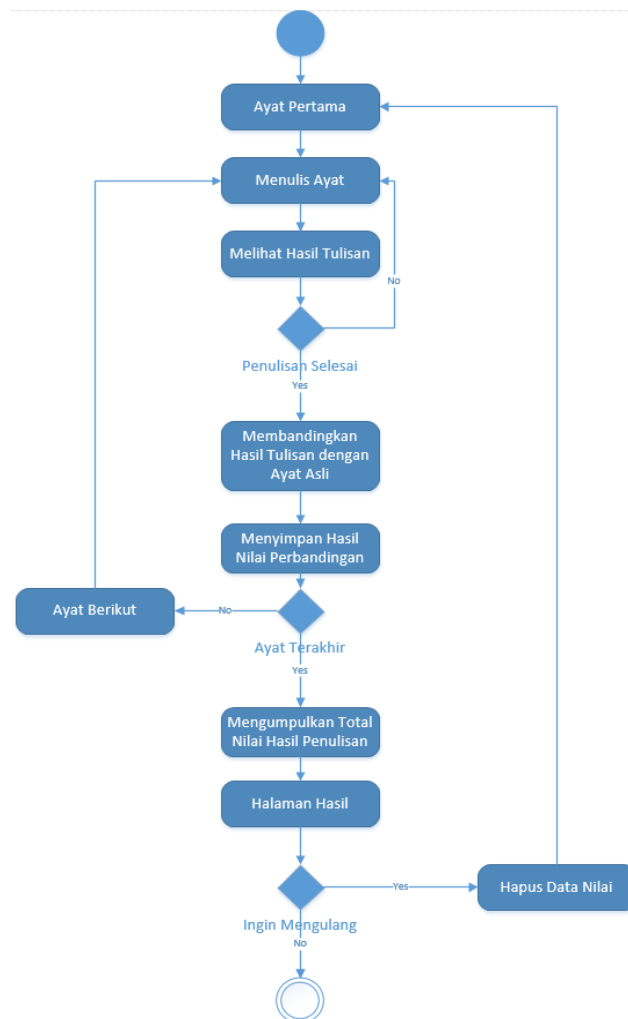




**Gambar 3.8** Activity Perbandingan

### 3.4.5. Activity Diagram Hasil Penulisan

Gambar 3.9 menampilkan bagaimana aktivitas dalam sistem untuk fitur hasil penulisan terjadi. Pertama, sistem akan menampilkan ayat pertama, disini *user* akan melakukan aktivitas menulis ayat dan melihat hasil penulisan *user*. Jika *user* belum selesai menulis ayat maka *user* akan kembali untuk melanjutkan menulis ayat dan jika penulisan sudah selesai, *user* akan menuju ayat berikut dan kembali melakukan aktivitas menulis, tetapi jika penulisan sudah selesai dan ayat tersebut merupakan ayat terakhir, maka aplikasi akan menampilkan total skor dari hasil penulisan yang dilakukan. Jika *user* ingin mengulang aktivitas menulis maka aplikasi akan menghapus data berupa nilai yang tadi didapatkan dan kembali menulis dari ayat pertama.



**Gambar 3.9** Activity Hasil Penulisan

### 3.4.6. Class Diagram

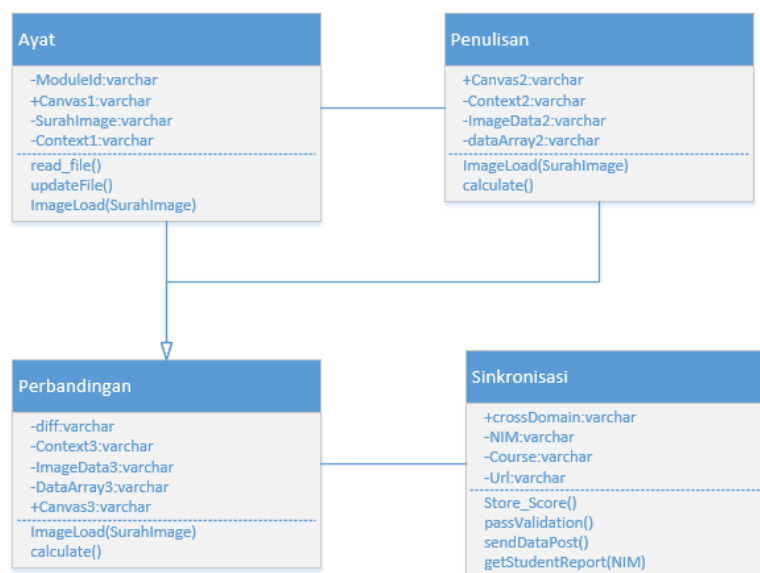
Pengembangan sistem ini mempunyai 4 bagian kelas yang utama yaitu kelas Ayat, Penulisan, Perbandingan, dan Sinkronisasi. Pada gambar 3.10 menunjukkan diagram kelas dari sistem ini. Adapun penjelasan dari diagram kelas yaitu:

1. Kelas Ayat. Kelas ini berfungsi untuk menyimpan seluruh gambar ayat dalam surah yang nantinya akan digunakan. Atribut dari kelas ini berupa "Canvas1" yang berjenis *public*, artinya kelas lain dapat membaca atribut tersebut. Sedangkan untuk relasi antar kelas, kelas Ayat berasosiasi dengan kelas Penulisan yang artinya hubungan antara *object* dapat digunakan oleh *object* yang lain tetapi tidak bersifat *owner* atau kelas tetap bisa dijalankan

walaupun *object* tersebut tidak ada. Kelas ini juga memiliki relasi *inheritance* dengan kelas Perbandingan yang artinya ada atribut dari kelas Ayat yang akan diturunkan ke kelas Perbandingan untuk membuat sistem berfungsi seperti seharusnya.

2. Kelas Penulisan. Kelas ini berfungsi untuk melakukan aktivitas menulis ayat. Atribut dari kelas ini berupa “Canvas2” yang berjenis *public*, artinya kelas lain dapat membaca atribut tersebut. Sedangkan untuk relasi antar kelas, kelas Penulisan berasosiasi dengan kelas Ayat yang artinya hubungan antara *object* dapat digunakan oleh *object* yang lain tetapi tidak bersifat *owner* atau kelas tetap bisa dijalankan walaupun *object* tersebut tidak ada. Kelas ini juga memiliki relasi *inheritance* dengan kelas Perbandingan yang artinya ada atribut dari kelas Penulisan yang akan diturunkan ke kelas Perbandingan untuk membuat sistem berfungsi seperti seharusnya.
3. Kelas Perbandingan. Kelas ini berfungsi untuk membandingkan hasil penulisan dengan dengan ayat asli atau dalam hal ini, membandingkan data dari kelas Ayat dengan kelas Penulisan. Atribut dari kelas ini berupa “Canvas3” yang berjenis *public*, artinya kelas lain dapat membaca atribut tersebut. Sedangkan untuk relasi antar kelas, Sedangkan untuk relasi antar kelas, kelas Perbandingan berasosiasi dengan kelas Sinkronisasi yang artinya hubungan antara *object* dapat digunakan oleh *object* yang lain tetapi tidak bersifat *owner* atau kelas tetap bisa dijalankan walaupun *object* tersebut tidak ada. Kelas ini juga merupakan *inheritance* dari kelas Ayat dan kelas Penulisan yang artinya ada atribut dari kelas tersebut yang akan diturunkan ke kelas Perbandingan untuk membuat sistem berfungsi seperti seharusnya.
4. Kelas Sinkronisasi. Kelas ini berfungsi untuk *restore* data hasil penulisan *user* agar bisa diakses oleh *server*. Atribut dari kelas ini berupa “crossDomain” yang berjenis *public*, artinya kelas lain dapat membaca atribut tersebut. Sedangkan untuk relasi antar kelas, Sedangkan untuk relasi antar kelas, kelas Sinkronisasi berasosiasi dengan kelas

Perbandingan yang artinya hubungan antara *object* dapat digunakan oleh *object* yang lain tetapi tidak bersifat *owner* atau kelas tetap bisa dijalankan walaupun *object* tersebut tidak ada.



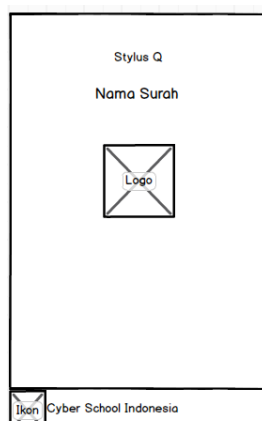
**Gambar 3.10** *Class Diagram*

### 3.4.7. Rancangan *Interface* (Antarmuka)

Rancangan untuk antarmuka merupakan bagian penting dalam proses desain sistem karena berfungsi untuk mendeskripsikan sistem kepada *user* dan membantu *programmer* untuk membuat sistem terutama dalam hal tampilan dan fungsi dari konten yang terdapat dalam sistem. Berikut adalah rancangan antarmuka untuk aplikasi Stylus-Q.

#### 3.4.1.1. *Home Page*

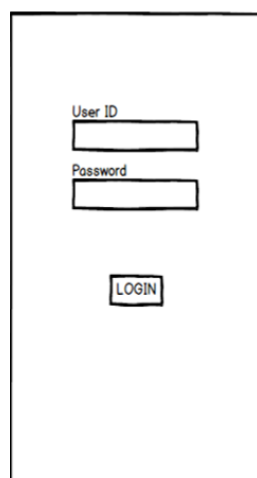
Gambar 3.11 merupakan rancangan untuk halaman awal untuk mengenalkan aplikasi yang dirancang yang terdiri dari nama aplikasi, nama surah yang akan dikerjakan, dan logo. Untuk masuk ke halaman berikutnya *user* cukup mengklik logo pada layar.



**Gambar 3.11** *Home Page*

#### 3.4.1.2. Halaman *Login*

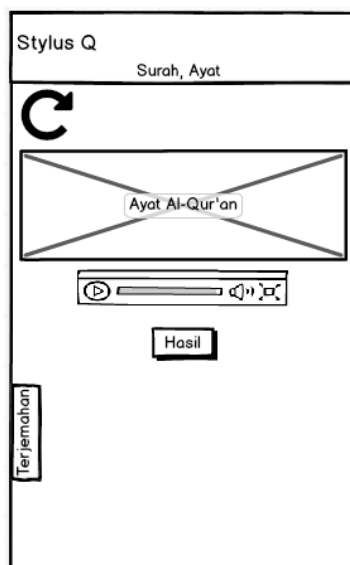
Gambar 3.12 merupakan rancangan untuk halaman yang berfungsi sebagai proses *authentication* pengguna yang hanya bisa jika sudah diregistrasi.



**Gambar 3.12** Halaman *Login*

#### 3.4.1.3. Halaman Utama

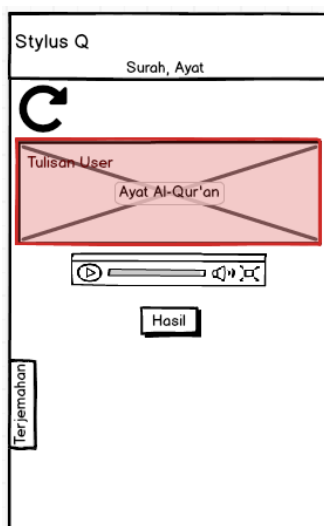
Gambar 3.13 merupakan rancangan untuk halaman utama untuk melakukan aktivitas menulis yang terdiri dari tombol *refresh* untuk mengulang penulisan, gambar yang menampilkan ayat yang akan ditulis, *playback* mendengarkan ayat, tombol untuk melihat hasil tulisan yang dibuat, dan tombol *slide menu* untuk melihat terjemahan ayat.



**Gambar 3.13** Halaman Utama

#### 3.4.1.4. Menulis Ayat

Gambar 3.14 merupakan rancangan ketika pengguna melakukan aktivitas menulis. aktivitas tersebut dilakukan secara *follow-the-line* atau cukup mengikuti garis pada ayat yang ditampilkan. Ketika menulis, tulisan pengguna akan berwarna berwarna merah.

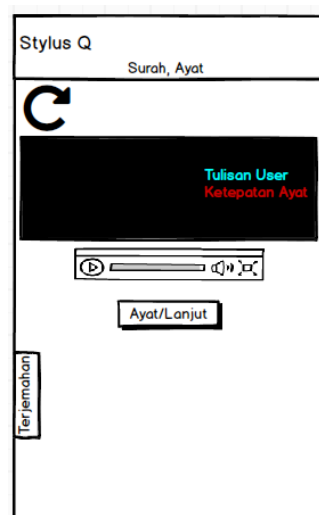


**Gambar 3.14** Menulis Ayat

#### 3.4.1.5. Melihat Hasil Tulisan

Gambar 3.15 merupakan rancangan ketika pengguna menekan tombol “Hasil” untuk melihat hasil tulisan pengguna. Di layar akan

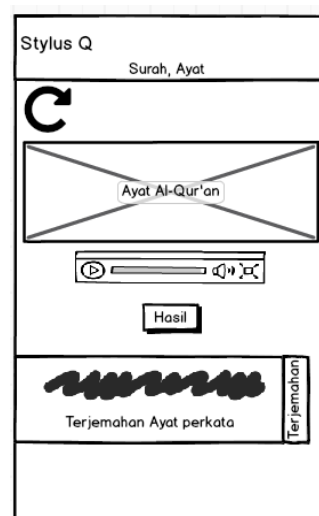
ditampilkan tulisan pengguna dengan warna *cyan* dan ketepatan penulisan pengguna dengan ayat asli berwarna merah. Serta tombol dengan fungsi untuk lanjut ke ayat berikut atau tetap lanjut menulis yang ditentukan dengan tingkat penulisan pengguna.



**Gambar 3.15** Melihat Hasil Penulisan

#### 3.4.1.6. Melihat Terjemahan

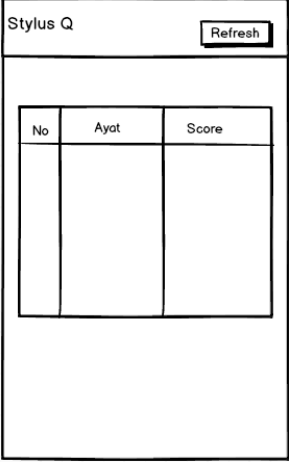
Gambar 3.16 merupakan rancangan ketika pengguna ingin melihat terjemahan pada ayat yang ingin dikerjakan berupa tombol “Terjemahan”. Tombol ini berupa *slide menu* dan terjemahan yang ditampilkan berupa terjemahan ayat perkata.



**Gambar 3.16** Melihat Terjemahan

#### 3.4.1.7. Halaman Nilai

Gambar 3.17 merupakan rancangan ketika pengguna sudah selesai mengerjakan seluruh ayat pada surah. Halaman ini terdiri dari tabel yang berisi nomor, daftar ayat, dan nilai yang didapat dari hasil penulisan dibandingkan dengan ketepatan penulisan. Terdapat juga tombol “*Refresh*” jika pengguna ingin menghapus data hasil penulisan terkini atau mengulangi pembelajaran dari awal.



The image shows a mobile application interface titled "Stylus Q". At the top right, there is a "Refresh" button. Below the title bar is a table with three columns: "No", "Ayat", and "Score". The table is currently empty, with only the header row visible. The "Ayat" column is significantly wider than the other two.

No	Ayat	Score
----	------	-------

**Gambar 3.17** Halaman Nilai