BAB III

METODE DAN PERANCANGAN APLIKASI

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut.

1) Studi Literatur

   Tahap ini merupakan tahap untuk mencari penelitian dan sumber yang berkaitan untuk dipelajari. Penelitian atau sumber dapat berupa media cetak, buku atau dapat juga berupa media elektronik, seperti artikel dari internet.

2) Perancangan dan Pengembangan

   Melakukan perancangan aplikasi *Spaced Repetition Software* dengan mengimplementasikan algoritma SuperMemo 2 ke dalam kode program berbasis iOS sesuai dengan tujuan dan batasan yang sudah ditentukan sebelumnya.

3) Implementasi

   Mengimplementasikan algoritma dan rancangan yang sudah dibuat sebelumnya ke dalam kode program untuk membangun *Spaced Repetition Software* berbasis iOS.

4) Penulisan laporan

   Penulisan laporan merupakan tahap dimana pendokumentasian seluruh tahap yang telah dilakukan dan disusun ke dalam bentuk laporan.

3.2 Analisis Algoritma

SM-2 merupakan sebuah algoritma yang dapat menghitung besarnya interval (jarak pengujian) pada sebuah *Spaced Repetition Software*. Algoritma ini menggunakan tiga variabel yang akan berubah-ubah yang menjadi acuan agar hasil interval pada saat pengujian berbeda-beda.

Yang pertama adalah EF, atau dapat disebut juga *easiness factor*, merupakan sebuah variabel yang digunakan oleh SM-2 untuk mengetahui tingkat kesulitan sebuah data yang akan diuji. Masing-masing dari data yang terdapat pada sebuah Spaced Repetition Software akan memiliki sebuah EF dan akan berubah seiring pembelajaran. Jarak dari nilai EF berkisar antara 1,0 hingga 2,5. Semakin kecil nilai EF, maka data tersebut dianggap sulit atau belum dikuasai oleh pengguna (Wozniak, 1998).

Variabel kedua yang digunakan adalah q atau *Quality*, yang berarti kualitas dari sebuah jawaban pada saat penggunaan *Spaced Repetition Software*. *Quality* sendiri memiliki jarak dari 0 sampai dengan 5 (Wozniak, 1998). Nilai dari q akan berpengaruh pada perubahan dari nilai EF itu sendiri, semakin kecil kualitas dari jawaban maka penurunan nilai EF akan semakin signifikan.

Variabel yang terakhir adalah interval pada pengujian. Variabel ini menyimpan berapa banyak pengulangan yang telah dilakukan oleh pengguna dan pengguna berhasil mengingat data yang di tampilkan. Ketika terjadi kegagalan, maka nilai interval akan kembali menjadi 1 dan proses akan berulang kembali.

Sebagai contoh, penggunaan aplikasi algoritma SM-2 menggunakan Rumus 2.1,

$$EF' = 2.5 + (0.1 - (5 - 3) * (0.08 + (5 - 3) * 0.02))$$
$$EF' = 2.5 + (0.1 - 2 * (0.08 + 2 * 0.02))$$
$$EF' = 2.5 + (0.1 - 2 * (0.12))$$
$$EF' = 2.36$$

Perhitungan diatas memperlihatkan contoh perhitungan dengan variabel q = 3 akan menurunkan nilai EF sebanyak 0.14 sehingga nilai akhir EF menjadi 2.36. Untuk algoritma SM-2 nilai q sebesar 3 akan memberikan penurunan nilai, sedangkan q dengan nilai 4 tidak akan mengubah nilai EF dan q sebesar 5 akan menambahkan nilai EF. Nilai EF yang menjadi variabel yang digunakan untuk memberikan nilai interval yang bervariasi dan hasilnya dipengaruhi dari performa *user* yang menggunakan.

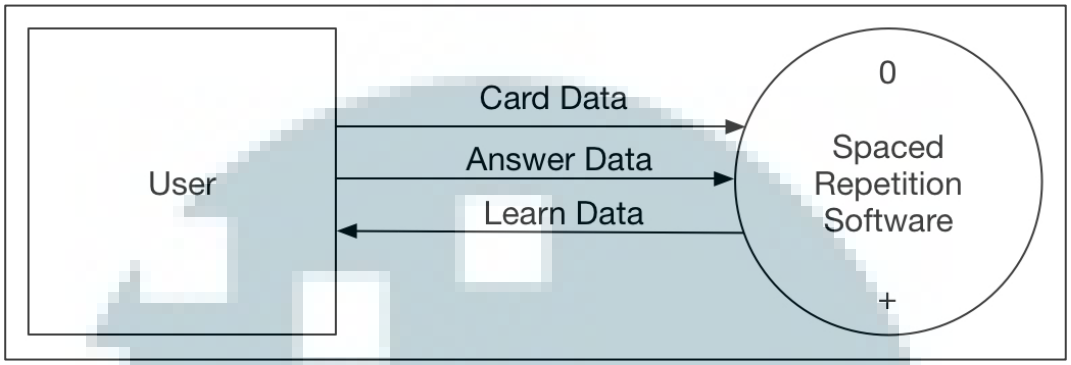
Sebagai contoh perhitungan menggunakan Rumus 2.2 dengan EF sebesar 2.36 dan nilai n sebesar 3 maka,

$$EF' = 2.36$$
$$I(3) = I(2) * EF$$
$$I(3) = 6 * 2.36$$
$$I(3) = 14.16$$

Selanjutnya untuk interval untuk iterasi pertama dan kedua memiliki besaran yang tetap, I(1) = 1 dan I(2) = 6, untuk selanjutnya digunakan Rumus 2.2 seperti pada contoh diatas. Hasil perhitungan akan berbeda-beda antara satu sama lain dikarenakan nilai EF dapat berubah-ubah. Interval yang dihasilkan akan berupa berapa jarak *inter-repetition* dengan besaran hari dan dilakukan pembulatan, sebagai contoh hasil dari perhitungan interval diatas adalah 14 hari.

3.3 Perancangan Aplikasi

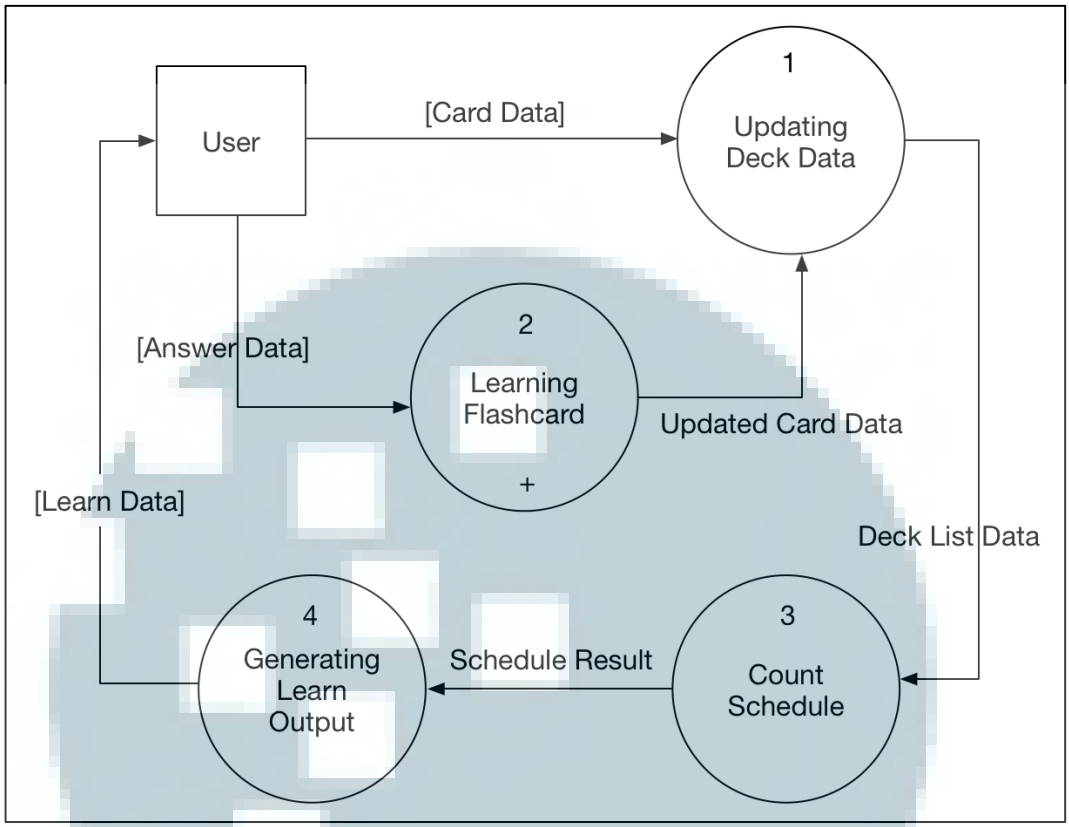
3.3.1 Data Flow Diagram



Gambar 3.1 Diagram Konteks Aplikasi

Gambar 3.1 merupakan *Data Flow Diagram* level 0 atau diagram konteks dari aplikasi. *User* dapat berinteraksi dengan program tersebut, antara lain dengan memberikan pilihan kartu yang akan digunakan untuk pengetesan dan dimasukan kedalam *deck*. Selain itu *user* juga mampu untuk menghilangkan kartu yang telah terdaftar pada *deck*.

Selain menambahkan atau menghilangkan, hal lain yang dapat dilakukan *user* adalah untuk meminta jadwal pengetesan *flashcard*. *User* akan disuguhkan dengan sebuah *flashcard* yang telah terdaftar pada *deck* yang telah sebelumnya *user* lakukan. Terakhir, *user* diharuskan untuk memberikan inputan jawaban untuk *flashcard* yang telah disuguhkan. Selanjutnya, inputan tersebut akan digunakan untuk perhitungan interval *flashcard* agar diketahui jadwal pengetesan selanjutnya.



Gambar 3.2 *Data Flow Diagram* Sub-Proses dari Proses SRS

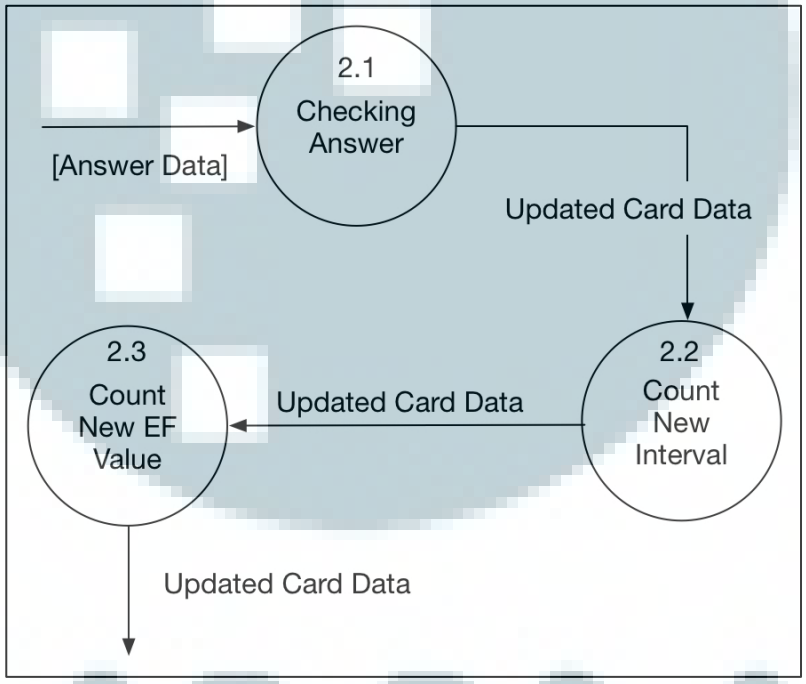
Gambar 3.2 merupakan *Data Flow Diagram* (DFD) level 1, yang sekaligus merupakan sub-proses dari *Spaced Repetition Software* yang berada pada level 0 atau diagram konteks. Terdapat tiga sub-proses dari aplikasi yang dibangun.

Sub-proses yang pertama merupakan proses *Updating Deck Data*. Proses ini akan menerima input berupa *flashcard* pilihan dari *user* dimana *user* dapat menambahkan *flashcard* yang ingin dipelajari ataupun menghapus *flashcard* yang sudah tidak diperlukan dari *deck.*

Sub-proses yang kedua memiliki nama *Learn Flashcard.* Proses ini menerima data-data *flashcard* yang berada dalam *deck user*. Pada proses ini dilakukan perhitungan SM-2 yang akan memberikan jadwal pengetesan yang harus dilakukan *user* pada kartu tersebut.

Sub-proses yang ketiga merupakan *count schedule.* Pada proses ini dilakukan perbandingan antara tanggal perhitungan dengan tanggal pengecekan. Jarak dari tanggal tersebut akan menjadi indikator apakah *flashcard* tersebut perlu untuk dilakukan pengetesan atau tidak pada proses selanjutnya.

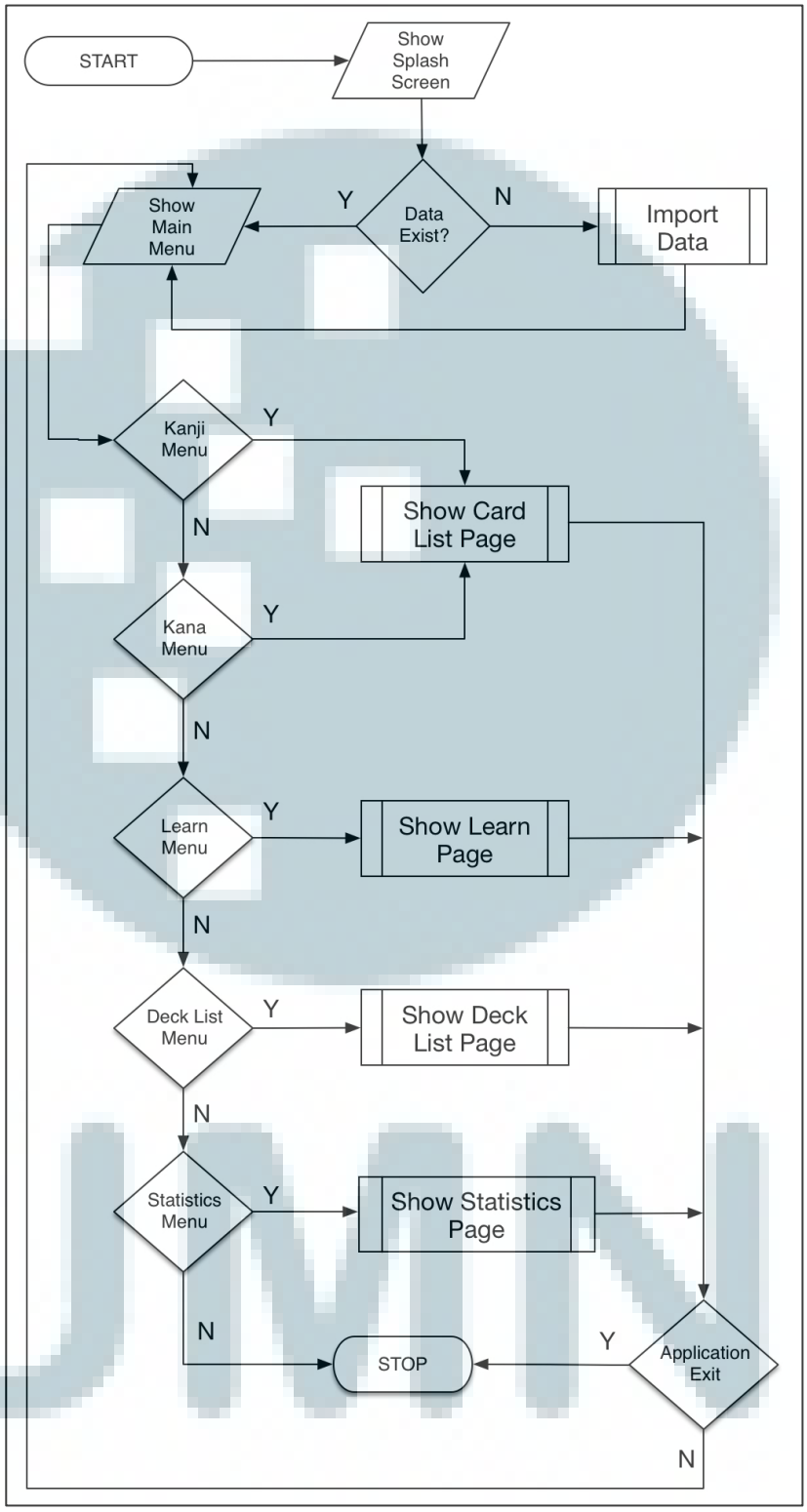
Proses selanjutnya yang bernama *generating learn output* bertugas untuk memberikan *user*. *Flashcard* yang sudah mencapai waktu yang ditentukan agar dilakukan pengetesan. Setelah proses perhitungan selesai, maka hasil akan diberikan kepada *user.*



Gambar 3.3 *Data Flow Diagram* Sub Proses *Learning Flashcard*

Gambar 3.3 menggambarkan sub-proses *Learning Flashcard* dimana alur perhitungan diawali dengan *user* mengirimkan data jawaban yang akan digunakan dalam perhitungan *Interval* dan EF. Selanjutnya, data kartu akan diperbarui menggunakan perhitungan-perhitungan yang digunakan dalam algoritma SM-2 dan dimasukan kedalam *deck list*.

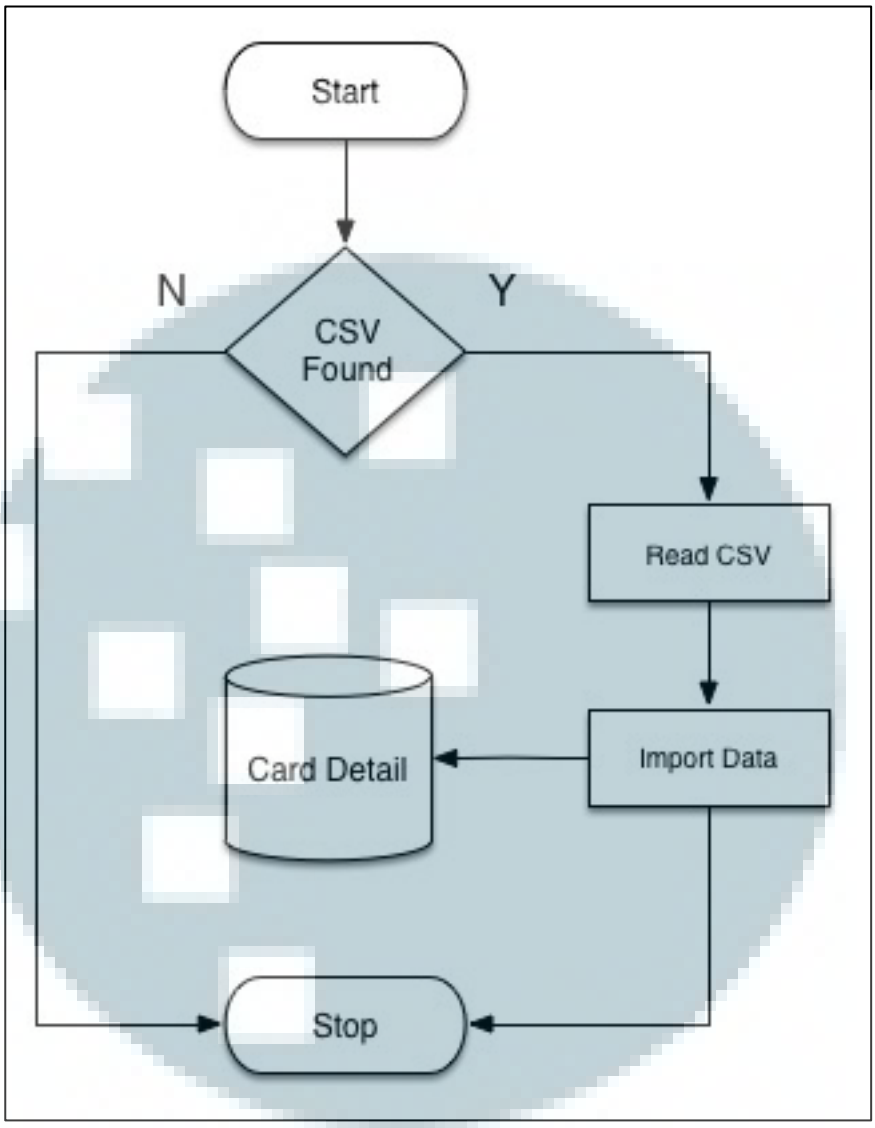
3.3.2 Flow Chart Aplikasi



Gambar 3.4 *Flow Chart* Aplikasi

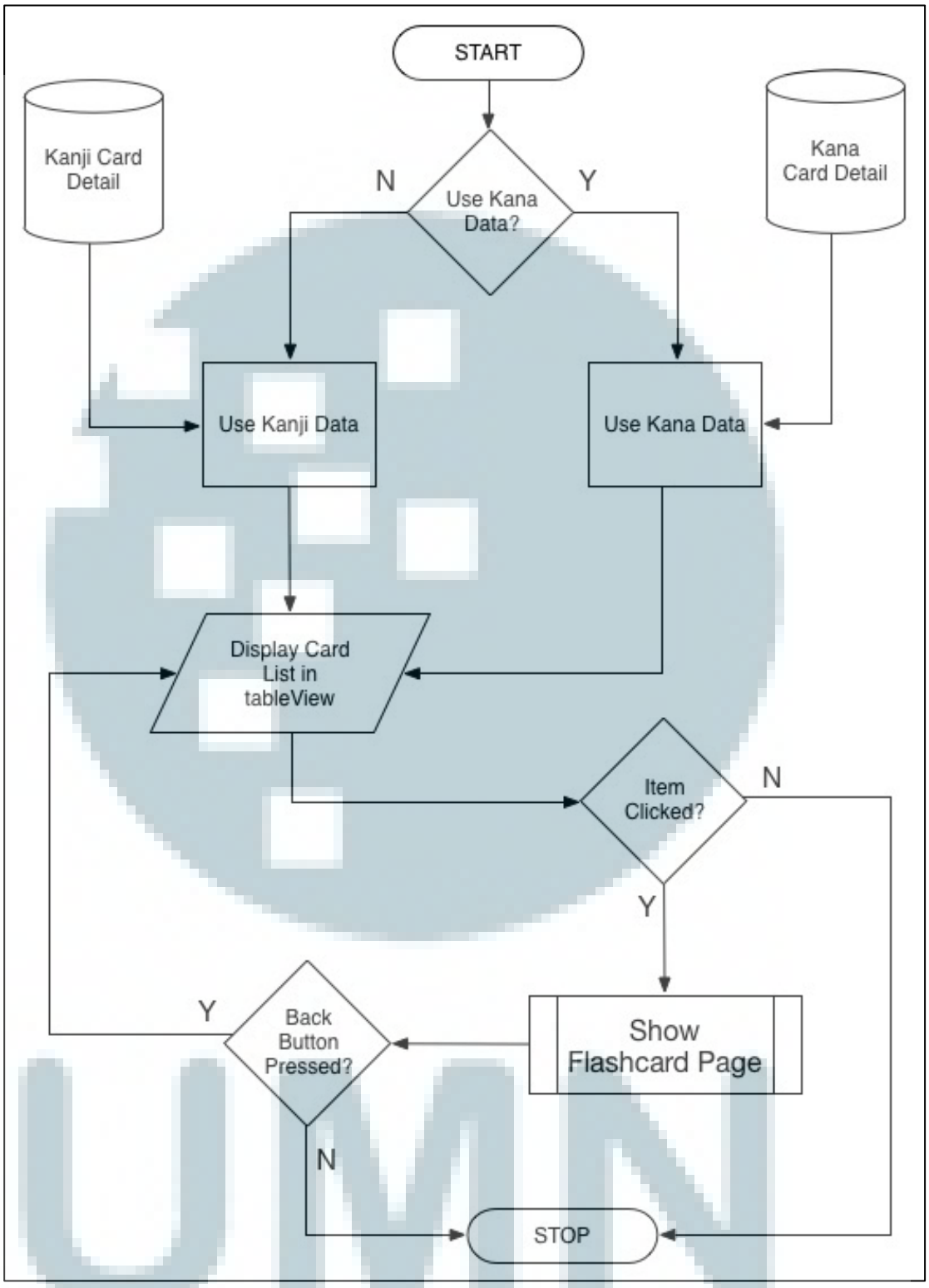
Gambar 3.4 merupakan *flow chart* umum untuk aplikasi SRS yang dibangun. Proses dimulai dengan menampilkan *splash screen* dan dilanjutkan dengan pengecekan terhadap data program. Jika data belum ditemukan maka akan dilakukan sub proses *import data*.

Setelah pengecekan dilakukan tampilan *main menu* akan ditampilkan. Pada menu ini, *user* akan memilih satu dari lima tombol menu yang tesedia. Jika *user* menekan tombol Kanji atau Kana, maka *user* akan memasuki sub-proses *Show Card List Page.* Perbedaan terletak pada jenis data yang ditampilkan. Selanjutnya jika *user* menekan tombol *Learn,* maka *user* akan memasuki sub proses *Show Learn Page* dimana sistem utama dari aplikasi, yaitu *Spaced Repetition* ini akan dijalankan. Tombol *Deck List* akan membawa *user* pada sub-proses *Show Deck List Page* dimana *Deck Management* akan dilakukan. Terakhir tombol *Statistics* akan membawa *user* kepada sub proses *Show Statistics Page* dimana *user* akan disuguhkan informasi yang bersangkutan dengan *deck* yang dimilikinya.



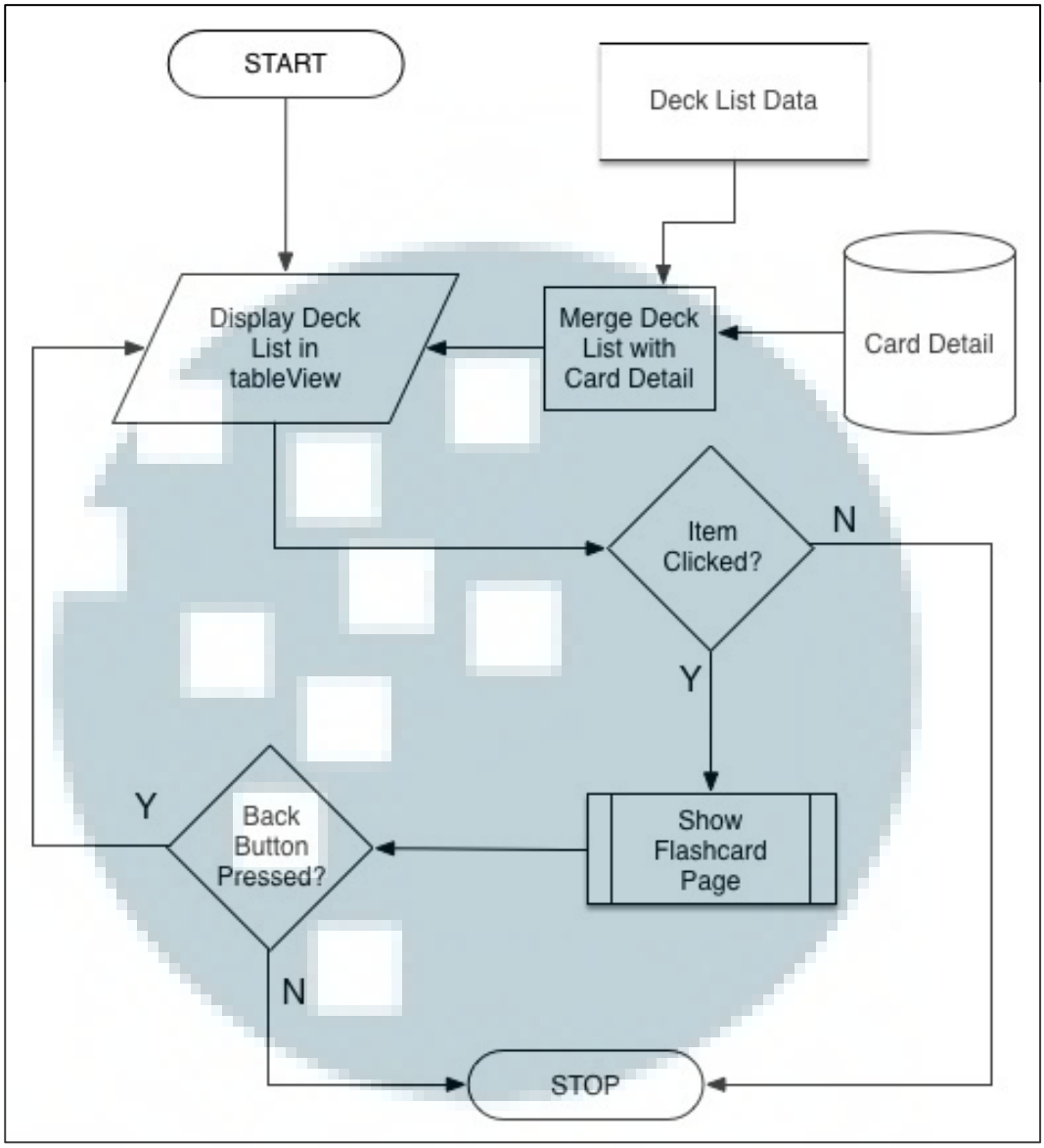
Gambar 3.5 *Flow Chart Import Data*

Gambar 3.5 merupakan *flow chart* detail untuk sub-proses *Import Data* yang terdapat pada *flow chart* aplikasi. Pada proses ini dilakukan pengecekan terhadap *file* CSV yang berisi data kartu yang akan ditampilan pada *flashcard*. Jika ditemukan, maka *file* CSV akan dibaca dan proses *Import* akan dilakukan. Jika *file* CSV tidak ditemukan, maka proses *Import* tidak berjalan.



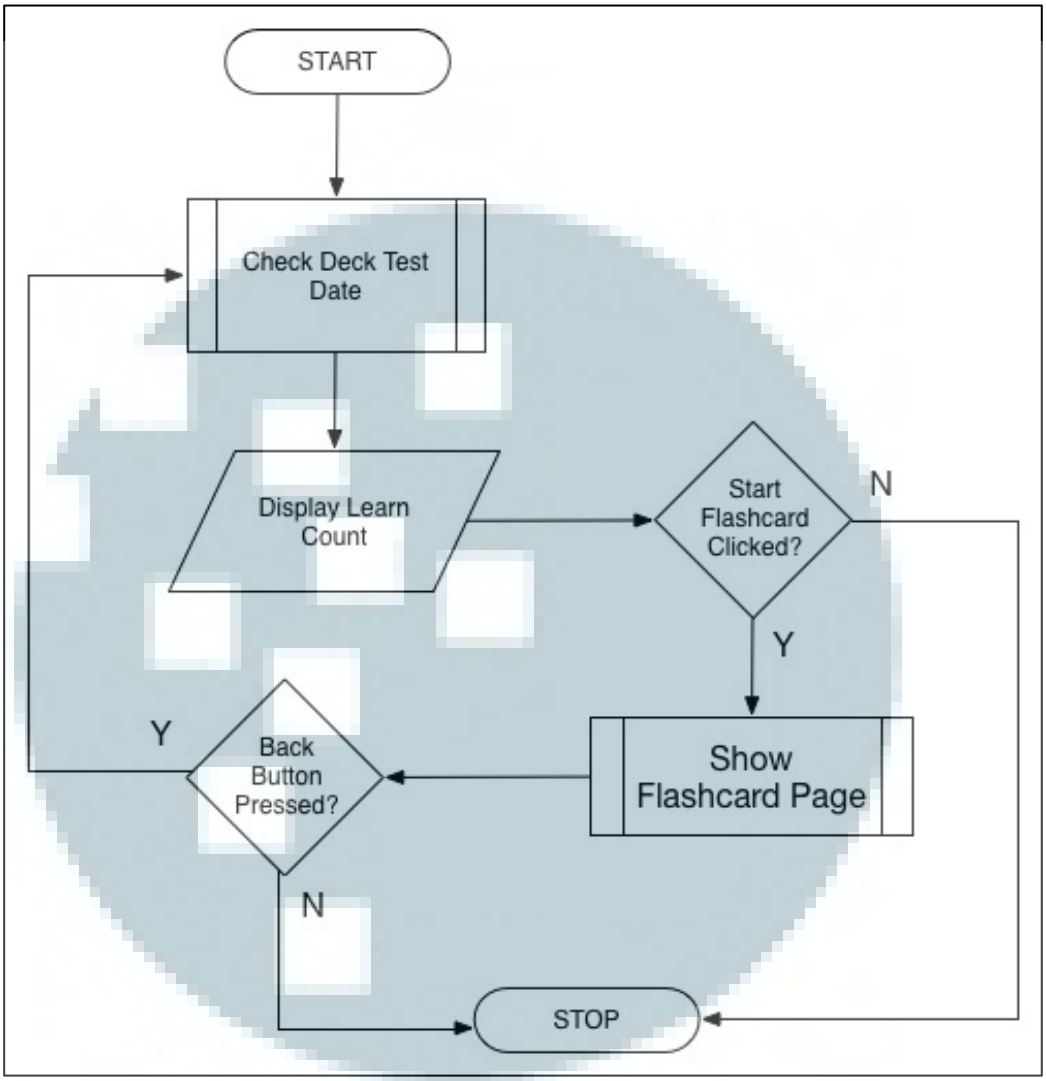
Gambar 3.6 *Flow Chart Show Card List Page*

Gambar 3.6 menggambarkan *flow chart* untuk sub-proses *Show Card List Page*. Proses dimulai dengan pengecekan terhadap data yang akan digunakan. Jika *user* masuk melalui menu Kana, maka data yang digunakan adalah data untuk seluruh Hiragana dan Katakana. Sedangkan jika masuk melalui menu Kanji maka data yang digunakan adalah data Kanji. Setelah pengecekan data dilakukan dan data telah diambil, data tersbut akan ditampilkan dalam sebuah tabel yang terdapat pada aplikasi. Ketika *user* melakukan interaksi *click* pada salah satu *item* di tabel, maka *user* akan dibawa ke sub-proses *Show Flashcard Page,* dimana *user* dapat menambahkan *item* yang dipilih untuk di masukan ke dalam *deck* maupun melihat detail dari kartu tersebut.



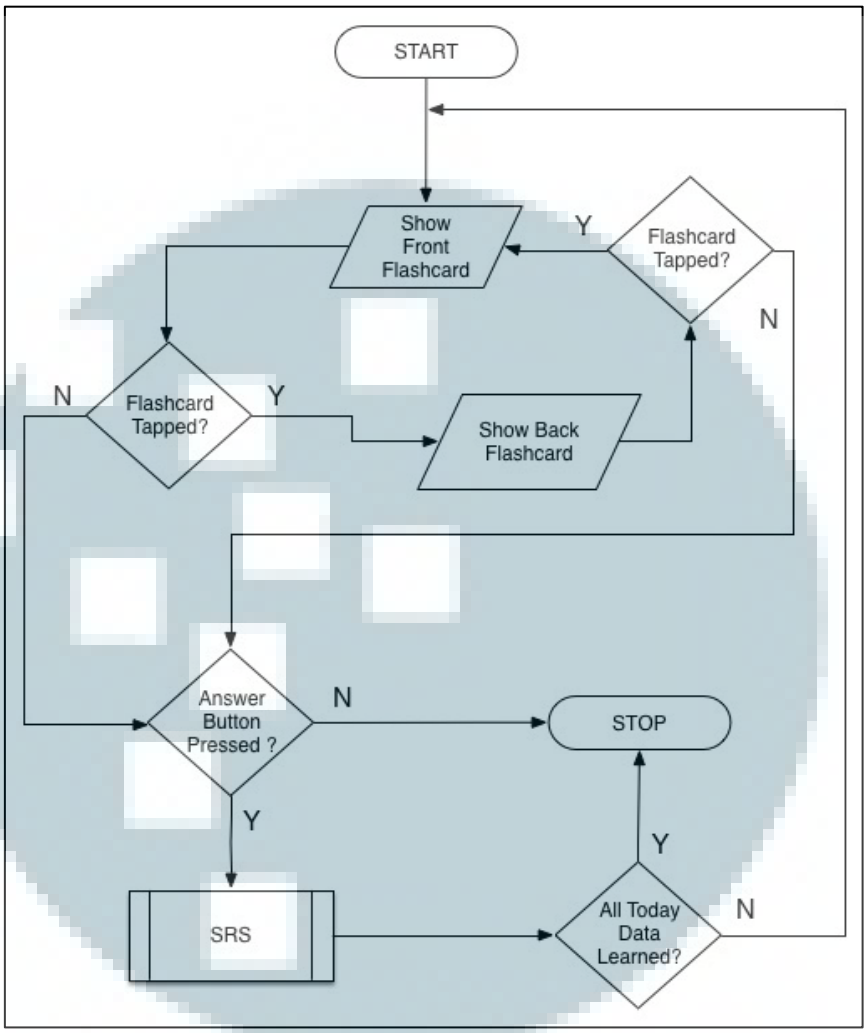
Gambar 3.7 *Flow Chart Show Deck List*

Gambar 3.7 menggambarkan *flow chart* dari sub-proses *Show Deck List Page*. Proses dimulai dengan *user* mendapat tampilan tabel yang berisikan data yang merupakan *list* dari kartu di *deck* yang *user* telah pilih sebelumnya. Pada menu ini *user* dapat melakukan penghapusan data dan juga dapat melihat detail dari kartu dengan melakukan *click* pada salah satu *item* di tabel yang akan membawa *user* ke sub-proses *Show Flashcard Page*.



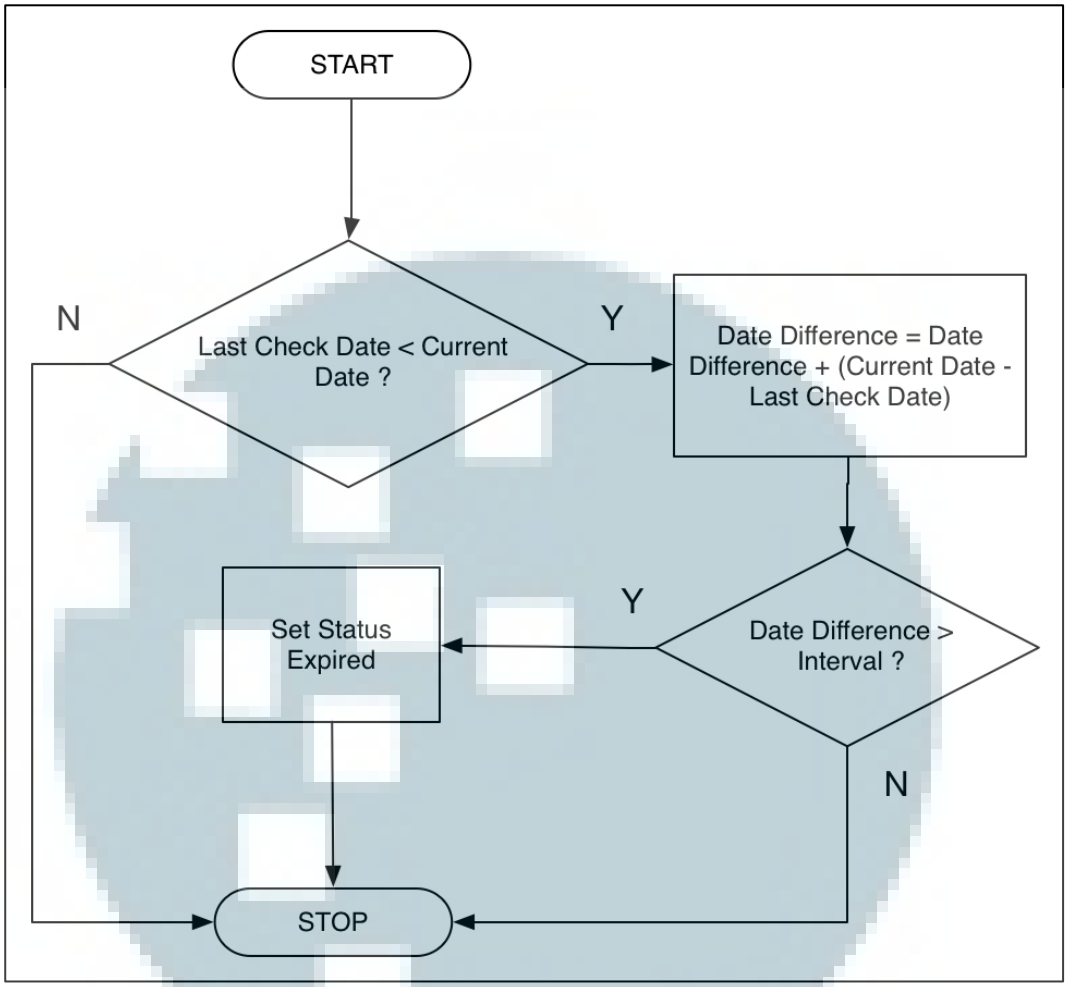
Gambar 3.8 *Flow Chart Show Learn Page*

Gambar 3.8 memperlihatkan *flow chart* dari sub-proses *Show Learn Page*. Proses dimulai dengan sistem akan memasuki sub-proses *Check Deck Test Date*. Setelah selesai pengecekan, maka jumlah *flashcard* yang harus dipelajari akan ditampilkan. *User* dapat menekan tombol *Start Flashcard* yang akan menampilkan sub-proses *Show Flashcard Page* dimana proses pembelajaran pada *flashcard* yang terjadwal akan dilakukan.



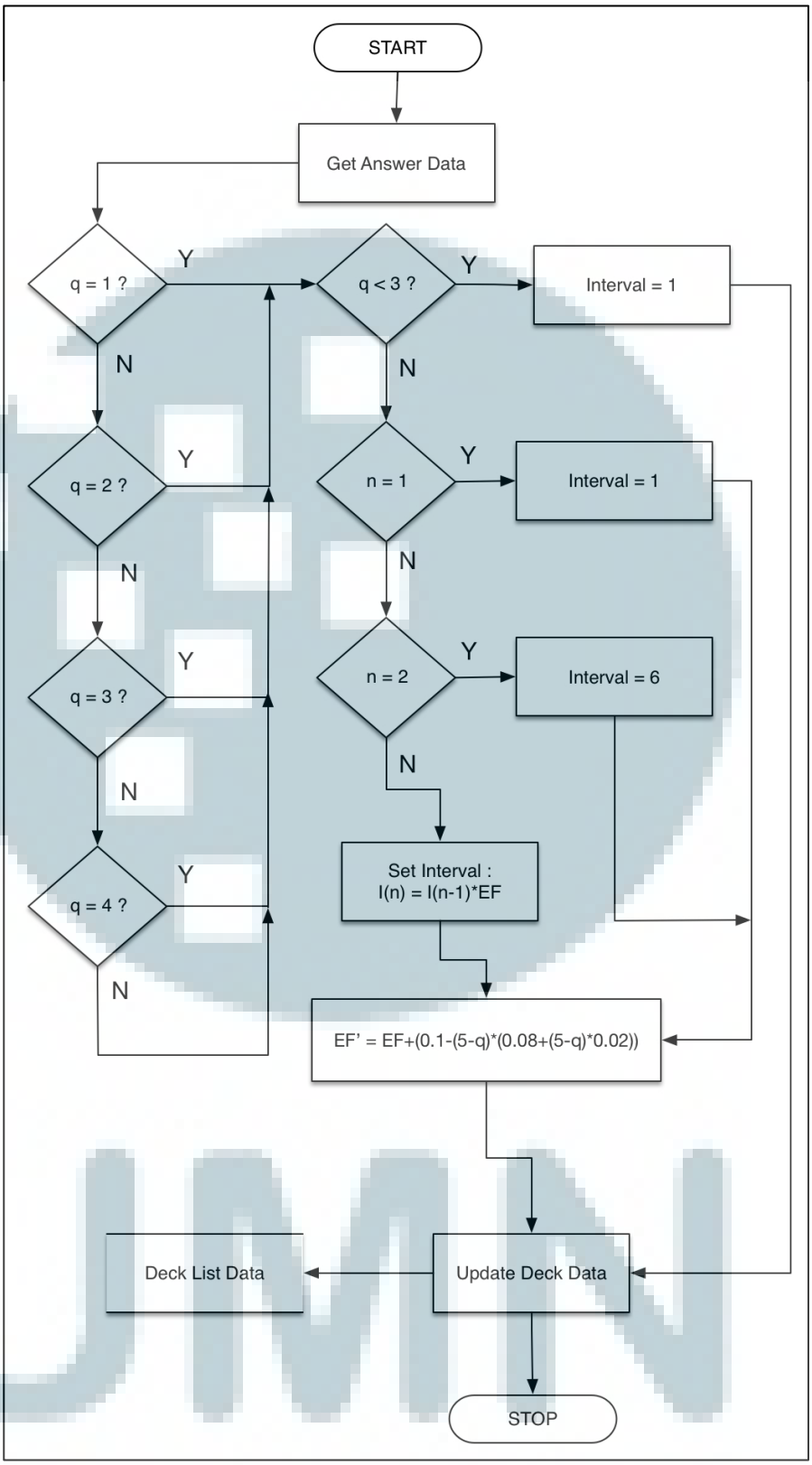
Gambar 3.9 *Flow Chart Show Flashcard Page*

Gambar 3.9 memperlihatkan *flow chart* dari sub proses *Show Flashcard Page*. Proses dimulai dengan menampilkan bagian depan dari *flashcard* yang dipilih dan ketika *user* melakukan *tap* terhadap *flashcard*, maka bagian belakang *flashcard* akan ditampilkan. Dalam mode SRS, user dapat menekan tombol jawaban yang akan membawa *user* ke sub proses SRS.



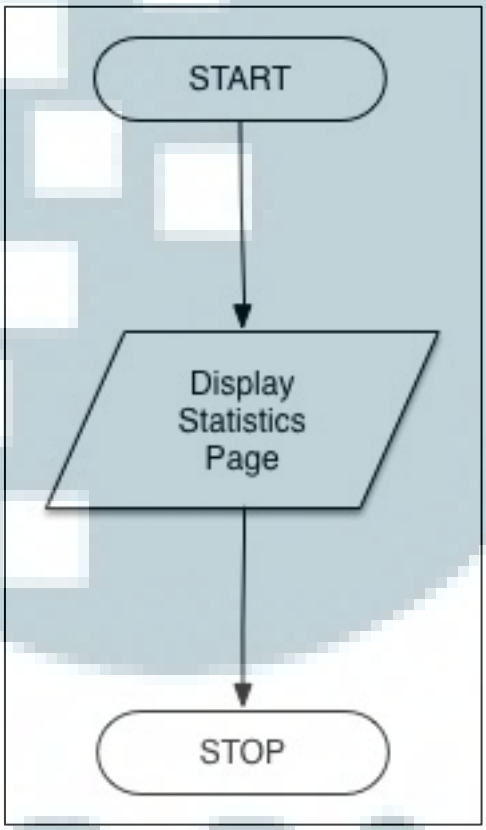
Gambar 3.10 *Flow Chart Check Deck Test Date*

Gambar 3.10 merupakan *flow chart* dari sub-proses *check deck test date*, dimulai dari pengecekan terhadap tanggal terkahir pengecekan dilakukan. Ketika belum dilakukan pengecekan maka akan terjadi perubahan status pada kartu yang terdapat dalam deck yaitu penambahan nilai variabel *Date Difference* yang digunakan untuk melakukan pengecekan terhadap *Interval* yang merupakan hasil perhitungan dari algoritma SM-2 yang telah dilakukan pada saat *user* melakukan pembelajaran. Jika nilai offset melebihi hasil interval maka status dari *flashcard* akan berubah menjadi *expired*.



Gambar 3.11 *Flow Chart SRS*

Gambar 3.11 merupakan *flow chart* dari sub-proses SRS. Dimulai dengan pengecekan jawaban yang telah *user* pilih sebelumnya. Setelah *user* menjawab, maka akan dilakukan pengecekan jika q < 3, maka interval akan dikembalikan menjadi satu tanpa mengubah EF dan *deck data* akan di *update*, jika tidak maka akan dilakukan perhitungan interval dan perubahan pada nilai EF dan *deck data* akan diperbarui.

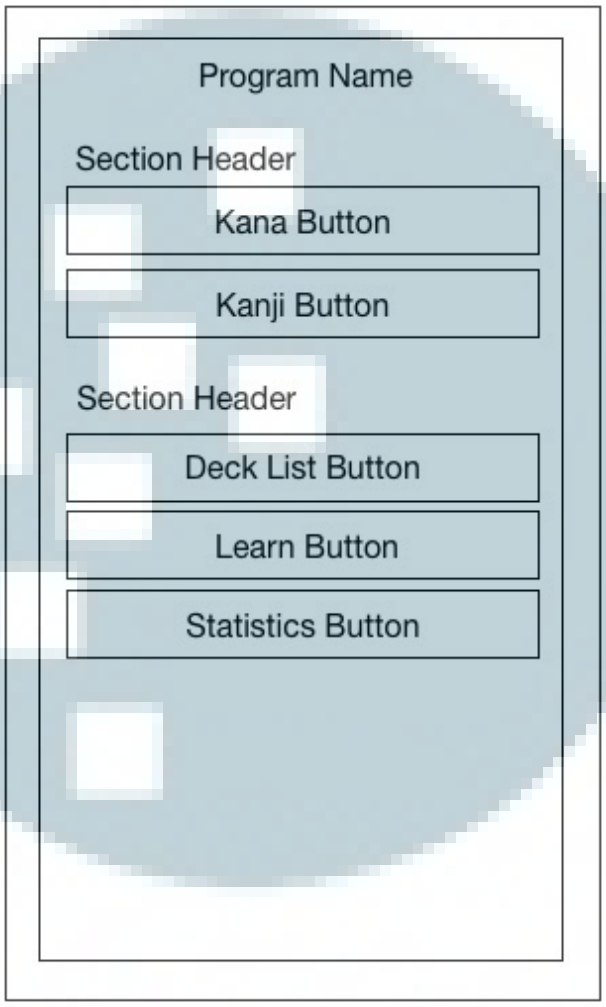


Gambar 3.12 *Flow Chart Show Statistics Page*

Gambar 3.12 merupakan *flow chart* yang menggambarkan sub proses *Show Statistics Page*. Dimana *user* dapat melihat *track record* dari penggunaan SRS dan melihat perkiraan jadwal pembelajaran dalam jangka waktu satu minggu.

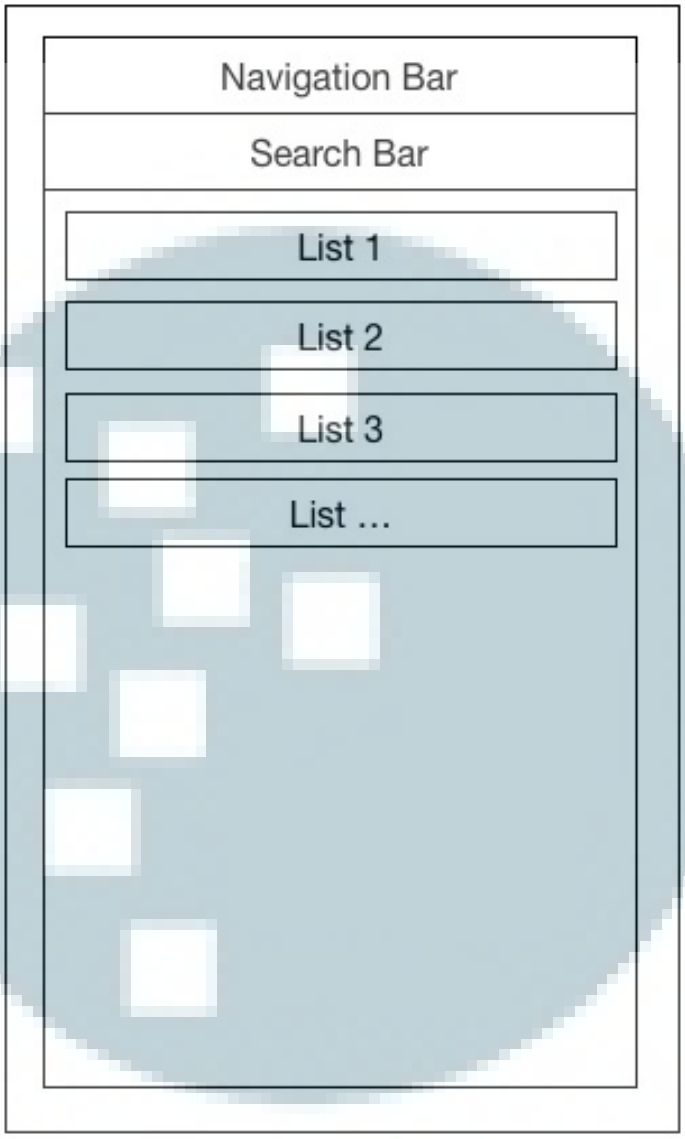
3.4 Perancangan Antar Muka Sistem

Berikut ini merupakan rancangan antar muka sistem untuk aplikasi yang dibangun.



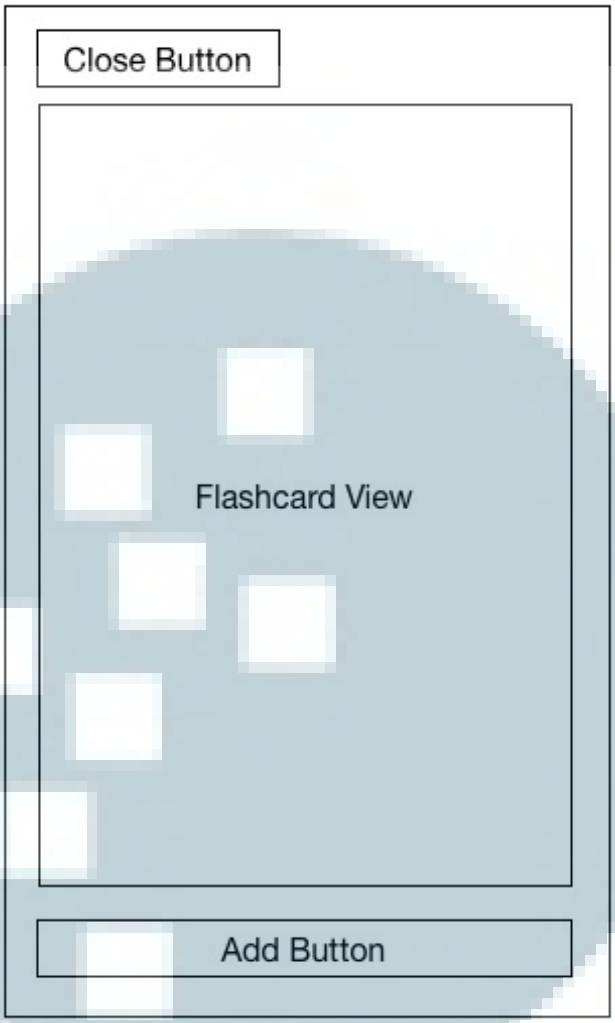
Gambar 3.13 Rancangan antar muka *Main Menu*

Gambar 3.13 merupakan rancangan antar muka awal saat seorang *user* pertama mengakses aplikasi. Pada bagian awal ini, *user* akan disuguhkan dengan lima buah tombol menu yang akan mengantar *user* ke *page* selanjutnya. Menu tersebut antara lain Kana, Kanji, *Deck List*, *Learn* dan *Statistics*.



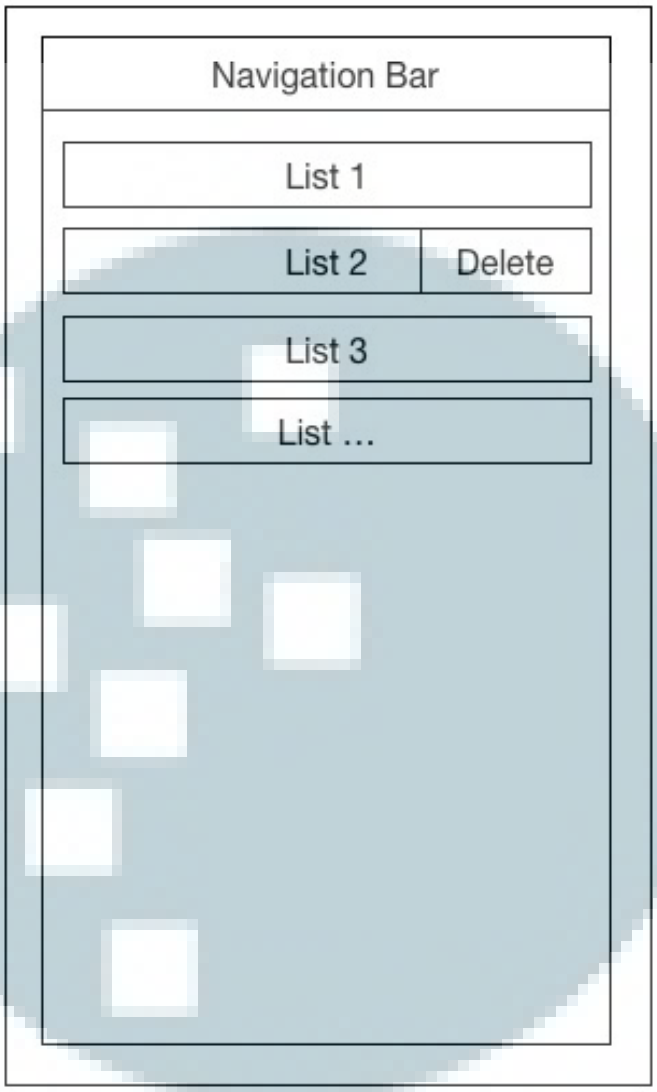
Gambar 3.14 Rancangan antar muka *Card List*

Gambar 3.14 merupakan rancangan antar muka setelah *user* menekan tombol kanji atau kana. Terdapat tabel yang berisi data kanji maupun kana yang ditampilkan kepada *user*. Pada menu ini juga terdapat search bar yang berfungsi untuk memfilter data yang ditampilkan di tabel. Melakukan *click* pada list item akan menampilkan *flashcard add page* kepada *user*.



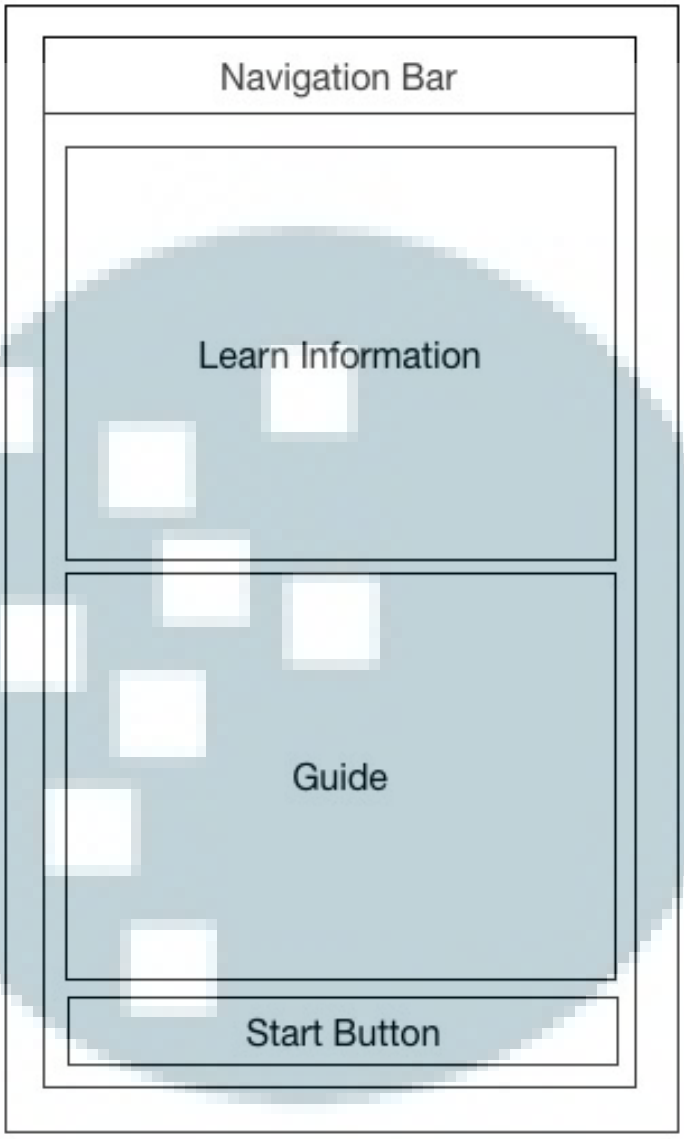
Gambar 3.15 Rancangan antar muka *flashcard add*

Gambar 3.15 merupakan rancangan antar muka setelah *user* melakukan *click* pada salah satu *item* di tabel yang terdapat pada antar muka *Card List*. Disini *user* akan disuguhkan sebuah *flashcard* yang berisi data tentang *item* yang dipilih dan *user* juga dapat menekan tombol *Add* yang akan menambahkan *flashcard* tersebut ke dalam *deck* miliknya. Melakukan *single tap* pada *flashcard* akan memutar *flashcard* tersebut dan menampilkan data pada bagian belakang *flashcard*.



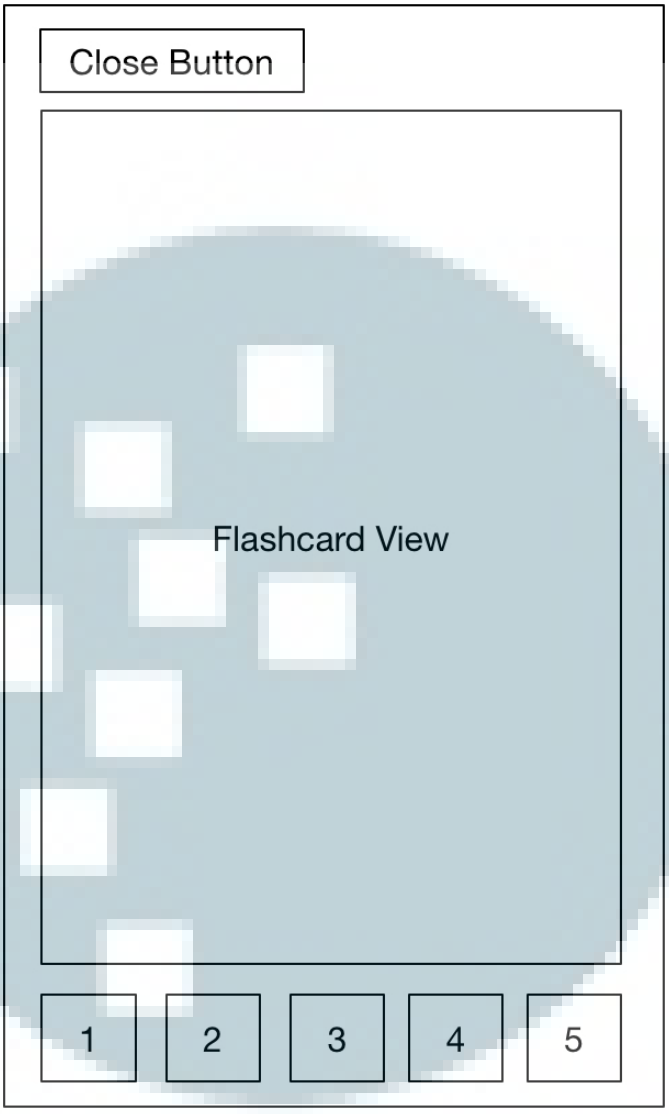
Gambar 3.16 Rancangan antar muka *deck list page*

Gambar 3.16 merupakan rancangan antar muka ketika *user* menekan tombol *deck list* pada *main menu*. Disini *user* dapat melihat *flashcard* apa saja yang telah *user* pilih untuk dipelajari. Selain itu, *user* juga dapat menghapus dengan melakukan *slide gesture* pada salah satu item dalam tabel dan menekan tombol Delete. *User* juga dapat melihat detail kartu dengan melakukan *click* pada item seperti halnya pada *card list page.*



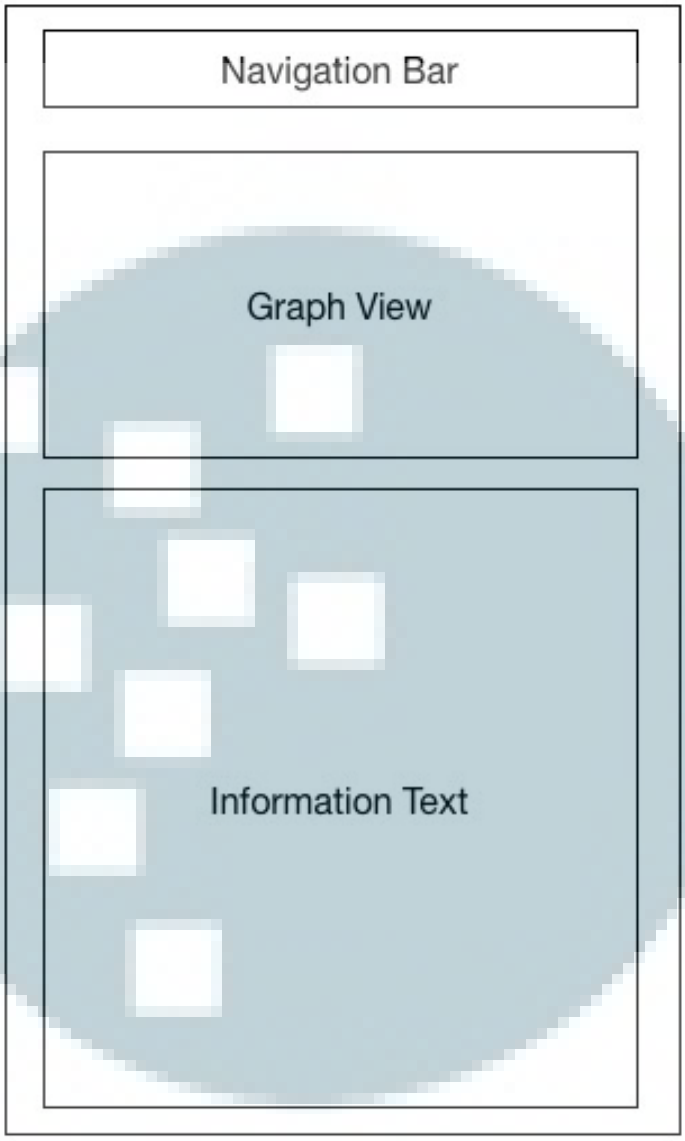
Gambar 3.17 Rancangan antar muka *Learn Page*

Gambar 3.17 merupakan rancangan antar muka ketika *user* menekan tombol *learn* pada *main menu*. Disini user akan diberikan informasi mengenai berapa banyak jumlah *flashcard* yang bisa dipelajari pada saat pengecekan. Terdapat sebuah section yang berisi paduan penggunaan. Dan pada bagian bawah terdapat tombol untuk memulai pembelajaran.



Gambar 3.18 Rancangan antar muka *flashcard test*

Gambar 3.18 merupakan rancangan antar muka untuk *flashcard* dengan mode pengetesan, Memiliki fungsi yang mirip seperti *flashcard add* dengan perbedaan pada fungsi dan tombol pada bagian bawah. Pada antar muka ini tombol menjadi lima buah yang berfungsi sebagai jawaban saat pembelajaran. Tombol yang tersedia antara lain adalah 1, 2, 3, 4 dan 5 (0 dihilangkan) yang merupakan nilai yang *user* berikan terhadap *flashcard* yang dipelajari.



Gambar 3.19 Rancangan antar muka *Statistics*

Gambar 3.19 merupakan rancangan untuk *statistics page* dimana *user* dapat melihat statistik pemakaian aplikasi ini seperti halnya jumlah kartu pada masing-masing status, serta informasi-informasi seperti jumlah kartu pada deck beserta jumlah *flashcard* untuk dipelajari selama 7 hari dari hari pengecekan.