



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB III

### METODOLOGI DAN PERANCANGAN SISTEM

#### 3.1 Metodologi Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut.

##### 1. Studi Literatur

Studi literatur merupakan tahapan untuk mempelajari referensi-referensi yang berkaitan dengan teori-teori yang akan menjadi dasar untuk perancangan dan pembangunan sistem ini. Teori-teori tersebut antara lain adalah bahasa mandarin, bahasa jepang, metode gamifikasi, *game design*, *game mechanics*, *game dynamics*, algoritma Knuth Shuffle, dan HMSAM.

##### 2. Perancangan Sistem

Sistem yang dibangun terdiri dari aplikasi *mobile* berbasis *mobile*, *application programming interface*, dan *content management system* (CMS) berbasis *web*. Seluruh data informasi mengenai pembelajaran bahasa asing tersimpan di dalam sebuah *database* yang dimiliki oleh *server*. Aplikasi *mobile* ditunjukkan kepada orang-orang yang ingin belajar bahasa mandarin ataupun bahasa jepang, setiap materi di dalam aplikasi didapat dari *server* setelah aplikasi mengirimkan *request* ke *server*. CMS ditunjukkan kepada *administrator* untuk mengelola materi-materi yang ada di dalam aplikasi.

##### 3. Pembuatan Sistem

Setelah rancangan arsitektur sistem dibuat, tahap ini dilanjutkan dengan pembuatan sistem pembelajaran bahasa asing, baik aplikasi *mobile*, API, dan CMS. Pembuatan sistem juga mencakup perancangan desain antarmuka pengguna atau *user interface*, *user experience*, dan penentuan fitur-fitur tambahan yang

dibutuhkan pada sistem. Pembuatan sistem dilakukan dengan menggunakan *framework* Ionic untuk *mobile* serta *framework* CodeIgniter untuk *backend web*.

#### 4. Pengujian Sistem

Pengujian sistem dilakukan dengan menggunakan metode Black Box Testing. Metode tersebut digunakan untuk memfokuskan pengujian terhadap fungsionalitas sistem secara keseluruhan. Pengujian dilakukan dengan mencoba setiap kemungkinan yang akan terjadi dari penggunaan sistem berdasarkan perancangan yang sudah dibuat.

#### 5. Evaluasi

Evaluasi sistem dilakukan dengan menggunakan metode studi lapangan, dimana akan dipilih sampel dengan metode *purposive sampling* yaitu orang yang sedang belajar atau ingin mempelajari bahasa mandarin ataupun bahasa jepang untuk mencoba menggunakan aplikasi *mobile*. Metode pengujian ini dilakukan agar dapat menghitung tingkat *behavioral intention to use* dan *immersion* dari sistem yang dibuat. Hasil evaluasi akan disimpulkan melalui penggunaan Likert Scale lima tingkat agar dapat mengukur kegunaan dari penggunaan sistem terhadap penggunaan metode gamifikasi terhadap pembelajaran bahasa asing.

### 3.2 Variabel Penelitian

Variabel terikat pada penelitian ini adalah tingkat kemudahan dari penggunaan sistem, kinerja penggunaan sistem, rasa ingin tahu yang dimiliki oleh pengguna, kesenangan yang didapatkan oleh pengguna, dan ukuran persepsi pengguna seakan pengguna berinteraksi langsung dengan sistem. Variabel bebas pada penelitian ini adalah komponen *game mechanics* yang digunakan dan

komponen *game dynamic* yang digunakan. Komponen *game mechanics* yang digunakan terdiri dari *points, levels, challenges, achievement, leaderboard*, dan *competitions*. Komponen *game dynamic* yang digunakan terdiri dari *reward, status, achievement, self-expression, competition*, dan *alturism*. Variabel penelitian akan diukur menggunakan Likert Scale lima tingkat dengan pembobotan beragam antar jawaban.

### **3.3 Teknik Pengumpulan Data**

Pada tahap pembuatan, data akan dikumpulkan dengan cara studi dokumen atau meneliti berbagai macam dokumen yang berguna untuk menentukan materi apa yang perlu dimasukkan ke dalam *database* dari sistem yang dibuat. Pada tahap pengujian, data dikumpulkan menggunakan metode studi lapangan dengan menyediakan aplikasi *mobile* ke pengguna dan menanyakan pengalaman dari pemakaian aplikasi tersebut dalam bentuk kuesioner. Alat yang akan digunakan untuk mengukur kuesioner tersebut adalah Likert Scale lima tingkat, lalu dihitung menggunakan model HMSAM.

### **3.4 Teknik Pengambilan Sampel**

Teknik pengambilan sampel yang digunakan dalam penelitian ini adalah *purposive sampling*. Teknik ini digunakan karena setiap sampel harus mewakili populasi dari kumpulan seseorang yang sedang atau ingin mempelajari bahasa asing secara umum. Jumlah sample yang diperlukan adalah antara tiga puluh sampai dengan lima ratus orang responden, semakin besar jumlah sampel yang diambil mendekati populasinya, maka peluang kesalahan generalisasi akan semakin kecil

(Sugiyono, 2012). Jumlah responden sebanyak tiga puluh responden berdasarkan teori yang dituliskan oleh Sugiyono (2012).

### 3.5 Analisis Game Design

*Game design* dilakukan berdasarkan poin-poin penting yang harus dibuat yang dijelaskan oleh Adams (2010). Poin yang dilihat adalah mendapatkan konsep, mendefinisikan pemain, dan menentukan peran dari pemain.

#### 1. Mendapatkan konsep

Di dalam sistem ini, pengguna mempunyai tujuan untuk mempelajari bahasa asing dan menyelesaikan pertanyaan-pertanyaan yang diberikan. Pengguna akan ditempatkan sebagai pelajar dari Indonesia yang sedang belajar di luar negeri. Pengguna akan bertemu dengan NPC (*Non-player Character*) yang akan mengajarkan materi-materi pembelajaran yang ada. Selain itu, pengguna juga dapat bertemu dengan NPC yang akan menguji kemampuan dari pengguna tersebut. Pengguna dapat menguji kemampuannya dengan pemain lain melalui pertarungan antara pengguna. Pertarungan antara pemain akan

#### 2. Mendefinisikan pemain

Pemain pada sistem ini merupakan siswa sekolah menengah atas, sekolah menengah kejuruan, atau mahasiswa universitas yang sedang mempelajari bahasa asing disekolahnya atau memiliki keinginan untuk belajar sendiri.

#### 3. Menentukan peran dari pemain

Pemain berperan sebagai seorang pelajar yang sedang belajar di luar negeri. Pemain pada aplikasi ini akan berinteraksi dengan NPC yang akan mengajarkan bahasa asing dan memberikan hadiah kepada pemain berupa *point* dan *achievement*.

Selain itu pemain akan berinteraksi dengan NPC yang akan menguji kemampuan yang dimiliki oleh pemain.

### 3.6 Analisis Game Mechanics

Komponen *game mechanics* yang digunakan pada penelitian ini adalah sebagai berikut.

#### 1. *Points*

Pengguna akan mendapatkan *point* setelah menyelesaikan sebuah materi, melakukan *challenges*, mendapatkan *achievement*, dan mengikuti kompetisi yang dilakukan antara pengguna dan pengguna lainnya. Pengguna akan mendapatkan 100 *point* saat menyelesaikan satu materi, maksimal 100 *point* saat melakukan *challenges*, dan maksimal 100 *point* saat mengikuti kompetisi yang dilakukan antara pengguna. *Point* yang didapatkan melalui *challenges* dan kompetisi ditentukan sesuai dengan nilai yang didapatkan oleh pengguna.

#### 2. *Levels*

Pengguna akan mendapatkan *level* sesuai dengan *point* yang telah pengguna kumpulkan. Semakin banyak *point* yang dikumpulkan, maka *level* dari pengguna akan semakin tinggi. Terdapat lima puluh *level* yang dapat diraih oleh pengguna dimulai dari level 1 hingga level 50, setiap *level* membutuhkan 100 *point* untuk naik ke-*level* selanjutnya. Setelah pengguna mencapai level 50, pengguna tidak akan naik ke level selanjutnya, tetapi poin yang dikumpulkan akan tetap tersimpan.

#### 3. *Challenges*

Pengguna dapat mengumpulkan *point* dengan cara melakukan *challenge*, *challenge* juga bertujuan menguji kemampuan dari pengguna. Pengguna yang

menyelesaikan *challenge* akan mendapatkan *point*. *Point* yang didapatkan oleh pengguna ditentukan sesuai dengan nilai yang didapatkan oleh pengguna saat mengerjakan *challenge*. Jumlah maksimal *point* yang bisa didapatkan oleh pengguna saat melakukan satu *challenge* adalah 100 *point*.

#### 4. *Achievement*

Pengguna dapat mengumpulkan *achievement* dengan cara menaikan *level* pengguna, menyelesaikan materi, atau memenangkan pertandingan. *Achievement* pada aplikasi ini dapat dilihat pada halaman profil pengguna. *Achievement* dapat dipilih oleh pengguna sebagai gambar profil pengguna.

#### 5. *Leaderboard*

Pengguna yang memiliki *level* tertinggi akan tertulis namanya pada *leaderboard*. Jumlah nama yang tertulis pada *leaderboard* ini akan dibatasi sebanyak sepuluh, sehingga pengguna yang ingin namanya berada pada *leaderboard* harus mendapatkan *level* lebih tinggi dari pengguna dengan peringkat terendah pada *leaderboard*.

#### 6. *Competitions*

Aplikasi yang dibuat memiliki menu papan peringkat. Penggunaan papan peringkat pada aplikasi memiliki tujuan yaitu setiap pengguna saling berkompetisi untuk mendapatkan peringkat tertinggi. Sepuluh pengguna dengan *level* tertinggi akan ditulis namanya pada papan peringkat.

Selain papan peringkat, kompetisi dalam aplikasi yang dibuat adalah pertarungan antar pengguna. Pertarungan antar pengguna terjadi saat ada satu pengguna yang memulai pertarungan dengan pengguna lain secara acak. Pengguna

yang memenangkan pertarungan mendapatkan *point* lebih banyak dari pengguna yang kalah.

### 3.7 Analisis Game Dynamic

Komponen *game dynamic* yang digunakan pada penelitian ini adalah sebagai berikut.

#### 1. *Reward*

*Reward* terdiri dari *point* dan *achievement* yang didapatkan melalui *challenges* dan *competitions*. *Reward* dalam bentuk *point* akan diubah menjadi *level*, sedangkan *reward* dalam bentuk *achievement* akan ditampilkan dalam bentuk gambar pengguna.

#### 2. *Status*

Status didapatkan dengan cara menaikan *level* pengguna. *Status* akan terlihat pada halaman profil masing-masing pengguna. Terdapat enam *status* yang dapat diraih oleh pengguna dalam aplikasi.

Tabel 3.1 Tabel Daftar Status

No.	Level	Status
1.	1-10	<i>Newbie</i>
2.	11-20	<i>Junior Learner</i>
3.	21-30	<i>Senior Learner</i>
4.	31-40	<i>Semi-Pro Learner</i>
5.	41-49	<i>Pro Learner</i>
6.	50	<i>Legend</i>

#### 3. *Achievement*

*Achievement* pada aplikasi berbentuk gambar pengguna yang dapat dipilih oleh pengguna. Gambar yang dapat dipilih oleh pengguna hanya *achievement* yang telah didapatkan oleh pengguna. Terdapat delapan belas belas *achievement* yang bisa didapatkan oleh pengguna. Enam *achievement* didapatkan dari pelajaran

bahasa mandarin, enam *achievement* didapatkan dari pelajaran bahasa jepang, dan enam *achievement* didapatkan dari *status*.

Tabel 3.2 Tabel Daftar Achievement

No.	Gambar	Nama	Keterangan
1.		<i>China explorers</i>	<i>Achievement</i> didapatkan saat pengguna menyelesaikan materi <i>Greetings and number</i> pada pelajaran bahasa mandarin.
2.		<i>Chinese friend</i>	<i>Achievement</i> didapatkan saat pengguna menyelesaikan materi <i>Family and school/university</i> pada pelajaran bahasa mandarin.
3.		<i>Weather in Chinese</i>	<i>Achievement</i> didapatkan saat pengguna menyelesaikan materi <i>Time and weather</i> pada pelajaran bahasa mandarin.
4.		<i>Work in Chinese</i>	<i>Achievement</i> didapatkan saat pengguna menyelesaikan materi <i>Work and hobbies</i> pada pelajaran bahasa mandarin.
5.		<i>Shop and Dine in Chinese</i>	<i>Achievement</i> didapatkan saat pengguna menyelesaikan materi <i>Shopping and food</i> pada pelajaran bahasa mandarin.
6.		<i>Complete Mandarin language</i>	<i>Achievement</i> didapatkan saat pengguna menyelesaikan materi <i>Traffic and directions</i> pada pelajaran bahasa mandarin.

Tabel 3.2 Tabel Daftar Achievement (lanjutan)

7.		<i>Japan explorers</i>	Achievement didapatkan saat pengguna menyelesaikan materi <i>Greetings and number</i> pada pelajaran bahasa jepang.
8.		<i>Japanese friend</i>	Achievement didapatkan saat pengguna menyelesaikan materi <i>Family and school/university</i> pada pelajaran bahasa jepang.
9.		<i>Weather in Japanese</i>	Achievement didapatkan saat pengguna menyelesaikan materi <i>Time and weather</i> pada pelajaran bahasa jepang.
10.		<i>Work in Japanese</i>	Achievement didapatkan saat pengguna menyelesaikan materi <i>Work and hobbies</i> pada pelajaran bahasa jepang.
11.		<i>Shop and Dine in Japanese</i>	Achievement didapatkan saat pengguna menyelesaikan materi <i>Shopping and food</i> pada pelajaran bahasa jepang.
12.		<i>Complete Japanese language</i>	Achievement didapatkan saat pengguna menyelesaikan materi <i>Traffic and directions</i> pada pelajaran bahasa jepang.
13.		<i>Initiated to learn foreign language</i>	Achievement didapatkan saat pengguna mendapatkan <i>status Newbie</i> pertama kali.

Tabel 3.2 Tabel Daftar Achievement (lanjutan)

14.		<i>Getting used to learn foreign language</i>	<i>Achievement</i> didapatkan saat pengguna mendapatkan <i>status Junior Learner</i> pertama kali.
15.		<i>Getting more used to learn foreign language</i>	<i>Achievement</i> didapatkan saat pengguna mendapatkan <i>status Senior Learner</i> pertama kali.
16.		<i>Good enough as a foreign language learner</i>	<i>Achievement</i> didapatkan saat pengguna mendapatkan <i>status Semi-Pro Learner</i> pertama kali.
17.		<i>I'm a pro learner</i>	<i>Achievement</i> didapatkan saat pengguna mendapatkan <i>status Pro Learner</i> pertama kali.
18.		<i>Become a legend</i>	<i>Achievement</i> didapatkan saat pengguna mendapatkan <i>status Legend</i> pertama kali.

#### 4. *Self-expression*

Pengguna memiliki kebebasan memberikan nama pengguna sesuai dengan keinginan dari pengguna dan kebebasan untuk memilih gambar pengguna sesuai dengan *achievement* yang telah dimiliki oleh pengguna. Kebebasan pengguna dalam sistem ini juga mencakup kebebasan memilih bahasa asing yang ingin dipelajari yang terdiri dari bahasa mandarin dan bahasa jepang.

## 5. *Competition*

Kompetisi adalah cara lain untuk mengumpulkan *point* yang diberikan kepada pengguna selain menyelesaikan materi dan melakukan *challenge*. Pengguna dapat mengadu kecerdasan dengan pengguna lainnya. Pengguna yang memulai pertarungan akan mengerjakan pertanyaan terlebih dahulu. Setelah itu, pengguna yang menjadi lawan akan menerima pemberitahuan pertarungan. Pengguna lawan dapat memulai mengerjakan pertanyaan yang belum lebih dari tiga hari. Jika dalam tiga hari pemberitahuan pertarungan tidak dibuka oleh lawannya, maka pertarungan dimenangkan oleh pengguna yang menantang.

Pemenang pertarungan ditentukan dengan membandingkan nilai yang didapatkan oleh masing-masing pengguna. Pengguna yang mendapatkan nilai lebih tinggi dari lawannya akan mendapatkan poin yang lebih besar dari pengguna yang kalah. Jika poin dari kedua pengguna sama, maka poin yang diperoleh oleh kedua pengguna sama banyaknya.

## 6. *Alturism*

Pengguna yang menggunakan sistem akan mendapatkan kemampuan berbahasa asing. Selain itu, pengguna juga akan mendapatkan peningkatan motivasi dalam mempelajari bahasa asing dengan adanya media pembelajaran yang lebih menarik.

### **3.8 Perancangan Kuesioner Evaluasi**

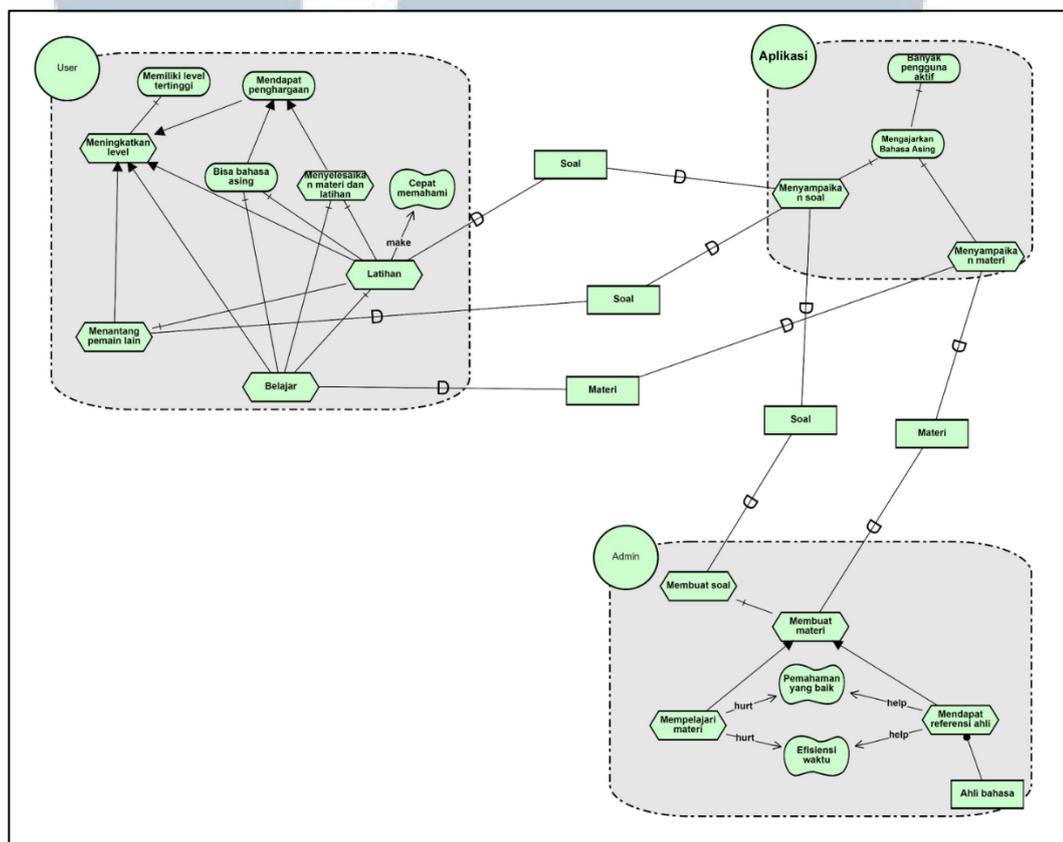
Kuesioner dibuat dalam beberapa aspek berdasarkan teori Hedonic Motivation System Adoption Model. Pernyataan yang disusun adalah sebagai berikut.

1. Kemudahan penggunaan (*Perceived Ease-of-Use*)
  - a. Interaksi saya dengan permainan sangat jelas dan mudah dimengerti.
  - b. Interaksi dengan permainan tidak membutuhkan banyak usaha mental.
  - c. Saya merasa permainan bebas dari masalah.
  - d. Saya merasa mudah untuk mendapatkan permainan melakukan apa yang saya inginkan.
  - e. Mempelajari cara memakai aplikasi menyenangkan untuk saya.
  - f. Mudah bagi saya untuk belajar mengoperasikan permainan.
  - g. Mudah bagi saya untuk menjadi terampil untuk mengoperasikan permainan.
  - h. Saya merasa aplikasi mudah dipakai.
2. Kegunaan (*Perceived Usefulness*)
  - a. Permainan mengurangi rasa beban pada pikiran saya.
  - b. Permainan membantu saya melewati waktu yang lebih baik.
  - c. Permainan menyediakan saya pelarian yang bermanfaat.
  - d. Permainan membantu saya berpikir lebih jernih.
  - e. Permainan membantu saya merasa lebih muda kembali.
3. Rasa penasaran (*Curiosity*)
  - a. Pengalaman ini membuat saya merasa penasaran.
  - b. Pengalaman ini membuat saya penasaran.
  - c. Pengalaman ini membangkitkan imajinasi saya.
4. Kesenangan (*Joy*)
  - a. Saya merasakan bermain permainan menjadi menyenangkan.
  - b. Saya bersenang-senang bermain permainan ini.

- c. Memainkan permainan ini membosankan.
  - d. Permainan ini sangat membuat saya kesal.
  - e. Pengalaman dari permainan ini menyenangkan.
  - f. Permainan ini membuatku tidak puas.
5. Kontrol (*Control*)
- a. Saya punya banyak kendali.
  - b. Saya bisa memilih dengan bebas apa yang saya ingin lihat atau lakukan.
  - c. Saya tidak bisa mengendalikan apa yang saya ingin lakukan.
  - d. Saya memegang kendali.
  - e. Saya tidak punya kendali atas interaksi saya.
  - f. Saya diizinkan mengendalikan interaksi saya.
6. Penggunaan kembali (*Behavioral Intention to Use*)
- a. Saya berencana untuk menggunakan aplikasi lagi di masa yang akan datang.
  - b. Saya akan terus menggunakannya di masa depan.
  - c. Saya berharap melanjutkan penggunaan saya di masa depan.
7. Keterikatan (*Focused Immersion*)
- a. Saya mampu memblokir gangguan lainnya.
  - b. Saya asyik dengan apa yang saya lakukan.
  - c. Saya tenggelam dalam permainan.
  - d. Saya dengan mudah terganggu oleh gangguan dari luar.
  - e. Perhatian saya tidak dapat dialihkan dengan mudah.

### 3.9 Analisis Kebutuhan Sistem

Sistem pada penelitian ini dibuat pada platform Android dan iOS menggunakan *framework* Ionic untuk aplikasinya, dan web untuk *content management system* atau disingkat menjadi CMS. Aplikasi memerlukan bantuan *application programming interface* atau API untuk melakukan proses *create, read, update* dan *delete* ke dalam database dari sistem. Untuk memperjelas kebutuhan sistem, dibuat *iStar 2.0 Hybrid Strategic Dependency/Strategic Rationale Model* yang dapat dilihat pada gambar 3.1.



Gambar 3.1 Diagram iStar 2.0

Aktor pengguna memiliki tujuan yaitu memiliki *level* tertinggi, mendapatkan penghargaan, dan bisa berbahasa asing. Untuk memiliki *level* tertinggi, pengguna harus meningkatkan *level*. Meningkatkan *level* dapat dilakukan dengan cara menantang pemain lain, mengerjakan latihan, belajar, atau mendapatkan

penghargaan. Pengguna akan mendapatkan penghargaan jika pengguna bisa berbahasa asing atau menyelesaikan materi dan latihan. Proses belajar akan membantu pengguna untuk lebih cepat memahami. Kegiatan latihan pada aktor pengguna membutuhkan soal yang didapatkan dari aktor aplikasi. Kegiatan belajar pada aktor pengguna membutuhkan materi yang didapatkan dari aktor aplikasi. Kegiatan menantang pemain lain pada aktor pengguna membutuhkan soal yang didapatkan dari aktor aplikasi.

Aktor aplikasi memiliki tujuan yaitu banyak pengguna aktif. Untuk memiliki banyak pengguna aktif, aplikasi harus mengajarkan bahasa asing. Aplikasi bisa mengajarkan bahasa asing dengan cara menyampaikan soal kepada pengguna, dan menyampaikan materi kepada pengguna. Kegiatan menyampaikan soal pada aktor aplikasi membutuhkan soal yang didapatkan dari aktor administrator. Kegiatan menyampaikan materi pada aktor aplikasi membutuhkan materi yang didapatkan dari aktor administrator.

Aktor Administrator memiliki kegiatan membuat soal. Kegiatan membuat soal bisa dilakukan jika aktor administrator melakukan kegiatan membuat materi. Aktor administrator bisa membuat materi dengan melakukan kegiatan mempelajari materi atau mendapatkan referensi ahli. Kegiatan mempelajari materi akan mengurangi pemahaman yang baik dan mengurangi efisiensi waktu. Kegiatan mendapatkan referensi ahli akan membantu pemahaman yang baik dan membantu efisiensi waktu. Kegiatan mendapatkan referensi ahli membutuhkan seorang ahli bahasa.

Pengguna aplikasi membutuhkan ID untuk menggunakan aplikasi, setiap ID bersifat unik atau tidak ada tiruannya. Pengguna yang belum atau tidak memiliki

ID dapat membuat ID menggunakan *e-mail*, dimana satu *e-mail* hanya bisa digunakan untuk membuat satu ID saja. Setelah pengguna memiliki ID, pengguna dapat memulai menggunakan aplikasi. ID pengguna berfungsi untuk menyimpan data-data milik pengguna. Selain itu, ID pengguna juga berfungsi untuk menyimpan progres milik pengguna.

Saat pertama kali pengguna membuka aplikasi, aplikasi akan menampilkan halaman *login*. Halaman *login* meminta pengguna memasukan ID miliknya. Jika pengguna tersebut belum memiliki ID, pengguna dapat berpindah ke halaman pendaftaran untuk membuat ID baru untuknya. Aplikasi ini memiliki dua pilihan bahasa yang dapat dipilih oleh pengguna untuk dipelajari. Bahasa tersebut adalah pelajaran bahasa mandarin dan pelajaran bahasa jepang. Pengguna dapat memilih pelajaran bahasa yang ingin dipelajari. Setiap pelajaran bahasa memiliki mata pelajarannya tersendiri. Setiap mata pelajaran memiliki bab yang didalamnya terdiri atas pertanyaan yang terdiri dari pilihan ganda, menyusun kalimat, dan melengkapi kalimat serta materi yang berupa kata dan kalimat.

Setiap materi di dalam subbab akan memberikan poin setelah materi diselesaikan. Pertanyaan dapat dikerjakan setelah pengguna menyelesaikan materi. Untuk membuka subbab selanjutnya, pengguna harus terlebih dahulu menyelesaikan pembelajaran materi sebelumnya dan pengguna juga harus menyelesaikan pertanyaan dengan minimal delapan puluh persen tingkat keberhasilan untuk membuka subbab selanjutnya. Pertanyaan yang berhasil diselesaikan akan memberikan poin kepada penggunanya. Poin yang didapat oleh pengguna dapat dilihat pada menu profil pengguna. Jumlah poin akan diakumulasikan menjadi *level*, dimana semakin tinggi *level* dari pengguna maka

semakin banyak poin yang harus dikumpulkan. Pada menu profil pengguna, pengguna juga dapat melihat dan memilih gambar pengguna sesuai dengan *achievement* yang telah didapatkan oleh pengguna.

Selain menu belajar, aplikasi juga memiliki halaman untuk bertarung. Di dalam menu bertarung, pengguna dapat memulai bertarung melawan pengguna lain, melihat riwayat pertarungan dan papan peringkat. Pada saat pengguna memilih bertarung dengan pengguna lain, aplikasi akan memilih pengguna lain sebagai lawan secara acak. Pengguna yang memulai pertarungan akan mengerjakan pertanyaan terlebih dahulu. Setelah itu, pengguna yang menjadi lawan akan menerima pemberitahuan pertarungan. Pengguna lawan dapat memulai mengerjakan pertanyaan yang belum lebih dari tiga hari. Jika dalam tiga hari pemberitahuan pertarungan tidak dibuka oleh lawannya, maka pertarungan dimenangkan oleh pengguna yang menantang. Pemenang pertarungan ditentukan dengan membandingkan nilai yang didapatkan oleh masing-masing pengguna. Pengguna yang mendapatkan nilai lebih tinggi dari lawannya akan mendapatkan poin yang lebih besar dari pengguna yang kalah. Jika poin dari kedua pengguna sama, maka poin yang diperoleh oleh kedua pengguna sama banyaknya. Riwayat pertarungan berisikan riwayat pertarungan yang telah terjadi dan melibatkan pengguna. Pada menu riwayat pertarungan, pengguna hanya bisa melihat riwayat lawan dan status dari pertarungan. Status pertarungan berisikan apakah pertarungan itu dimenangkan oleh pengguna atau lawannya.

Menu profil berisikan gambar pengguna, nama pengguna, poin, dan *level* pengguna. Selain itu pada menu profil, pengguna juga dapat melihat daftar peringkat pengguna. Daftar penghargaan yang dimiliki oleh pengguna juga dapat

dilihat pada menu profil. Pada menu pengaturan, pengguna dapat memilih pelajaran utama. Selain itu, pada menu pengaturan, pengguna juga dapat melihat informasi tentang aplikasi dan logout dari aplikasi.

Aplikasi menggunakan API untuk mendapatkan data bersifat dinamis yang berada di dalam *database*. Selain aplikasi dan API, sistem juga terdiri dari CMS yang dibangun menggunakan bahasa PHP dan menggunakan *framework* CodeIgniter dan bantuan *library* tambahan yaitu GroceryCRUD. Penggunaan GroceryCRUD mempermudah pembuatan tampilan daftar pengguna aplikasi. CMS memerlukan ID administrator untuk melakukan login. ID administrator hanya dapat didaftarkan oleh administrator lainnya. Setelah pengguna melakukan login menggunakan ID administrator, CMS akan menampilkan halaman awal. Halaman awal CMS berisi jumlah pengguna aplikasi, jumlah subbab, dan waktu terakhir administrator melakukan *login*. Pada CMS terdapat bar navigasi yang terdapat di sebelah kiri. Pada bar navigasi, administrator dapat berpindah ke halaman lain. Bar navigasi terbagi menjadi tiga yaitu navigasi pengguna, navigasi materi pembelajaran, dan gamifikasi.

Pada bagian navigasi pengguna, administrator dapat memilih menu pengguna untuk melihat daftar pengguna. Bagian navigasi pembelajaran berisikan daftar bahasa. Pada menu daftar bahasa, administrator dapat melihat daftar bahasa yang bisa dipelajari di dalam aplikasi. Pada daftar bahasa, administrator dapat melihat daftar bab yang dimiliki oleh setiap bahasa. Pada halaman daftar bab, administrator dapat melihat daftar bab yang ada pada sistem dan melakukan fungsi *add*, *update*, dan *delete* dengan bantuan *library* GroceryCRUD. Selain itu administrator juga dapat melihat subbab yang ada pada setiap bab dengan berpindah ke halaman

subbab. Pada halaman daftar subbab, administrator dapat melihat daftar subbab yang ada pada sistem dan melakukan fungsi *add*, *update*, dan *delete* dengan bantuan *library* GroceryCRUD. Selain itu administrator juga dapat melihat materi yang dimiliki setiap subbab yang terdiri dari kata, kalimat, dan pertanyaan dengan berpindah ke masing-masing halaman yang terdiri dari halaman daftar kata, daftar kalimat, dan daftar pertanyaan.

Pada halaman daftar kata, administrator dapat melihat daftar kata yang ada pada sistem dan melakukan fungsi *add*, *update*, dan *delete* dengan bantuan *library* GroceryCRUD. Pada halaman kalimat, administrator dapat melihat daftar kalimat yang ada pada sistem dan melakukan fungsi *add*, *update*, dan *delete* dengan bantuan *library* GroceryCRUD. Pada halaman pertanyaan, administrator dapat melihat daftar pertanyaan yang ada pada sistem dan melakukan fungsi *add*, *update*, dan *delete* dengan bantuan *library* GroceryCRUD. Pada halaman pertanyaan, administrator dapat berpindah ke halaman daftar jawaban untuk melihat, mengubah, menambahkan, dan menghapus jawaban yang ada.

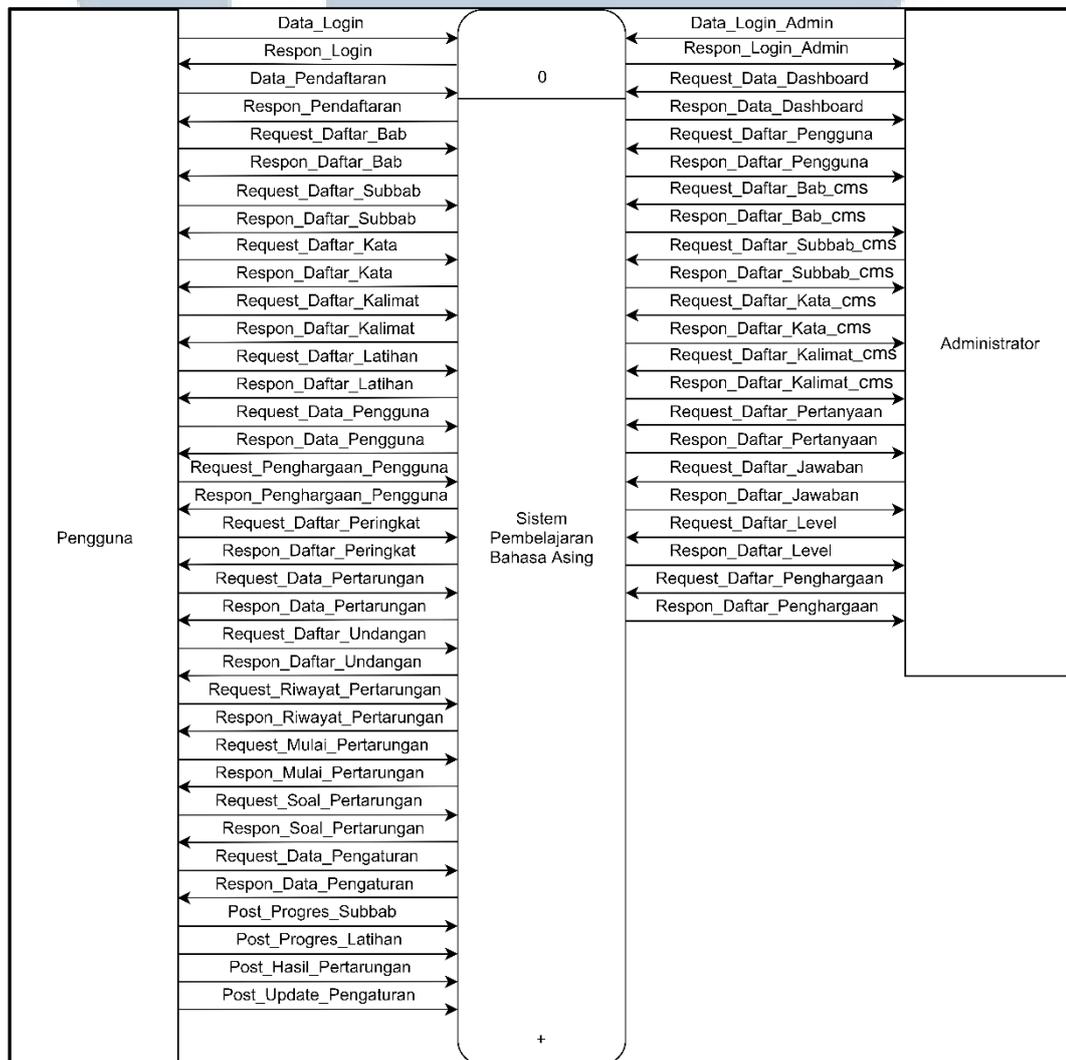
Pada bagian navigasi gamifikasi, administrator dapat memilih halaman penghargaan atau *level*. Pada halaman penghargaan, administrator dapat melihat daftar penghargaan yang ada pada sistem dan melakukan fungsi *add*, *update*, dan *delete* dengan bantuan *library* GroceryCRUD. Pada halaman *level*, administrator dapat melihat daftar *level* yang ada pada sistem dan melakukan fungsi *add*, *update*, dan *delete* dengan bantuan *library* GroceryCRUD.

### 3.10 Perancangan Sistem

Metode perancangan sistem yang digunakan pada penelitian ini adalah metode prosedural, sehingga dalam proses perancangan digunakan *Data Flow Diagram*, *Flowchart*, *Entity Relation Diagram*, *Database Schema*, struktur tabel, dan desain antarmuka dari sistem.

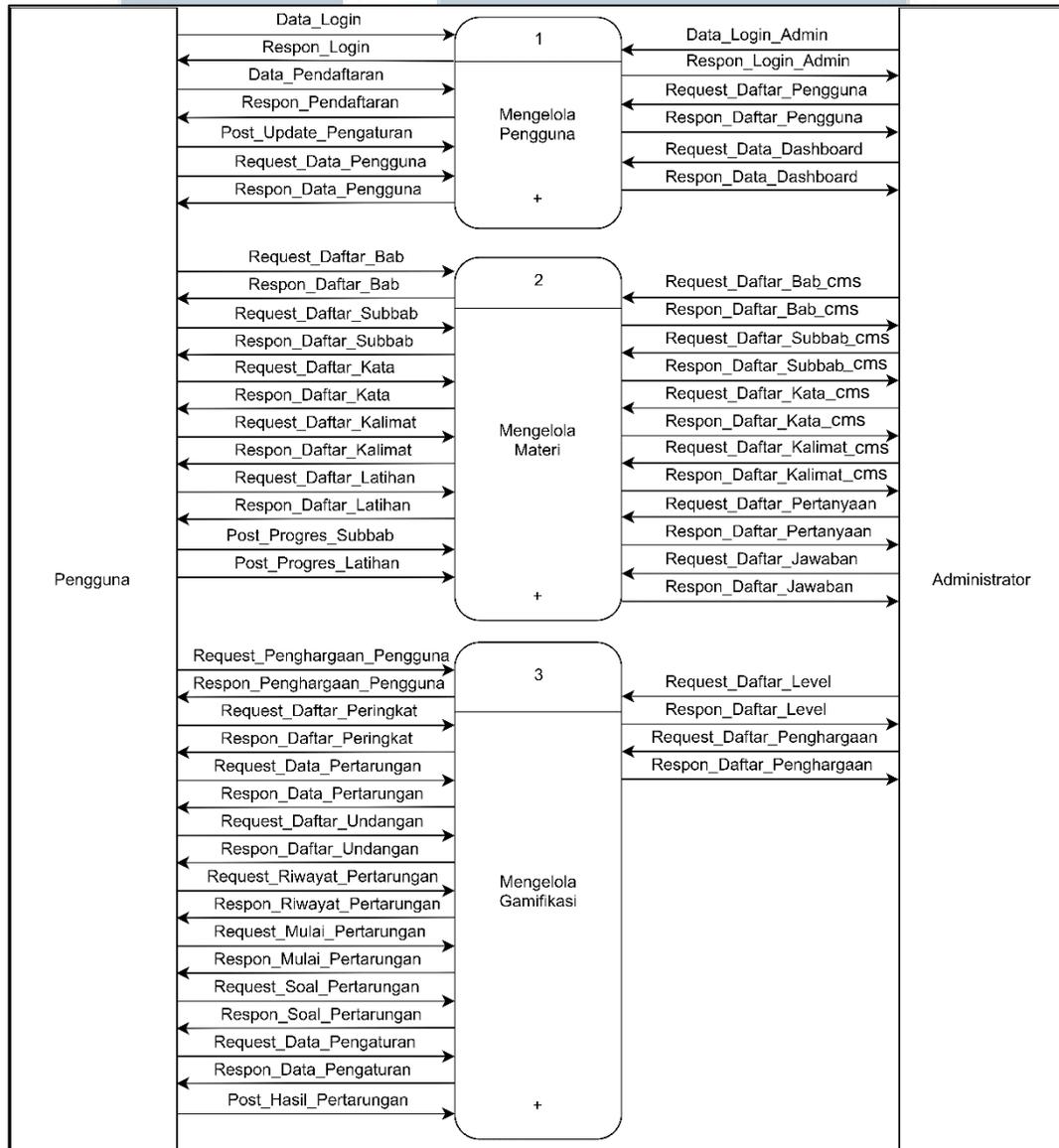
#### 3.10.1 Data Flow Diagram

*Data Flow Diagram* (DFD) dibuat dengan tujuan menggambarkan bagaimana alur data yang masuk dan keluar pada proses gamifikasi pembelajaran bahasa asing. DFD dibuat dalam bentuk diagram konteks, *level 0*, dan *level 1*.



Gambar 3.2 Diagram Konteks Gamifikasi Pembelajaran Bahasa Asing

Gambar 3.2 menunjukkan diagram konteks gamifikasi sistem pembelajaran bahasa asing. Terdapat dua entitas yaitu Pengguna dan Administrator. Pada diagram tersebut dapat dilihat terdapat tiga puluh satu data masuk ke dalam proses dan dua puluh tujuh data keluar dari proses. Pada Gambar 3.2 ditunjukkan DFD *level 1* untuk memperjelas kegunaan dari masing-masing alur data.

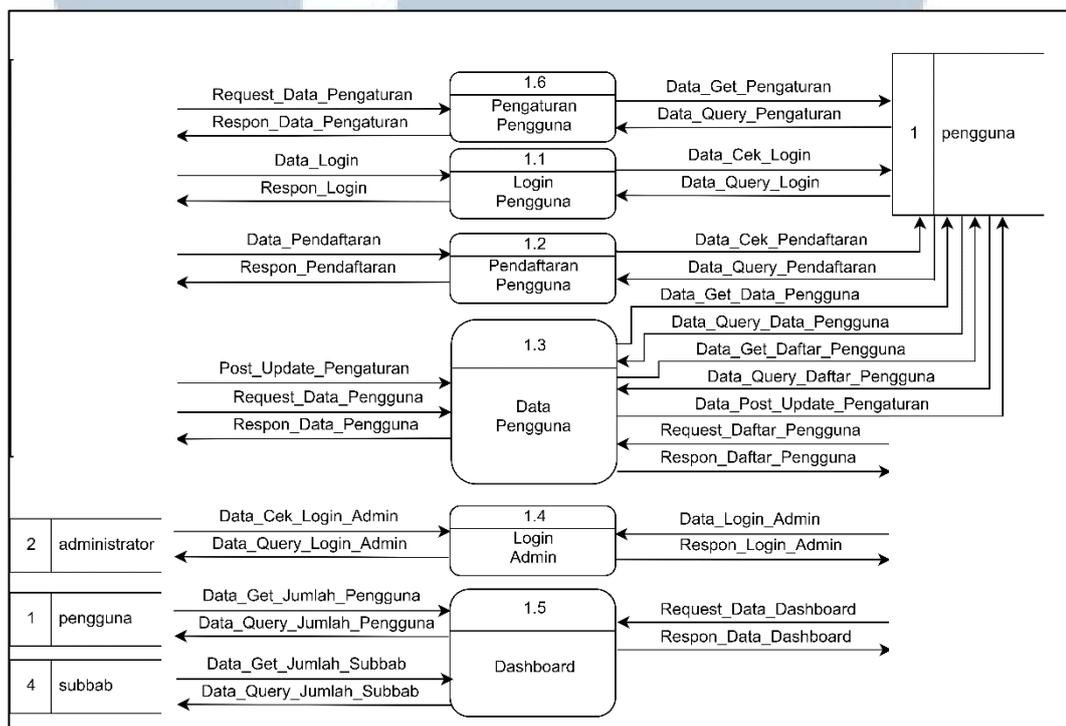


Gambar 3.3 DFD Level 0 Gamifikasi Pembelajaran Bahasa Asing

Gambar 3.3 menunjukkan DFD *level 0* dari proses gamifikasi pembelajaran bahasa asing. Proses gamifikasi pembelajaran bahasa asing dibagi menjadi tiga

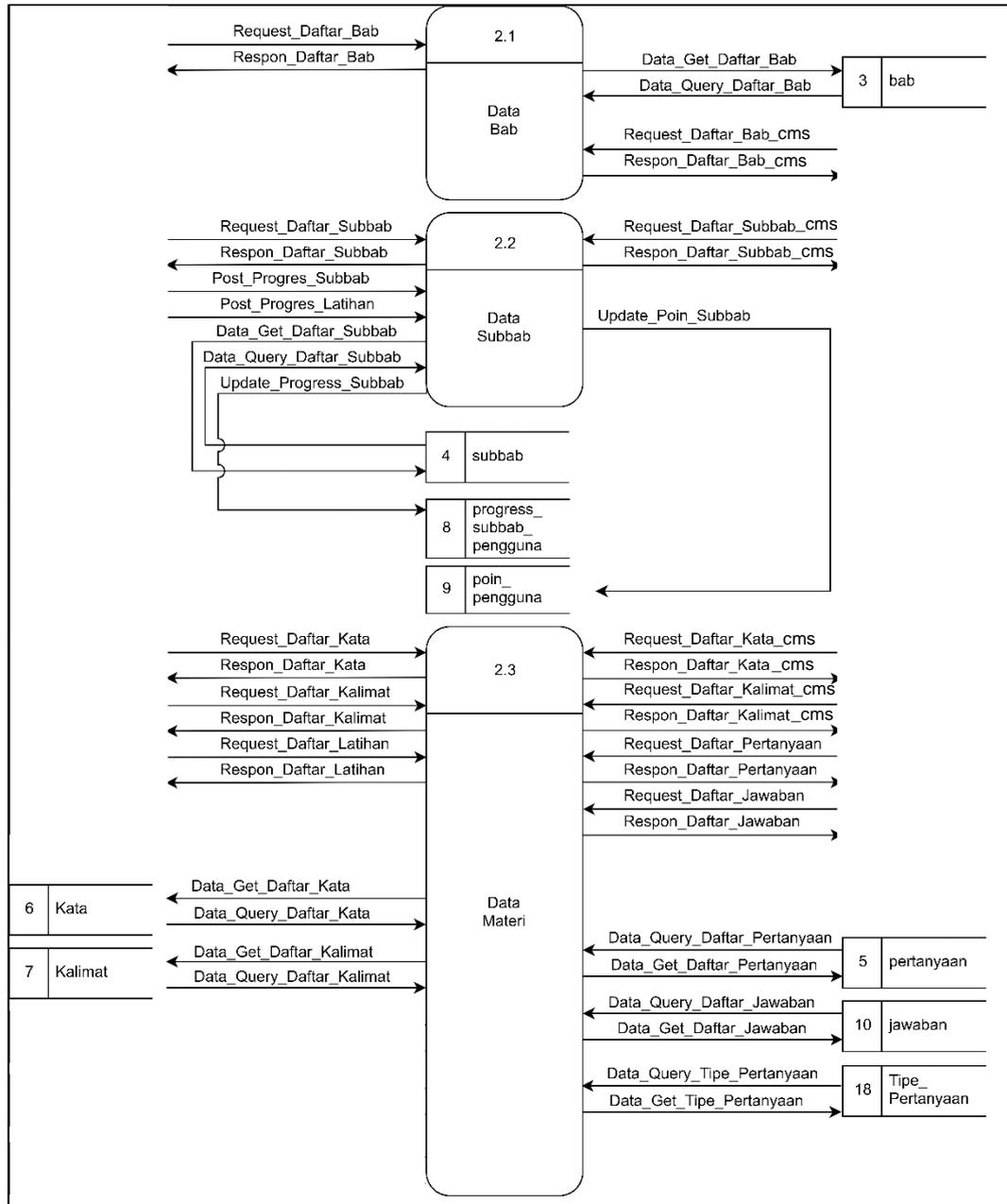
proses pada DFD *level 0* yaitu proses mengelola pengguna, proses mengelola materi, dan proses mengelola gamifikasi.

Pada proses mengelola pengguna terdapat tujuh data masuk ke dalam proses dan enam data keluar dari proses. Pada proses mengelola materi terdapat tiga belas data masuk ke dalam proses dan sebelas data keluar dari proses. Pada proses mengelola gamifikasi terdapat sebelas data masuk ke dalam proses dan sepuluh data keluar dari proses. Pada Gambar 3.3 ditunjukkan DFD *level 1* dari proses mengelola pengguna untuk memperjelas kegunaan dari proses.



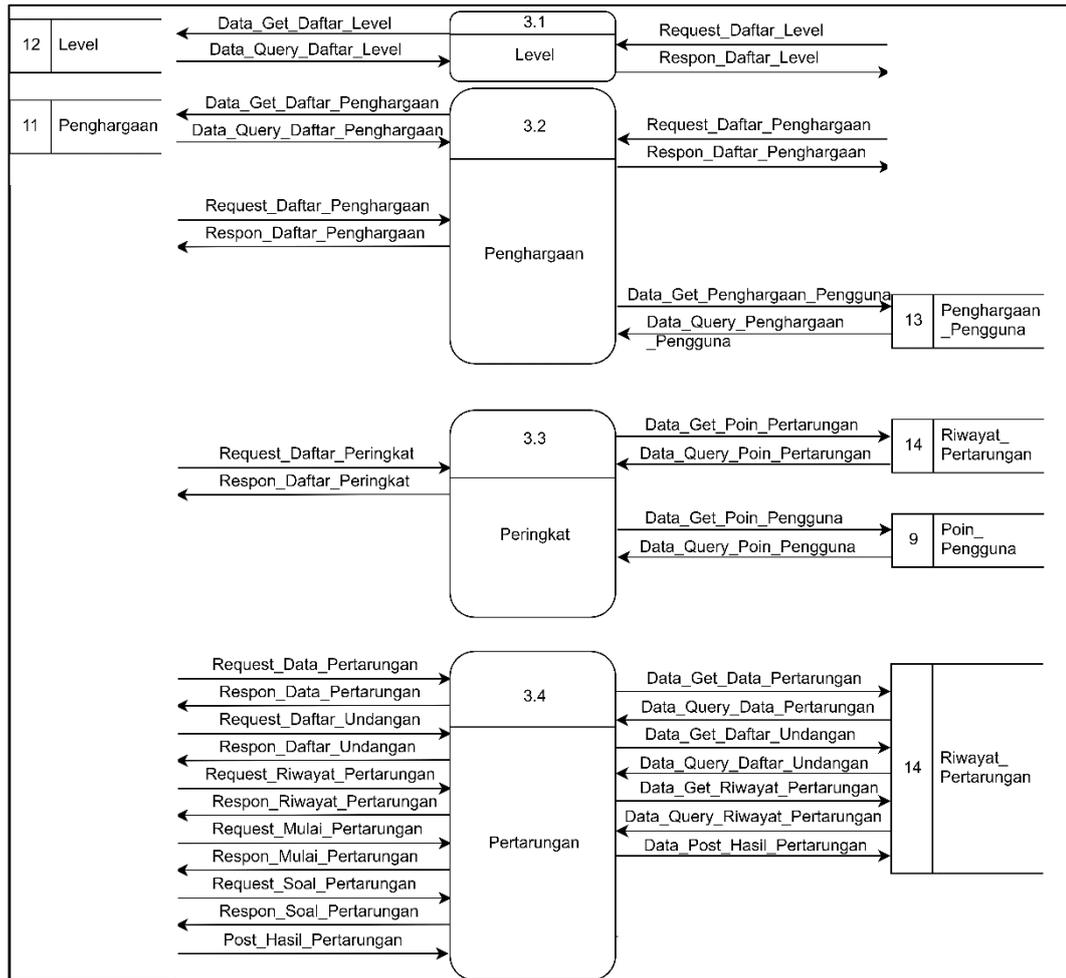
Gambar 3.4 DFD Level 1 Proses Mengelola Pengguna

Proses mengelola pengguna terbagi menjadi enam proses yang terdiri dari proses *login* pengguna, proses pendaftaran pengguna, proses data pengguna, proses login admin, proses *dashboard*, dan proses pengaturan pengguna. Selain itu, proses pengguna berhubungan dengan tiga tabel *database* yang terdiri dari tabel pengguna, tabel administrator, dan tabel subbab.



Gambar 3.5 DFD Level 1 Proses Mengelola Materi

Proses mengelola materi terbagi menjadi tiga proses yaitu proses data bab, proses data subbab, dan proses data materi. Selain itu, proses mengelola materi berhubungan dengan sembilan tabel *database* yang terdiri dari tabel bab, tabel subbab, tabel progres\_subbab\_pengguna, tabel poin\_pengguna, tabel pertanyaan, tabel jawaban, tabel tipe\_pertanyaan, tabel kata, dan tabel kalimat.

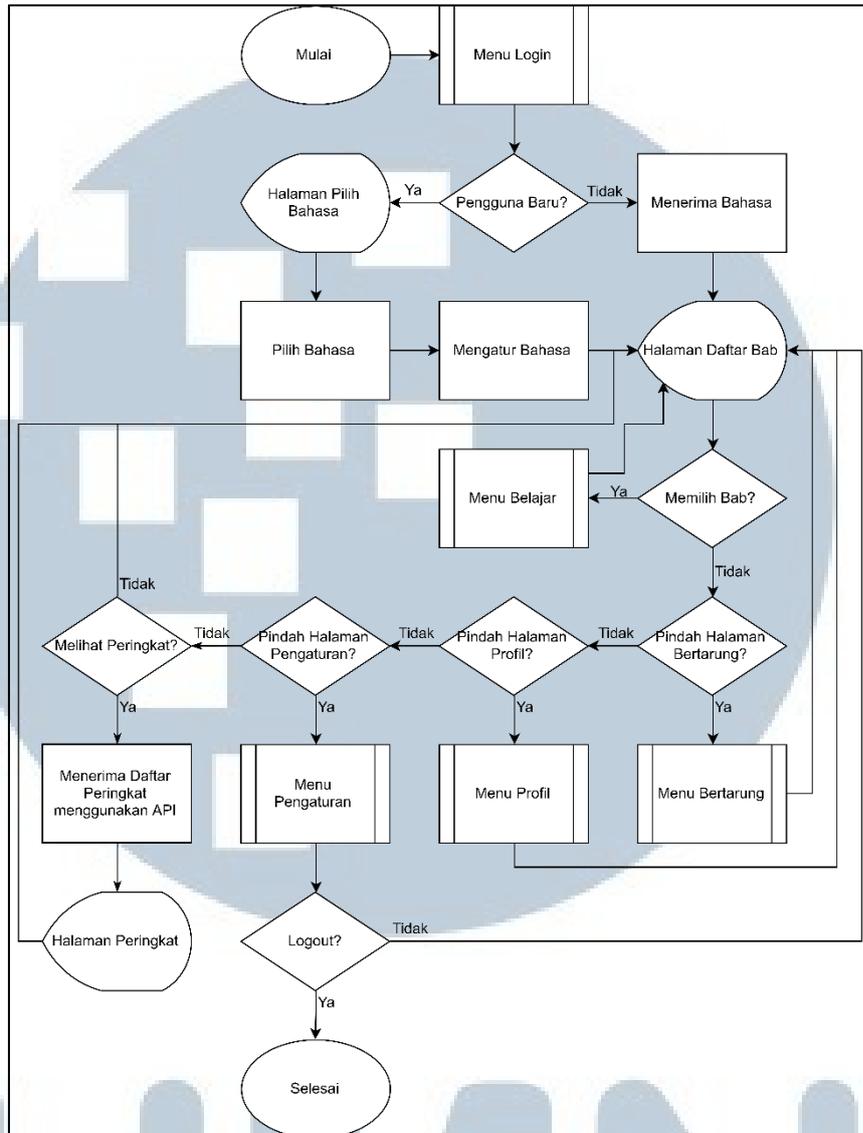


Gambar 3.6 DFD Level 1 Proses Mengelola Gamifikasi

Proses mengelola gamifikasi terbagi menjadi empat proses yaitu proses *level*, proses penghargaan, proses peringkat, dan proses pertarungan. Selain itu proses mengelola gamifikasi berhubungan dengan lima tabel *database* yang terdiri dari tabel *level*, tabel *penghargaan*, tabel *penghargaan\_pengguna*, tabel *riwayat\_penghargaan*, dan tabel *poin\_pengguna*.

### 3.10.2 Flowchart

*Flowchart* ditujukan untuk menggambarkan bagaimana alur proses pada sistem gamifikasi pembelajaran bahasa asing. Pada Gambar 3.7 ditunjukkan *flowchart* pada aplikasi pembelajaran bahasa asing.



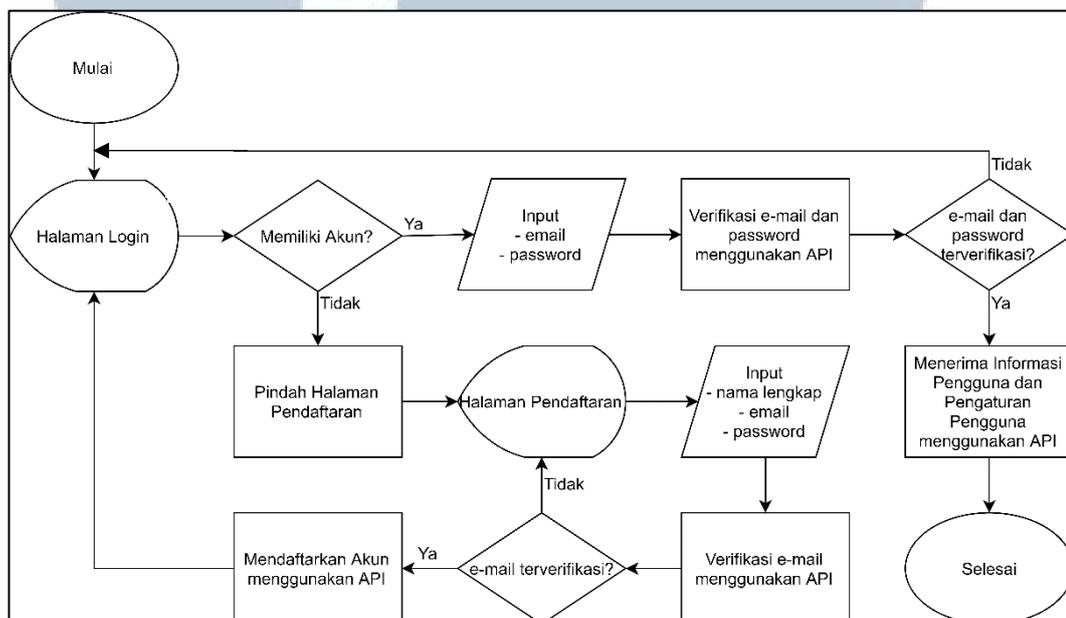
Gambar 3.7 Flowchart Aplikasi Pembelajaran Bahasa Asing

Gambar 3.7 menunjukkan *flowchart* pada aplikasi pembelajaran bahasa asing. Proses login ditunjukkan lebih detail pada Gambar 3.8. Setelah pengguna melakukan *login*, aplikasi akan memeriksa apakah pengguna merupakan pengguna baru. Jika pengguna merupakan pengguna baru, aplikasi akan menampilkan halaman pilih bahasa.

Pada halaman pilih bahasa, pengguna diminta untuk memilih bahasa yang ingin dipelajari oleh pengguna. Setelah pengguna memilih bahasa, aplikasi akan mengatur bahasa setelah itu aplikasi akan menunjukkan halaman daftar bab. Jika

pengguna bukan pengguna baru, aplikasi akan langsung menuju ke halaman daftar bab setelah melakukan proses menerima bahasa.

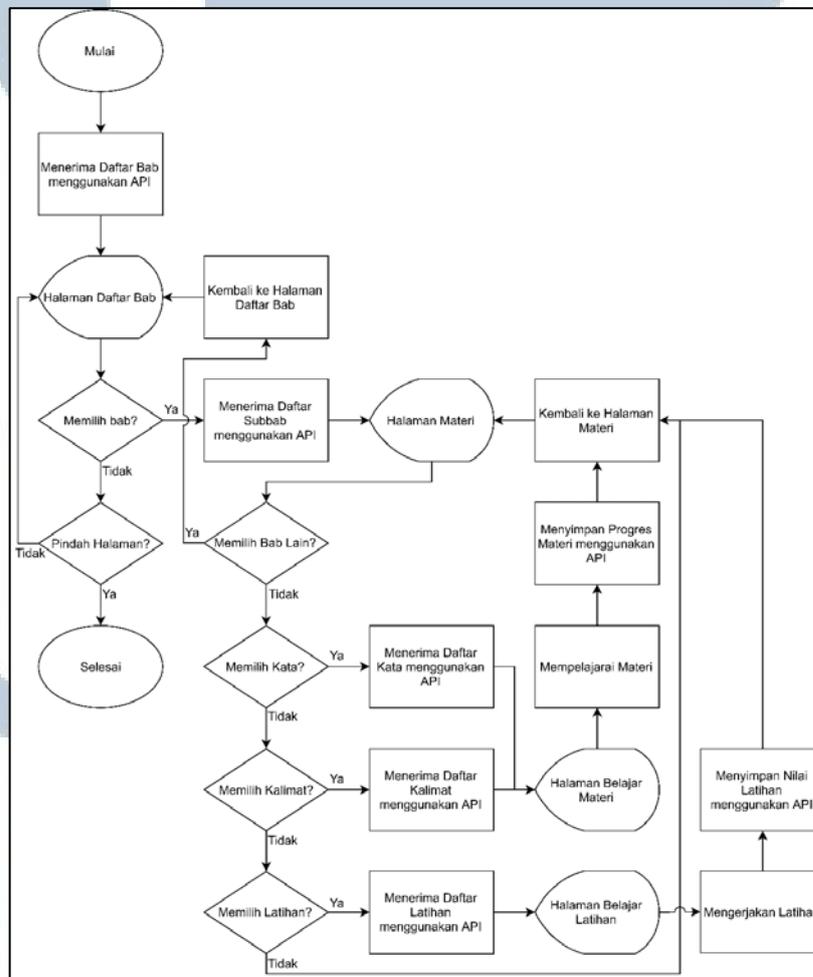
Jika pengguna memilih bab, aplikasi akan memulai proses belajar. Jika pengguna memilih halaman bertarung, aplikasi akan memulai proses menu bertarung. Jika pengguna memilih halaman profil, aplikasi akan memulai proses menu profil. Jika pengguna memilih halaman pengaturan, aplikasi akan memulai proses menu pengaturan. Aplikasi akan selesai jika pengguna melakukan *logout* pada proses menu pengaturan.



Gambar 3.8 Flowchart Proses Menu Login Aplikasi

Gambar 3.8 menunjukkan *flowchart* dari proses login aplikasi. Pada saat memulai proses, aplikasi akan menampilkan halaman login. Jika pengguna belum memiliki akun, maka aplikasi akan berpindah ke halaman pendaftaran dan menampilkan halaman pendaftaran. Pada halaman pendaftaran, pengguna diminta untuk memasukan nama lengkap, *e-mail*, dan *password*. Setelah itu aplikasi akan melakukan verifikasi *e-mail* menggunakan API.

Jika *e-mail* terverifikasi maka aplikasi akan mendaftarkan akun menggunakan API dan kembali ke halaman login. Jika tidak terverifikasi maka aplikasi akan kembali ke halaman pendaftaran. Saat pengguna telah memiliki akun, aplikasi akan meminta pengguna untuk memasukkan *e-mail* dan *password* pengguna. Setelah itu aplikasi akan melakukan verifikasi *e-mail* dan *password* menggunakan API. Jika *e-mail* dan *password* tidak terverifikasi, maka aplikasi akan kembali menampilkan halaman login. Jika *e-mail* dan *password* terverifikasi maka aplikasi akan menerima informasi pengguna dan pengaturan pengguna menggunakan API lalu proses *login* akan selesai.



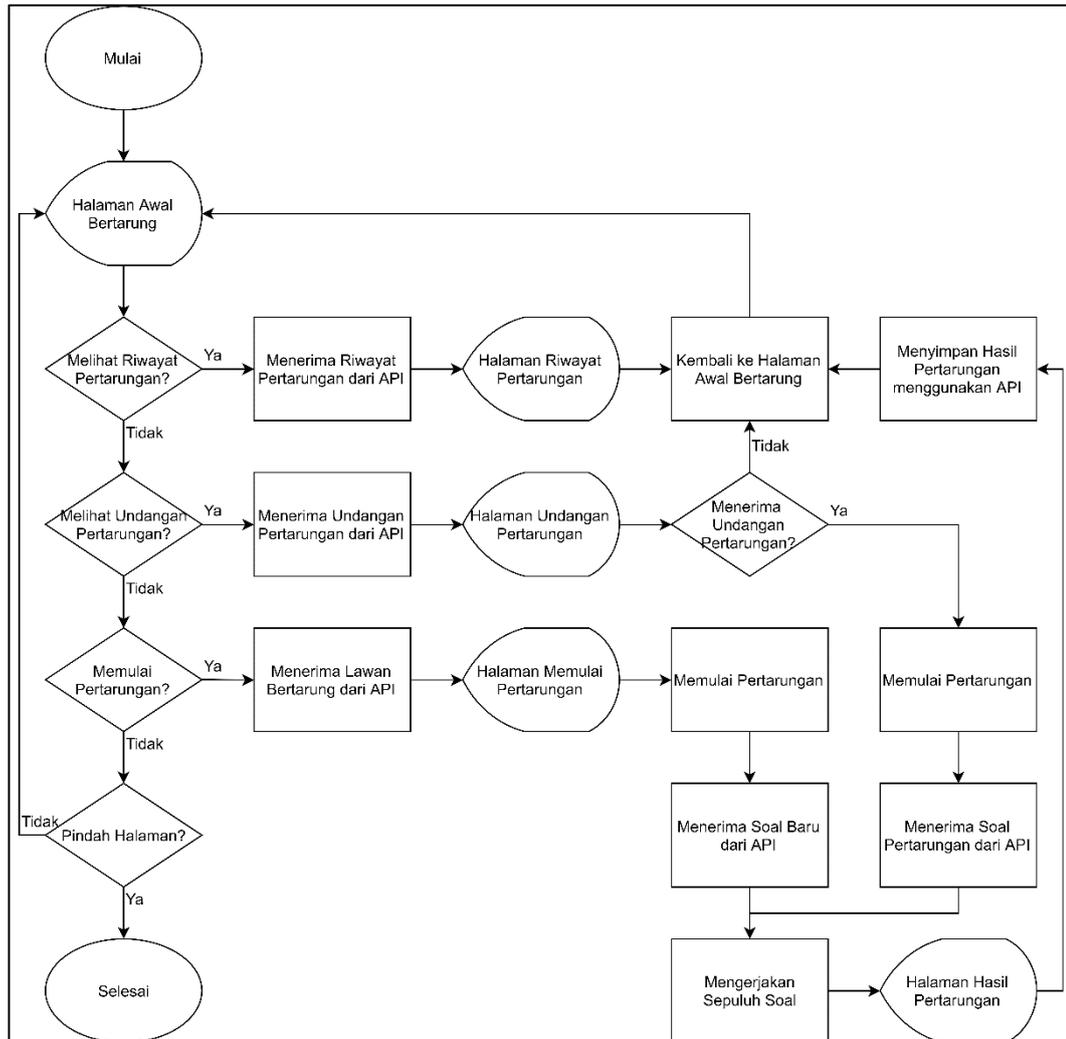
Gambar 3.9 Flowchart Proses Menu Belajar Aplikasi

Gambar 3.9 menunjukkan *flowchart* dari proses belajar aplikasi. Pada saat memulai proses, aplikasi akan menerima daftar bab menggunakan API. Setelah melakukan proses tersebut, aplikasi akan menampilkan halaman daftar bab. Jika pengguna memilih bab, maka aplikasi akan menerima daftar subbab dan berpindah ke halaman materi. Jika pengguna memilih bab lain maka aplikasi akan kembali ke halaman daftar bab.

Pada halaman materi, jika pengguna memilih kata maka aplikasi akan menerima daftar kata menggunakan API. Setelah itu aplikasi akan menampilkan halaman belajar materi. Jika pengguna memilih kalimat, maka aplikasi akan menerima daftar kalimat menggunakan API. Setelah itu aplikasi akan menampilkan halaman belajar materi. Pada halaman belajar materi, pengguna akan melalui proses belajar materi. Setelah itu aplikasi akan melakukan proses penyimpanan progres materi menggunakan API. Setelah itu aplikasi akan kembali menampilkan halaman materi.

Jika pengguna memilih latihan, aplikasi akan menerima daftar latihan menggunakan API. Setelah itu pengguna akan melalui proses mengerjakan latihan. Setelah itu aplikasi akan melakukan proses penyimpanan nilai latihan menggunakan API dan aplikasi akan kembali menampilkan halaman materi. Proses belajar akan selesai saat pengguna pindah ke halaman lain.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



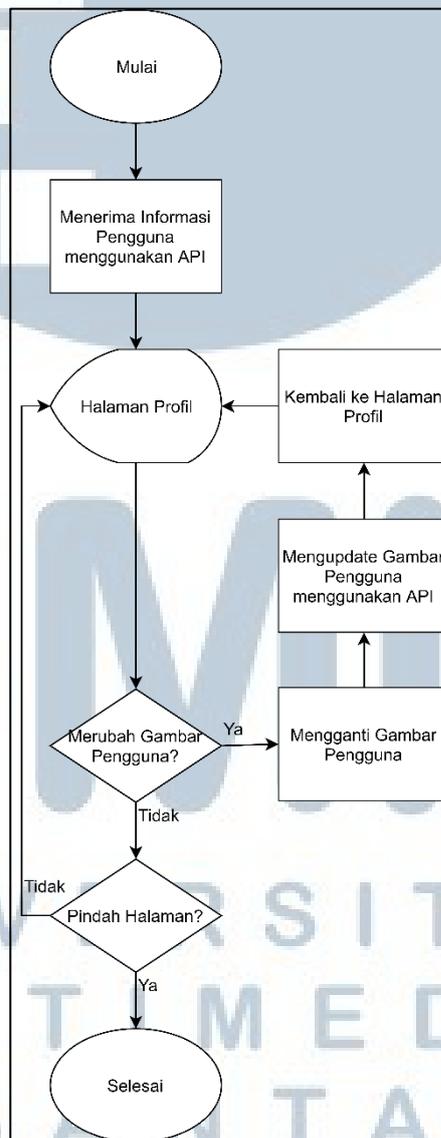
Gambar 3.10 Flowchart Proses Menu Bertarung Aplikasi

Gambar 3.10 menunjukkan *flowchart* dari proses bertarung aplikasi. Pada saat memulai proses, aplikasi akan menampilkan halaman awal bertarung. Jika pengguna melihat riwayat pertarungan, aplikasi akan menerima riwayat pertarungan dari API lalu menampilkan halaman riwayat pertarungan. Pengguna dapat kembali ke halaman awal bertarung dari halaman riwayat pertarungan.

Jika pengguna melihat undangan pertarungan, aplikasi akan menerima undangan pertarungan dari API lalu menampilkan halaman undangan pertarungan. Jika pengguna menerima pertarungan pada halaman pertarungan, aplikasi akan memulai pertarungan. Aplikasi akan menerima soal pertarungan dari API.

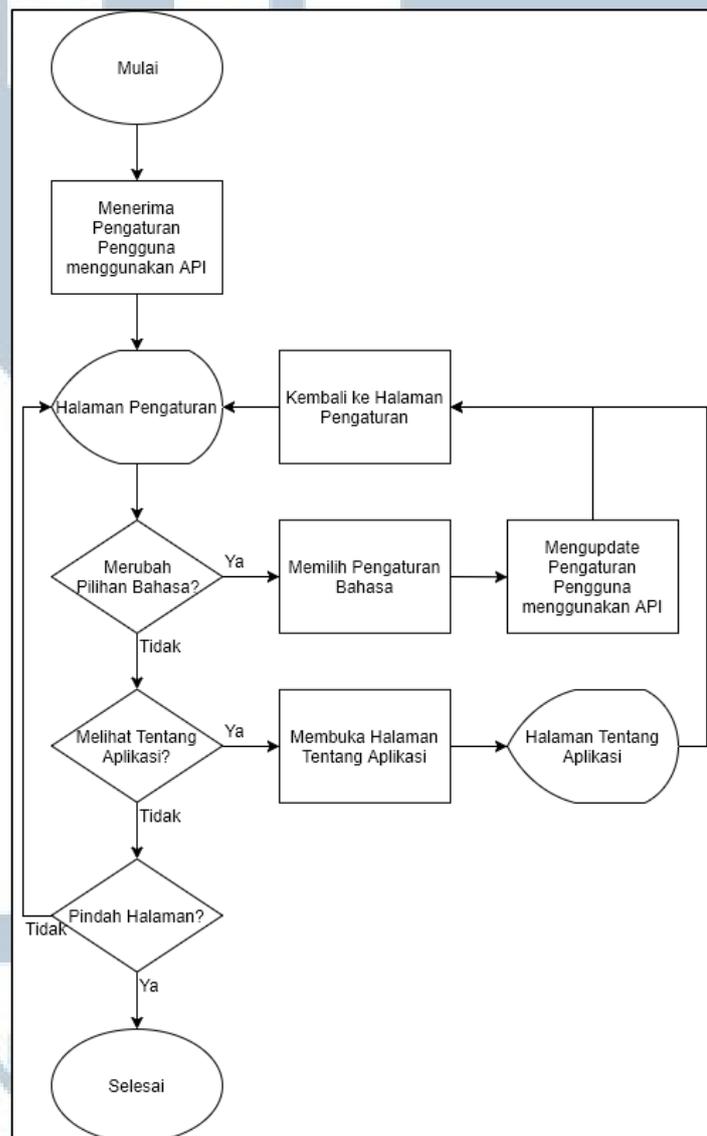
Jika pengguna memulai pertarungan, aplikasi akan menerima lawan bertarung dari API lalu menampilkan halaman memulai pertarungan. Saat pengguna memulai pertarungan, aplikasi akan menerima soal baru dari API.

Setelah itu pengguna akan mengerjakan sepuluh soal. Setelah mengerjakan soal, aplikasi akan menampilkan halaman hasil pertarungan. Setelah itu aplikasi akan menyimpan hasil pertarungan menggunakan API lalu kembali ke halaman awal pertarungan. Proses menu bertarung akan selesai saat pengguna pindah ke halaman lain.



Gambar 3.11 Flowchart Proses Menu Profil Aplikasi

Gambar 3.11 menunjukkan *flowchart* pada proses profil aplikasi. Pada saat memulai proses, aplikasi akan menerima informasi pengguna menggunakan API. Setelah melakukan proses tersebut, aplikasi akan menampilkan halaman profil. Pada halaman profil, jika pengguna mengubah gambar pengguna maka aplikasi akan mengganti gambar pengguna dan memperbaharui gambar pengguna menggunakan API.

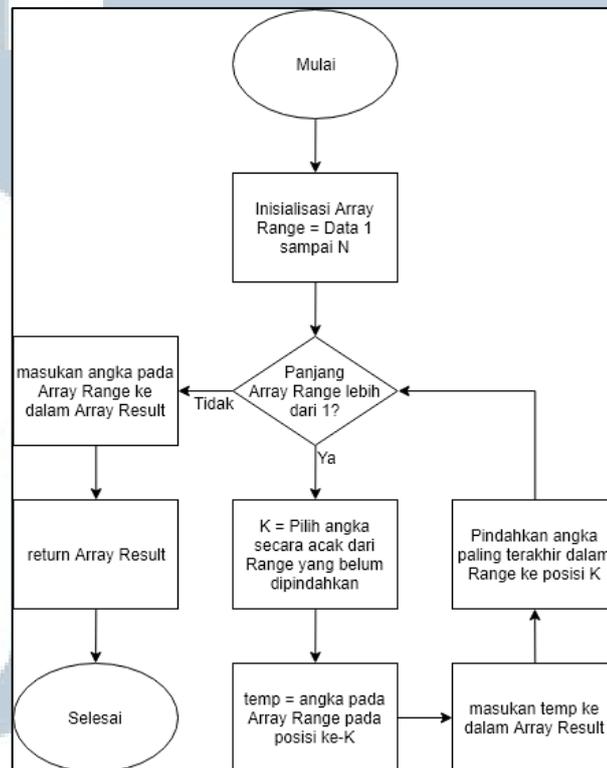


Gambar 3.12 Flowchart Proses Menu Pengaturan Aplikasi

Gambar 3.12 menunjukkan *flowchart* dari proses pengaturan aplikasi. Pada saat memulai proses, aplikasi akan menerima pengaturan pengguna menggunakan

API. Setelah melakukan proses tersebut, aplikasi akan menampilkan halaman pengaturan. Pada halaman pengaturan, jika pengguna mengubah pilihan bahasa maka aplikasi akan meminta pengguna memilih pengaturan bahasa lalu aplikasi akan memperbaharui pengaturan pengguna menggunakan API. Setelah selesai, aplikasi akan kembali menampilkan halaman pengaturan.

Jika pengguna melihat tentang aplikasi, aplikasi akan membuka halaman tentang aplikasi dan menampilkannya. Setelah itu, aplikasi akan kembali menampilkan halaman pengaturan. Proses pengaturan aplikasi akan selesai saat pengguna pindah ke halaman lain.

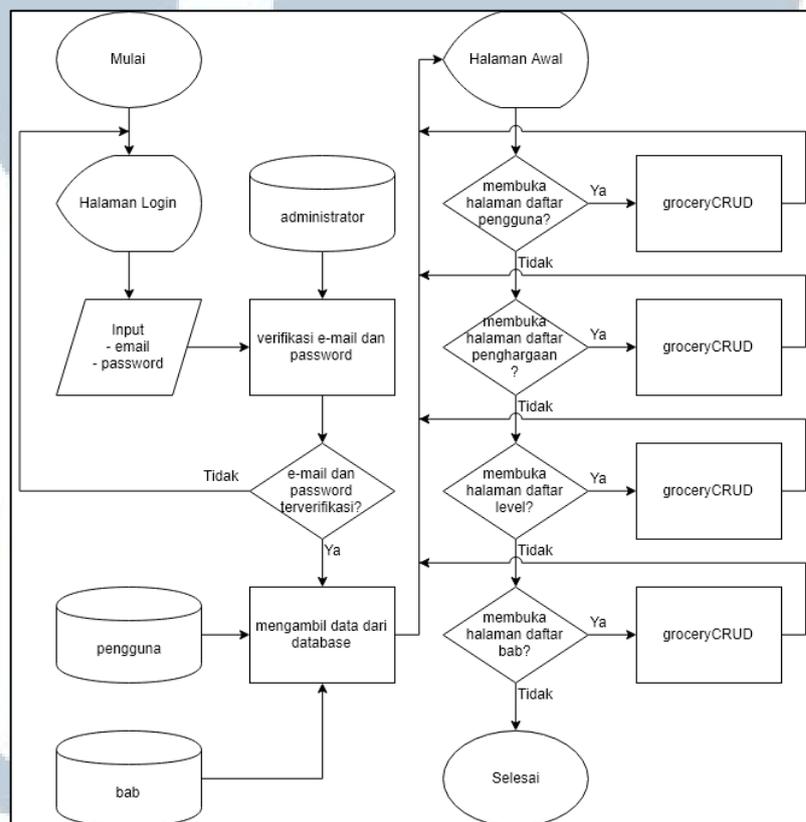


Gambar 3.13 Flowchart Proses Algoritma Knuth Shuffle

Gambar 3.13 menunjukkan *flowchart* dari proses Algoritma Knuth Shuffle. Pada saat memulai proses, aplikasi akan melakukan inisialisasi Array Range dengan data 1 sampai data N. Jika panjang Array Range tidak lebih dari satu, maka angka

pada Array Range akan dipindahkan ke dalam Array Result dan proses akan mengembalikan Array Result.

Jika panjang Array Range lebih dari satu, maka akan dibuat nilai K dengan cara memilih acak dari Array Range yang belum dipindahkan. Setelah itu simpan nilai yang ada pada K dalam variabel temp dan dimasukkan kedalam Array Result. Angka yang ada pada posisi paling terakhir Array Range akan menggantikan nilai yang ada pada posisi K. Proses ini akan diulang hingga panjang Array Range tidak lebih dari satu.



Gambar 3.14 Flowchart Backend

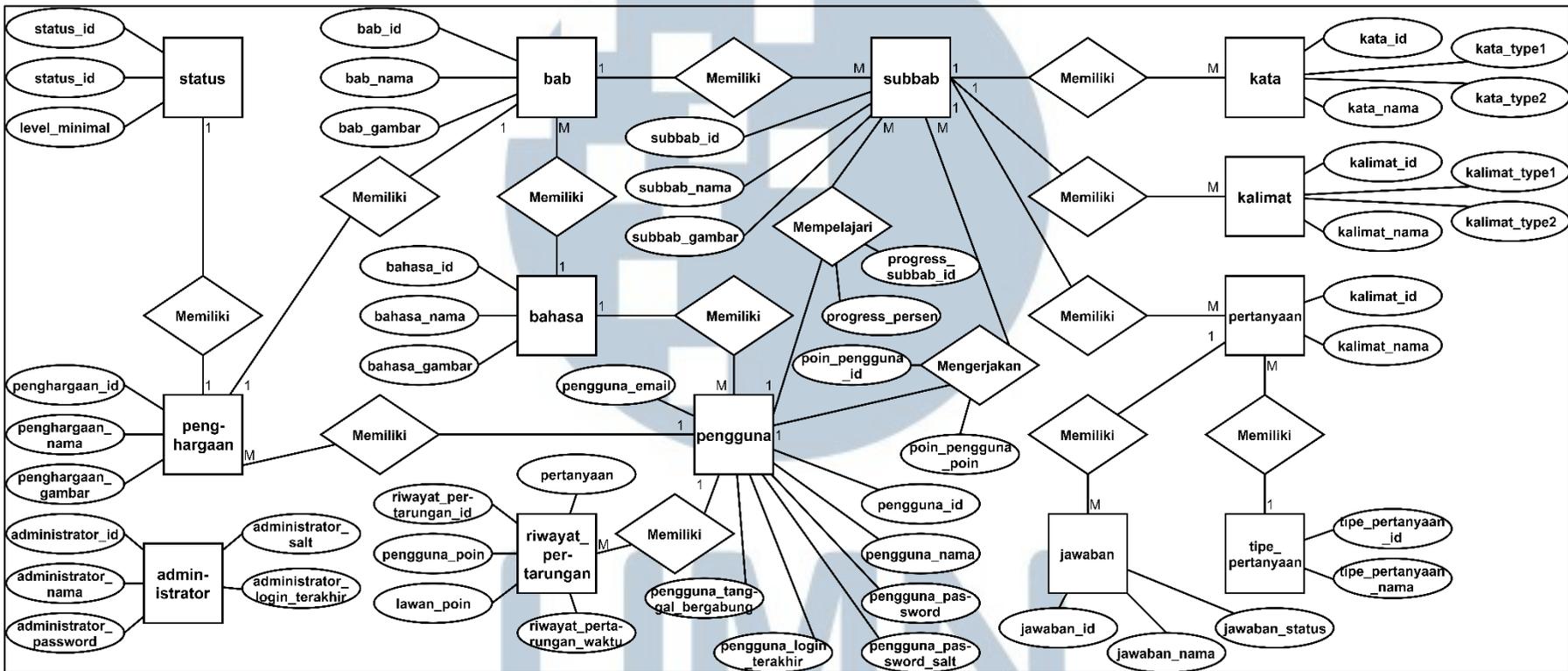
Gambar 3.14 menunjukkan *flowchart* dari *backend*. Backend aplikasi menampilkan halaman login pada saat pertama kali dibuka. Pada halaman ini administrator diminta untuk login terlebih dahulu menggunakan e-mail dan password. Setelah administrator memasukkan e-mail dan password, akan dilakukan

proses verifikasi e-mail dan password. Jika e-mail dan password tidak terverifikasi maka backend akan kembali menampilkan halaman login.

Jika e-mail dan password terverifikasi, backend akan mengambil data dari database lalu menampilkan halaman awal. Jika pada halaman awal administrator membuka halaman daftar pengguna, maka backend akan melakukan proses groceryCRUD pengguna. Jika pada halaman awal administrator membuka halaman daftar penghargaan, maka backend akan melakukan proses groceryCRUD penghargaan. Jika pada halaman awal administrator membuka halaman daftar level, maka backend akan melakukan proses groceryCRUD level. Jika pada halaman awal administrator membuka halaman daftar bab, maka backend akan melakukan proses groceryCRUD bab.

### **3.10.3 Entity Relation Diagram**

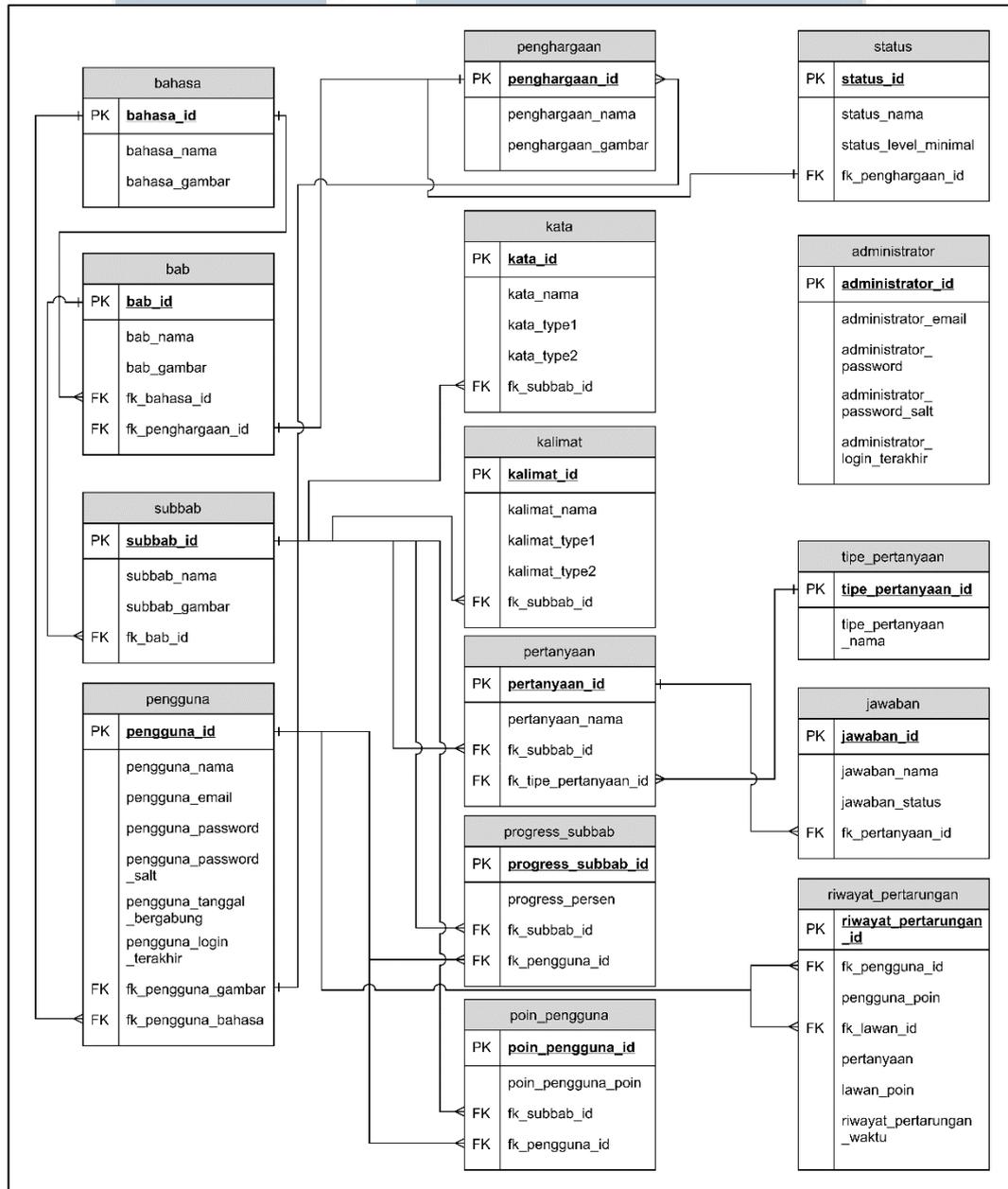
Entity Relation Diagram (ERD) digunakan untuk mengilustrasikan hubungan antar entitas dalam database (Chapple, 2017). Pada Gambar 3.14 dapat dilihat ERD dari sistem pada penelitian ini. Entitas administrator berisi data administrator. Entitas bahasa menyimpan data bahasa. Entitas bab mencatat bab pada bahasa. Entitas subbab mencatat subbab yang dimiliki bab. Entitas kata mencatat kata pada subbab. Entitas kalimat mencatat kalimat yang pada subbab. Entitas tipe pertanyaan menyimpan tipe pertanyaan. Entitas pertanyaan mencatat pertanyaan yang dimiliki oleh setiap subbab. Entitas jawaban mencatat jawaban yang dimiliki oleh setiap pertanyaan. Entitas penghargaan menyimpan data penghargaan. Entitas pengguna menyimpan data mengenai pengguna. Entitas riwayat pertarungan mencatat riwayat pertarungan yang dimiliki oleh pengguna.



Gambar 3.15 Entity Relation Diagram Sistem

### 3.10.4 Database Schema

*Database schema* dibuat untuk menunjukkan struktur kerangka tentang bagaimana data diatur dan bagaimana hubungan antara setiap data saling terkait. Gambar 3.15 menunjukkan *database schema* yang sudah dirancang berdasarkan ERD yang telah dibuat.



Gambar 3.16 Database Schema dari Sistem Pembelajaran Bahasa Asing

NUSANTARA

### 3.10.5 Struktur Tabel

Struktur tabel dibuat untuk menjelaskan data apa saja yang terdapat di dalam sistem basis data yang digunakan untuk penelitian ini.

Tabel 3.3 Struktur Tabel Bahasa

No.	Nama Kolom	Tipe Data	Keterangan
1.	bahasa_id	int (11)	<i>Primary Key, Auto Increment</i>
2.	bahasa_nama	varchar (255)	Nama dari bahasa
3.	bahasa_gambar	varchar (255)	Gambar dari bahasa

Tabel 3.3 menunjukkan Struktur Tabel Bahasa. Pada Tabel Bahasa terdapat tiga data yang digunakan. Data bahasa\_id memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data bahasa\_nama memiliki tipe data varchar (255) dan berfungsi untuk menyimpan nama dari bahasa. Data bahasa\_gambar memiliki tipe varchar (255) dan berfungsi untuk menyimpan URL tempat gambar disimpan.

Tabel 3.4 Struktur Tabel Penghargaan

No.	Nama Kolom	Tipe Data	Keterangan
1.	penghargaan_id	int (11)	<i>Primary Key, Auto Increment</i>
2.	penghargaan_nama	varchar (255)	Nama dari penghargaan
3.	penghargaan_gambar	varchar (255)	Gambar dari penghargaan

Tabel 3.4 menunjukkan Struktur Tabel Penghargaan. Pada Tabel Penghargaan terdapat tiga data yang digunakan. Data penghargaan\_id memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data penghargaan\_nama memiliki tipe data varchar (255) dan berfungsi untuk menyimpan nama dari penghargaan. Data penghargaan\_gambar memiliki tipe varchar (255) dan berfungsi untuk menyimpan URL tempat gambar disimpan.

Tabel 3.5 Struktur Tabel Bab

No.	Nama Kolom	Tipe Data	Keterangan
1.	<code>bab_id</code>	int (11)	<i>Primary Key, Auto Increment</i>
2.	<code>bab_nama</code>	varchar (255)	Nama dari bab
3.	<code>bab_gambar</code>	varchar (255)	Gambar dari bab
4.	<code>fk_bahasa_id</code>	int (11)	<i>Foreign Key</i> dari <code>bahasa_id</code> tabel bahasa
5.	<code>fk_penghargaan_id</code>	int (11)	<i>Foreign Key</i> dari <code>penghargaan_id</code> tabel penghargaan

Tabel 3.5 menunjukkan Struktur Tabel Bab. Pada Tabel Bab terdapat lima data yang digunakan. Data `bab_id` memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data `bab_nama` memiliki tipe data varchar (255) dan berfungsi untuk menyimpan nama dari bab. Data `bab_gambar` memiliki tipe varchar (255) dan berfungsi untuk menyimpan URL tempat gambar disimpan. Data `fk_bahasa_id` memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari `bahasa_id` tabel bahasa. Data `fk_penghargaan_id` memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari `penghargaan_id` tabel penghargaan.

Tabel 3.6 Struktur Tabel Subbab

No.	Nama Kolom	Tipe Data	Keterangan
1.	<code>subbab_id</code>	int (11)	<i>Primary Key, Auto Increment</i>
2.	<code>subbab_nama</code>	varchar (255)	Nama dari subbab
3.	<code>subbab_gambar</code>	varchar (255)	Gambar dari subbab
4.	<code>fk_bab_id</code>	int (11)	<i>Foreign Key</i> dari <code>bab_id</code> tabel bab

Tabel 3.6 menunjukkan Struktur Tabel Subbab. Pada Tabel Subbab terdapat empat data yang digunakan. Data `subbab_id` memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data `subbab_nama` memiliki tipe data varchar (255) dan berfungsi untuk menyimpan nama dari subbab. Data `subbab_gambar` memiliki tipe varchar (255) dan berfungsi untuk menyimpan URL tempat gambar disimpan. Data `fk_bab_id` memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari `bab_id` tabel bab

Tabel 3.7 Struktur Tabel Kata

No.	Nama Kolom	Tipe Data	Keterangan
1.	kata_id	int (11)	<i>Primary Key, Auto Increment</i>
2.	kata_nama	varchar (255)	Nama dari kata
3.	kata_type1	varchar (255)	Tipe 1 dari kata
4.	kata_type2	varchar (255)	Tipe 2 dari kata
5.	fk_subbab_id	int (11)	<i>Foreign Key</i> dari subbab_id tabel subbab

Tabel 3.7 menunjukkan Struktur Tabel Kata. Pada Tabel Kata terdapat lima data yang digunakan. Data kata\_id memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data kata\_nama memiliki tipe data varchar (255) dan berfungsi untuk menyimpan nama dari kata. Data kata\_type1 memiliki tipe data varchar (255) dan berfungsi untuk menyimpan tipe 1 dari kata. Data kata\_type2 memiliki tipe data varchar (255) dan berfungsi untuk menyimpan tipe 2 dari kata. Data fk\_bahasa\_id memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari bahasa\_id tabel bahasa.

Tabel 3.8 Struktur Tabel Kalimat

No.	Nama Kolom	Tipe Data	Keterangan
1.	kalimat_id	int (11)	<i>Primary Key, Auto Increment</i>
2.	kalimat_nama	varchar (255)	Nama dari kalimat
3.	kalimat_type1	varchar (255)	Tipe 1 dari kalimat
4.	kalimat_type2	varchar (255)	Tipe 2 dari kalimat
5.	fk_subbab_id	int (11)	<i>Foreign Key</i> dari subbab_id tabel subbab

Tabel 3.8 menunjukkan Struktur Tabel Kalimat. Pada Tabel Kalimat terdapat lima data yang digunakan. Data kalimat\_id memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data kalimat\_nama memiliki tipe data varchar (255) dan berfungsi untuk menyimpan nama dari kalimat. Data kalimat\_type1 memiliki tipe data varchar (255) dan berfungsi untuk menyimpan tipe 1 dari kalimat. Data kalimat\_type2 memiliki tipe data varchar (255) dan

berfungsi untuk menyimpan tipe 2 dari kalimat. Data *fk\_subbab\_id* memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari *subbab\_id* tabel *subbab*.

Tabel 3.9 Struktur Tabel Tipe Pertanyaan

No.	Nama Kolom	Tipe Data	Keterangan
1.	<i>tipe_pertanyaan_id</i>	int (11)	<i>Primary Key, Auto Increment</i>
2.	<i>tipe_pertanyaan_nama</i>	varchar (255)	Nama dari tipe pertanyaan

Tabel 3.9 menunjukkan Struktur Tabel Tipe Pertanyaan. Pada Tabel Tipe Pertanyaan terdapat dua data yang digunakan. Data *tipe\_pertanyaan\_id* memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data *tipe\_pertanyaan\_nama* memiliki tipe data varchar (255) dan berfungsi untuk menyimpan nama dari tipe pertanyaan.

Tabel 3.10 Struktur Tabel Pertanyaan

No.	Nama Kolom	Tipe Data	Keterangan
1.	<i>pertanyaan_id</i>	int (11)	<i>Primary Key, Auto Increment</i>
2.	<i>pertanyaan_nama</i>	varchar (255)	Nama dari pertanyaan
3.	<i>fk_tipe_pertanyaan_id</i>	int (11)	<i>Foreign Key</i> dari <i>tipe_pertanyaan_id</i> tabel <i>tipe_pertanyaan</i>
4.	<i>fk_subbab_id</i>	int (11)	<i>Foreign Key</i> dari <i>subbab_id</i> tabel <i>subbab</i>

Tabel 3.10 menunjukkan Struktur Tabel Pertanyaan. Pada Tabel Pertanyaan terdapat empat data yang digunakan. Data *pertanyaan\_id* memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data *pertanyaan\_nama* memiliki tipe data varchar (255) dan berfungsi untuk menyimpan pertanyaan. Data *fk\_tipe\_pertanyaan\_id* memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari *tipe\_pertanyaan\_id* tabel *tipe\_pertanyaan*. Data *fk\_subbab\_id* memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari *subbab\_id* tabel *subbab*.

Tabel 3.11 Struktur Tabel Jawaban

No.	Nama Kolom	Tipe Data	Keterangan
1.	jawaban_id	int (11)	<i>Primary Key, Auto Increment</i>
2.	jawaban_nama	varchar (255)	Nama dari jawaban
3.	jawaban_status	enum ( <i>'true', 'false'</i> )	Status dari jawaban
4.	fk_pertanyaan_id	int (11)	<i>Foreign Key</i> dari pertanyaan_id tabel pertanyaan

Tabel 3.11 menunjukkan Struktur Tabel Jawaban. Pada Tabel Jawaban terdapat empat data yang digunakan. Data jawaban\_id memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data jawaban\_nama memiliki tipe data varchar (255) dan berfungsi untuk menyimpan jawaban. Data jawaban\_status memiliki tipe data enum (*'true', 'false'*) dan berfungsi untuk menyimpan status dari jawaban. Data fk\_pertanyaan\_id memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari pertanyaan\_id tabel pertanyaan.

Tabel 3.12 Struktur Tabel Pengguna

No.	Nama Kolom	Tipe Data	Keterangan
1.	pengguna_id	int (11)	<i>Primary Key, Auto Increment</i>
2.	pengguna_nama	varchar (255)	Nama dari pengguna
3.	pengguna_email	varchar (255)	<i>E-mail</i> dari pengguna
4.	pengguna_password	varchar (255)	<i>Password</i> dari pengguna
5.	pengguna_password_salt	varchar (255)	<i>Salt</i> dari <i>password</i> pengguna
6.	pengguna_tanggal_bergabung	datetime	Tanggal pengguna bergabung
7.	pengguna_login_terakhir	datetime	Tanggal pengguna <i>login</i> terakhir
8.	fk_pengguna_gambar	int (11)	<i>Foreign Key</i> dari penghargaan_id tabel penghargaan
9.	fk_pengguna_bahasa	int (11)	<i>Foreign Key</i> dari bahasa_id tabel bahasa

Tabel 3.12 menunjukkan Struktur Tabel Pengguna. Pada Tabel Pengguna terdapat sembilan data yang digunakan. Data pengguna\_id memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data

pengguna\_nama memiliki tipe data varchar (255) dan berfungsi untuk menyimpan nama dari pengguna. Data pengguna\_email memiliki tipe data varchar (255) dan berfungsi untuk menyimpan *e-mail* dari pengguna. Data pengguna\_password memiliki tipe data varchar (255) dan berfungsi untuk menyimpan *password* dari pengguna. Data pengguna\_password\_salt memiliki tipe data varchar (255) dan berfungsi untuk menyimpan *salt* dari *password* pengguna.

Data pengguna\_tanggal\_bergabung memiliki tipe data datetime dan berfungsi untuk menyimpan tanggal pengguna membuat ID. Data pengguna\_login\_terakhir memiliki tipe data datetime dan berfungsi untuk menyimpan tanggal pengguna terakhir melakukan *login*. Data fk\_pengguna\_gambar memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari penghargaan\_id tabel penghargaan. Data fk\_bahasa\_id memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari bahasa\_id tabel bahasa.

Tabel 3.13 Struktur Tabel Progress Subbab Pengguna

No.	Nama Kolom	Tipe Data	Keterangan
1.	progress_subbab_id	int (11)	<i>Primary Key, Auto Increment</i>
2.	progress_persen	int (11)	Persentase progres dari pengguna
3.	fk_subbab_id	int (11)	<i>Foreign Key</i> dari subbab_id tabel subbab
4.	fk_pengguna_id	int (11)	<i>Foreign Key</i> dari pengguna_id tabel pengguna

Tabel 3.13 menunjukkan Struktur Tabel Progress Subbab Pengguna. Pada Tabel Progress Subbab Pengguna terdapat empat data yang digunakan. Data progress\_subbab\_id memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data progress\_persen memiliki tipe data int (11) dan berfungsi untuk menyimpan persentase progress dari pengguna. Data fk\_subbab\_id memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari subbab\_id tabel

subbab. Data *fk\_pengguna\_id* memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari *pengguna\_id* tabel pengguna.

Tabel 3.14 Struktur Tabel Poin Pengguna

No.	Nama Kolom	Tipe Data	Keterangan
1.	<i>poin_pengguna_id</i>	int (11)	<i>Primary Key, Auto Increment</i>
2.	<i>poin_pengguna_poin</i>	int (11)	Poin dari pengguna
3.	<i>fk_pengguna_id</i>	int (11)	<i>Foreign Key</i> dari <i>pengguna_id</i> tabel pengguna
4.	<i>fk_subbab_id</i>	int (11)	<i>Foreign Key</i> dari <i>subbab_id</i> tabel subbab

Tabel 3.14 menunjukkan Struktur Tabel Poin Pengguna. Pada Tabel Poin Pengguna terdapat empat data yang digunakan. Data *poin\_pengguna\_id* memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data *poin\_pengguna\_poin* memiliki tipe data int (11) dan berfungsi untuk menyimpan poin pengguna. Data *fk\_pengguna\_id* memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari *pengguna\_id* tabel pengguna. Data *fk\_subbab\_id* memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari *subbab\_id* tabel subbab.

Tabel 3.15 Struktur Tabel Riwayat Pertarungan

No.	Nama Kolom	Tipe Data	Keterangan
1.	<i>riwayat_pertarungan_id</i>	int (11)	<i>Primary Key, Auto Increment</i>
2.	<i>fk_pengguna_id</i>	int (11)	<i>Foreign Key</i> dari <i>pengguna_id</i> tabel pengguna
3.	<i>pengguna_poin</i>	int (11)	Poin dari pengguna
4.	<i>fk_lawan_id</i>	int (11)	<i>Foreign Key</i> dari <i>pengguna_id</i> tabel pengguna
5.	<i>lawan_poin</i>	int (11)	Poin dari lawan
6.	<i>pertanyaan</i>	varchar (255)	Pertanyaan pertarungan
7.	<i>riwayat_pertarungan_waktu</i>	datetime	Waktu pertarungan

Tabel 3.15 menunjukkan Struktur Tabel Riwayat Pertarungan. Pada Tabel Riwayat Pertarungan terdapat tujuh data yang digunakan. Data

riwayat\_pertarungan\_id memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data fk\_pengguna\_id memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari pengguna\_id tabel pengguna. Data pengguna\_poin memiliki tipe data int (11) dan berfungsi untuk menyimpan poin pengguna. Data fk\_lawan\_id memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari pengguna\_id tabel pengguna. Data lawan\_poin memiliki tipe data int (11) dan berfungsi untuk menyimpan poin lawan. Data pertanyaan memiliki tipe data varchar (255) dan berfungsi untuk menyimpan pertanyaan yang digunakan untuk pertarungan. Data riwayat\_pertarungan\_waktu memiliki tipe data datetime dan berfungsi untuk menyimpan waktu pertarungan.

Tabel 3.16 Struktur Tabel Status

No.	Nama Kolom	Tipe Data	Keterangan
1.	status_id	int (11)	<i>Primary Key, Auto Increment</i>
2.	status_nama	varchar (255)	Nama status
3.	status_level_minimal	int (11)	Level minimal yang dibutuhkan untuk mendapatkan status
4.	fk_penghargaan_id	int (11)	<i>Foreign Key</i> dari tabel penghargaan

Tabel 3.16 menunjukkan Struktur Tabel Status. Pada Tabel Status terdapat empat data yang digunakan. Data status\_id memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data status\_nama memiliki tipe data varchar (255) dan berfungsi menyimpan nama status. Data status\_level\_minimal memiliki tipe data int (11) dan berfungsi menyimpan level minimal yang dibutuhkan untuk mendapatkan status. Data fk\_penghargaan\_id memiliki tipe data int (11) dan berfungsi sebagai *foreign key* dari penghargaan\_id tabel penghargaan.

Tabel 3.17 Struktur Tabel Administrator

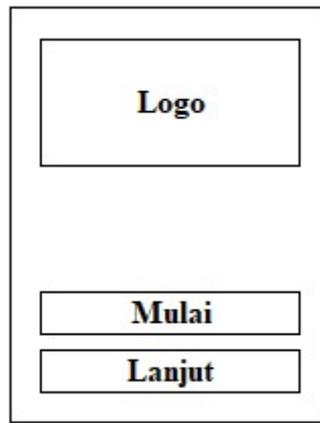
No.	Nama Kolom	Tipe Data	Keterangan
1.	administrator_id	int (11)	<i>Primary Key, Auto Increment</i>
2.	administrator_email	varchar (255)	<i>E-mail</i> dari administrator
3.	administrator_password	varchar (255)	<i>Password</i> dari administrator
4.	administrator_password_salt	varchar (255)	<i>Password salt</i> dari administrator
5.	administrator_login_terakhir	timestamp	Waktu <i>login</i> dari administrator

Tabel 3.17 menunjukkan Struktur Tabel Administrator. Pada Tabel Administrator terdapat empat data yang digunakan. Data administrator\_id memiliki tipe data int (11) dan berfungsi sebagai *primary key* yang bersifat *auto increment*. Data administrator\_email memiliki tipe data varchar (255) dan berfungsi untuk menyimpan *e-mail* dari administrator. Data administrator\_password memiliki tipe data varchar (255) dan berfungsi untuk menyimpan *password* dari administrator. Data administrator\_password\_salt memiliki tipe data varchar (255) dan berfungsi untuk menyimpan *password salt* dari administrator. Data administrator\_login\_terakhir memiliki tipe data timestamp dan berfungsi untuk menyimpan waktu *login* terakhir dari administrator.

### 3.10.6 Desain Antarmuka

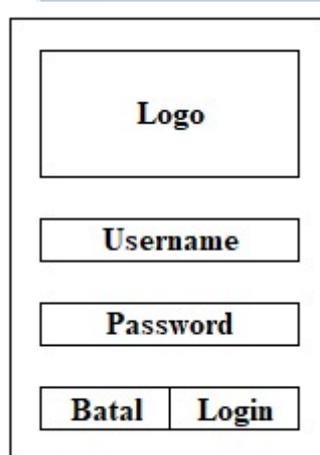
Desain antarmuka dibuat untuk menggambarkan tampilan aplikasi dan CMS pada perancangan penelitian ini.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.17 Desain Antarmuka Halaman Awal

Gambar 3.17 menunjukkan desain antarmuka pada halaman awal. Pada halaman ini, pengguna diminta untuk memilih, apakah pengguna akan memilih mulai untuk melakukan *login* ke dalam aplikasi atau pengguna dapat menekan tombol lanjut untuk pindah ke halaman pendaftaran.



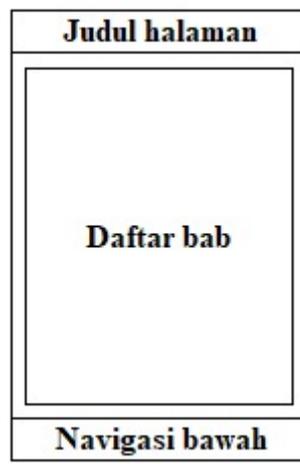
Gambar 3.18 Desain Antarmuka Halaman Login

Gambar 3.18 menunjukkan desain antarmuka pada halaman login. Pada halaman ini, pengguna diminta untuk melakukan *login* untuk bisa masuk ke dalam halaman utama. Pengguna hanya dapat melakukan login dengan menggunakan *e-mail* dan *password* yang sudah terdaftar.

<p><b>Ucapan selamat datang</b></p> <p><b>Batal</b>   <b>Lanjut</b></p>	<p><b>Pilih bahasa</b></p> <p><b>Batal</b>   <b>Lanjut</b></p>
<p><b>Masukan data diri</b></p> <p><b>Nama</b></p> <p><b>E-mail</b></p> <p><b>Password</b></p> <p><b>Batal</b>   <b>Lanjut</b></p>	<p><b>Pendaftaran selesai</b></p> <p><b>Lanjut</b></p>

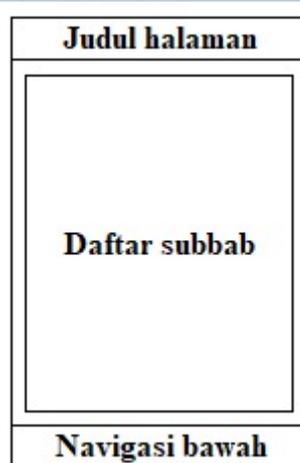
Gambar 3.19 Desain Antarmuka Halaman Pendaftaran

Setelah pengguna menekan tombol mulai, aplikasi akan menunjukkan tampilan pendaftaran. Gambar 3.19 menunjukkan desain antarmuka halaman pendaftaran. Pada halaman pendaftaran, pengguna harus menggunakan *e-mail* yang belum terdaftar pada sistem. Setelah memilih bahasa, pengguna diharuskan untuk mengisi data pengguna terlebih dahulu. Setelah mengisi data pengguna, pengguna dapat menekan tombol lanjut untuk melakukan pendaftaran atau menekan tombol batal untuk kembali ke halaman awal.



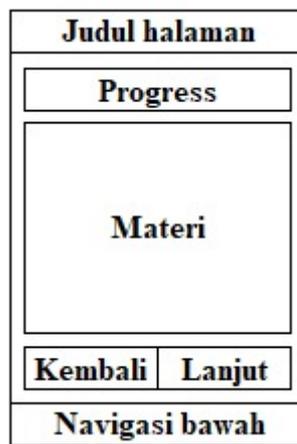
Gambar 3.20 Desain Antarmuka Halaman Daftar Bab

Setelah pengguna memiliki bahasa yang ingin dipelajari, aplikasi akan menampilkan halaman yang berisi daftar bab. Gambar 3.20 menunjukkan desain antarmuka halaman daftar bab. Pada halaman ini, pengguna dapat memilih bab yang ingin dipelajari dari daftar bab yang ditampilkan.



Gambar 3.21 Desain Antarmuka Halaman Daftar Subbab

Gambar 3.21 menunjukkan desain antarmuka halaman daftar subbab. Pada halaman ini, pengguna dapat memilih subbab yang ingin dipelajari dari daftar subbab yang ditampilkan. Pada halaman ini, pengguna dapat memilih apakah pengguna ingin mempelajari kata, kalimat, atau mengerjakan latihan dari setiap subbab.



Gambar 3.22 Desain Antarmuka Halaman Materi

Gambar 3.22 menunjukkan desain antarmuka halaman materi. Pada halaman ini, pengguna dapat melihat materi pada bagian materi, melihat progres pada bagian progres dan maju untuk melihat materi selanjutnya atau mundur untuk melihat materi sebelumnya.



Gambar 3.23 Desain Antarmuka Halaman Latihan

Gambar 3.23 menunjukkan desain antarmuka halaman latihan. Pada halaman ini, pengguna dapat memulai mengerjakan latihan yang ada pada bagian pertanyaan dan jawaban. Pada bagian progress menampilkan progress pengerjaan latihan yang dilakukan pengguna. Jika pengguna menekan kembali, maka pengguna akan membatalkan latihan.

Judul halaman	
Menang	Seri
Kalah	W/L
Riwayat	
Undangan	
Bahasa	
Mulai	
Navigasi bawah	

Gambar 3.24 Desain Antarmuka Halaman Awal Bertarung

Gambar 3.24 menunjukan desain antarmuka halaman awal bertarung. Pada halaman ini, pengguna dapat melihat jumlah menang, kalah, dan seri pertarungan dan persentase kemenangan. Pada bagian bahasa menampilkan bahasa *default* dari pengguna, bahasa dapat diganti oleh pengguna sebelum memulai bertarung atau menekan tombol bertarung. Pengguna juga dapat melihat undangan pertarungan yang dikirimkan oleh pengguna lain dengan menekan bagian undangan pertarungan. Pada bagian riwayat pertarungan akan menampilkan riwayat pertarungan.

Judul halaman
Riwayat Pertarungan
Navigasi bawah

Gambar 3.25 Desain Antarmuka Halaman Riwayat Pertarungan

Gambar 3.25 menunjukan desain antarmuka halaman riwayat pertarungan. Pada halaman ini, pengguna dapat melihat riwayat pertarungan yang telah dilakukan oleh pengguna.



Gambar 3.26 Desain Antamuka Halaman Undangan Pertarungan

Gambar 3.26 menunjukkan desain antamuka halaman undangan pertarungan. Pada bagian daftar undangan pertarungan memperlihatkan daftar undangan pertarungan milik pengguna yang dikirimkan oleh pengguna lain.



Gambar 3.27 Desain Antarmuka Halaman Mulai Pertarungan

Gambar 3.27 menunjukkan desain antarmuka halaman mulai pertarungan. Pengguna merupakan nama pengguna yang membuka aplikasi, sedangkan lawan merupakan nama pengguna yang menjadi lawan. Pada halaman ini pengguna dapat memilih antara memulai atau menyerah. Pengguna akan memulai pertarungan dengan menekan tombol mulai. Pengguna akan menyerah dan dianggap kalah jika menekan tombol menyerah.



Gambar 3.28 Desain Antarmuka Halaman Soal

Gambar 3.28 menunjukkan desain antarmuka halaman soal. Pada halaman ini, pengguna dapat memulai mengerjakan latihan yang ada pada bagian pertanyaan dan jawaban. Pada bagian progress menampilkan progress pengerjaan latihan yang dilakukan pengguna. Jika pengguna menekan kembali, maka pengguna akan dianggap menyerah dan kalah.



Gambar 3.29 Desain Antarmuka Halaman Peringkat

Gambar 3.29 menunjukkan desain antarmuka halaman peringkat. Daftar peringkat akan memperlihatkan sepuluh pengguna dengan peringkat level tertinggi. Info yang ditampilkan berupa gambar, nama, *level*, *status*, dan *point* dari pengguna yang masuk ke dalam daftar sepuluh pengguna dengan *level* tertinggi.

<b>Judul halaman</b>	
<b>Status</b>	
<b>Gambar</b>	<b>Nama</b>
	<b>Level</b>
	<b>EXP</b>
<b>Daftar Penghargaan</b>	
<b>Navigasi bawah</b>	

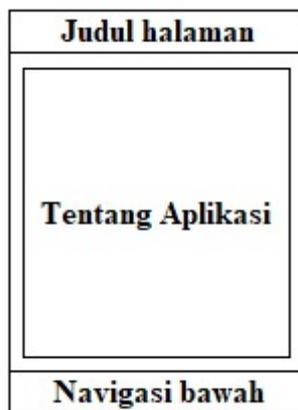
Gambar 3.30 Desain Antarmuka Halaman Profil

Gambar 3.30 menunjukkan desain antarmuka halaman profil. Pada halaman ini, pengguna dapat melihat informasi nama pengguna, gambar pengguna, dan *level* pengguna. Selain itu, pada halaman ini pengguna juga dapat melihat daftar penghargaan yang dimilikinya dan yang belum dimilikinya. Pengguna dapat mengganti gambar pengguna dengan memilih dari gambar yang telah dia dapatkan.

<b>Judul halaman</b>	
<b>Pelajaran utama</b>	
<b>Tentang aplikasi</b>	
<b>Logout</b>	
<b>Navigasi bawah</b>	

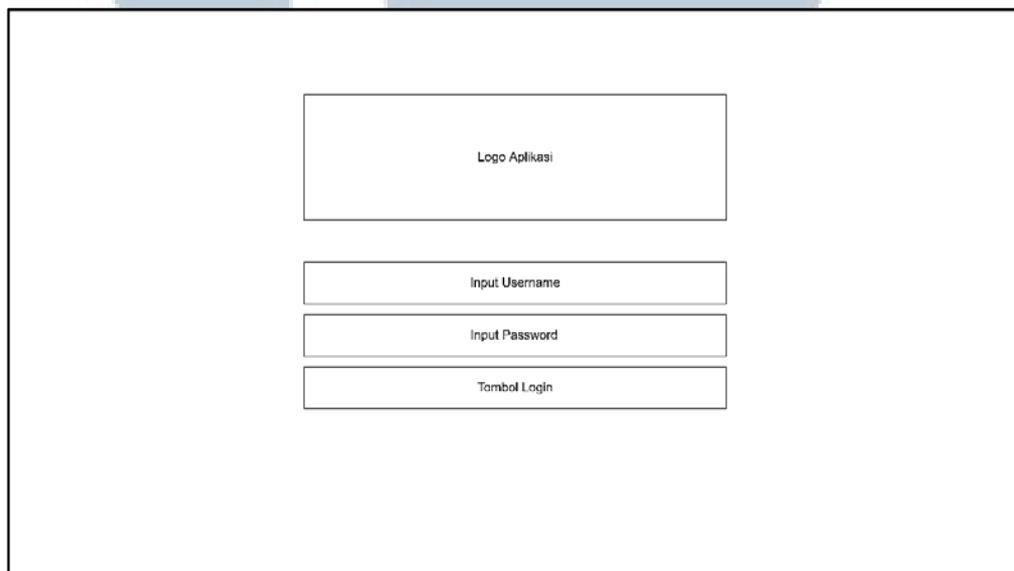
Gambar 3.31 Desain Antarmuka Halaman Pengaturan

Gambar 3.31 menunjukkan desain antarmuka halaman pengaturan. Pada bagian pelajaran utama terdapat *list* yang terdiri dari mandarin atau jepang, pelajaran utama akan ditampilkan pada awal halaman belajar dan bertarung. Pada bagian tentang aplikasi akan menampilkan halaman tentang aplikasi. Pada bagian *logout* akan melakukan fungsi *logout*.



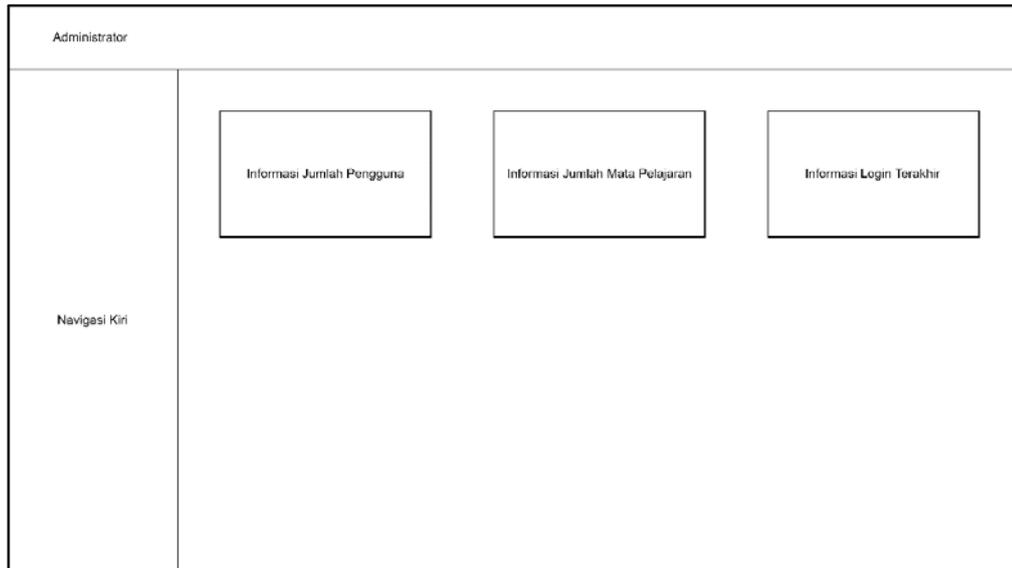
Gambar 3.32 Desain Antarmuka Halaman Tentang Aplikasi

Gambar 3.32 menampilkan halaman tentang aplikasi, pada halaman ini pengguna bisa melihat versi dari aplikasi, nama pembuat, dan nama pendukung proses pembuatan aplikasi.



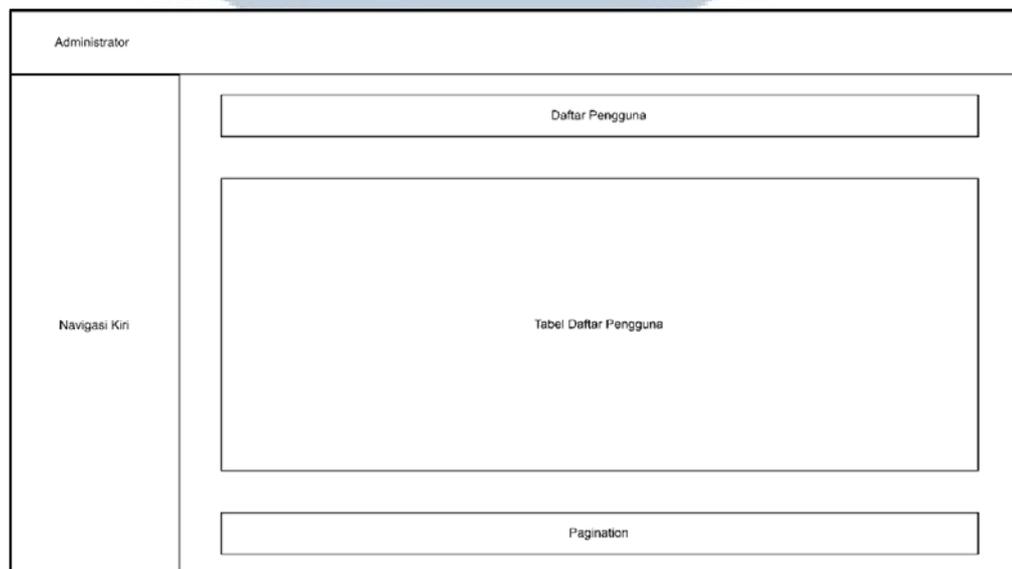
Gambar 3.33 Desain Antarmuka CMS Halaman Login

Gambar 3.33 menunjukan desain antarmuka halaman *login* untuk CMS. Pada halaman ini administrator diminta untuk melakukan *login* agar bisa memasuki halaman utama CMS.



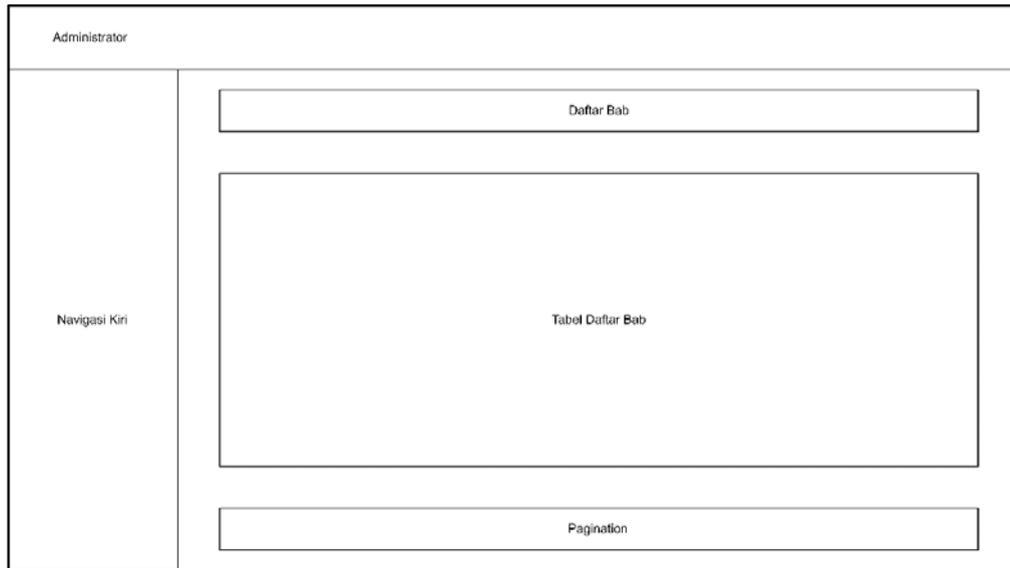
Gambar 3.34 Desain Antarmuka CMS Halaman Awal

Setelah administrator melakukan *login*, CMS akan menunjukkan halaman awal seperti pada Gambar 3.34. Pada halaman awal, terdapat informasi jumlah pengguna yang berisi jumlah keseluruhan pengguna aplikasi.



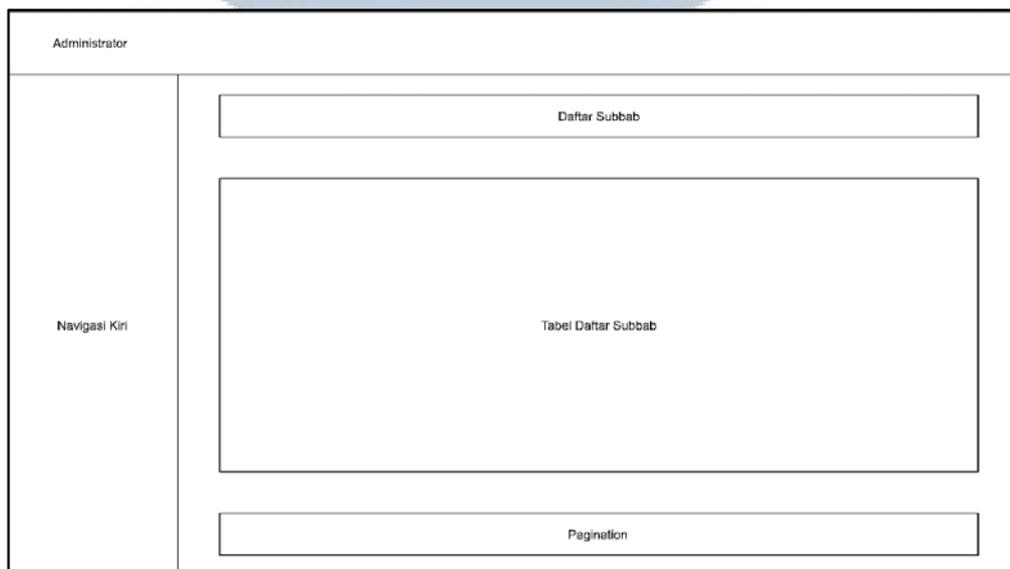
Gambar 3.35 Desain Antarmuka CMS Halaman Daftar Pengguna

Gambar 3.35 menunjukkan desain antarmuka halaman daftar pengguna untuk CMS. Pada halaman ini administrator bisa melihat daftar dari pengguna aplikasi pada tabel daftar pengguna.



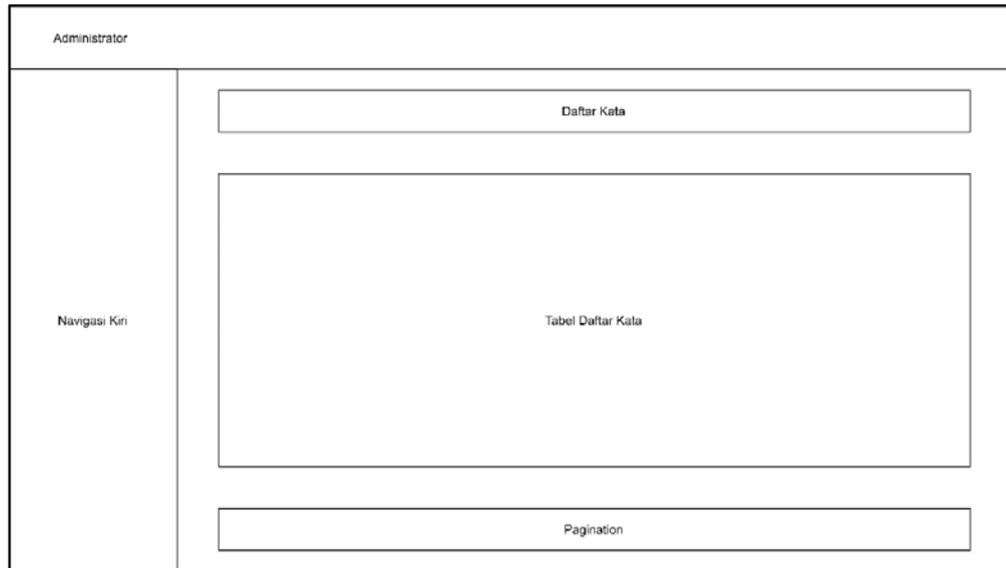
Gambar 3.36 Desain Antarmuka CMS Halaman Daftar Bab

Gambar 3.36 menunjukkan desain antarmuka halaman daftar bab untuk CMS. Pada halaman ini administrator bisa melihat, menambahkan, mengubah, atau menghapus bab yang ada di *database*.



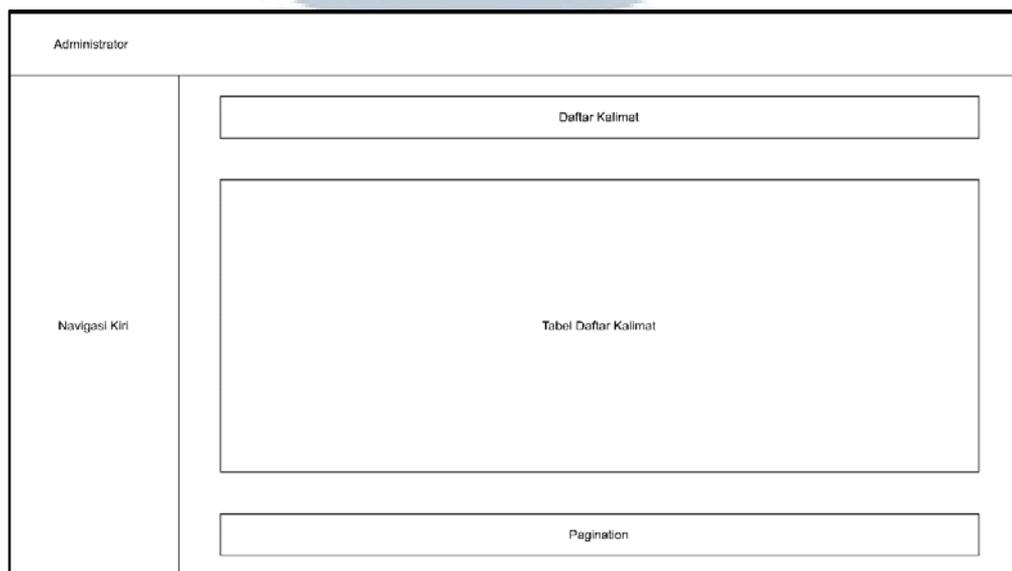
Gambar 3.37 Desain Antarmuka CMS Halaman Daftar Subbab

Gambar 3.37 menunjukkan desain antarmuka halaman subbab untuk CMS. Pada halaman ini administrator bisa melihat, menambahkan, mengubah, atau menghapus subbab yang ada di *database*.



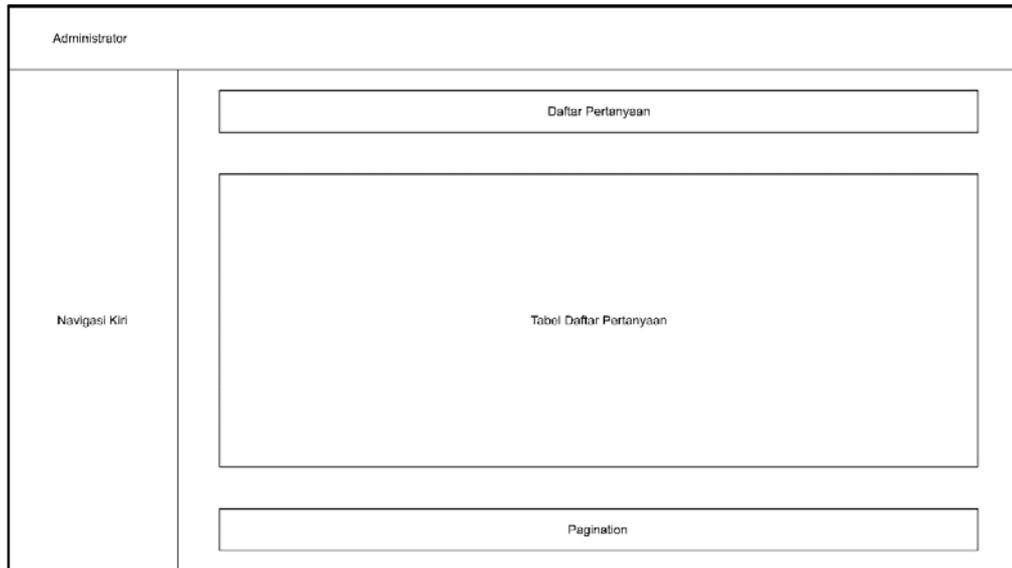
Gambar 3.38 Desain Antarmuka CMS Halaman Daftar Kata

Gambar 3.38 menunjukkan desain antarmuka halaman daftar kata untuk CMS. Pada halaman ini administrator bisa melihat, menambahkan, mengubah, atau menghapus kata yang ada di *database*.



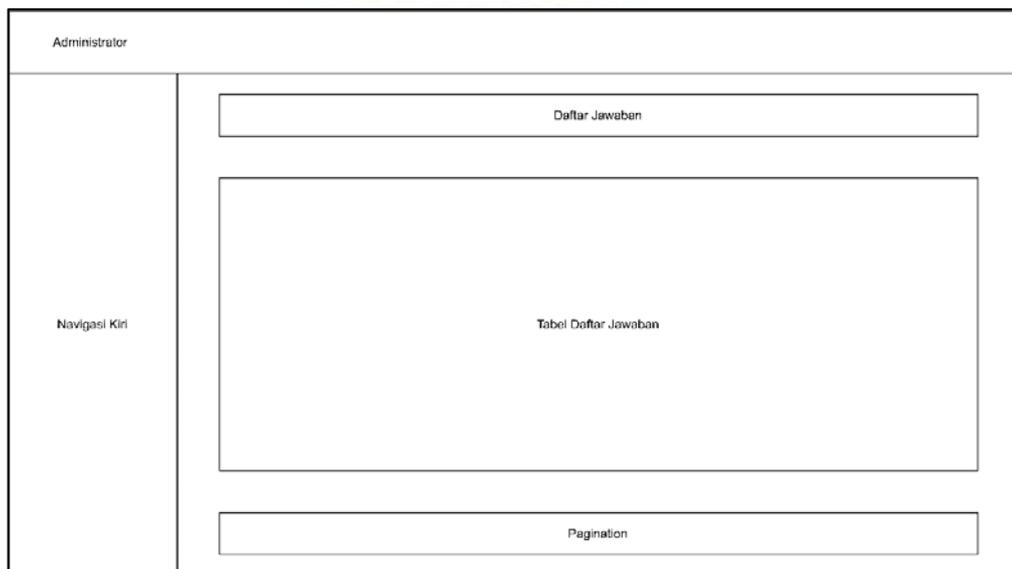
Gambar 3.39 Desain Antarmuka CMS Halaman Daftar Kalimat

Gambar 3.39 menunjukkan desain antarmuka halaman daftar kalimat untuk CMS. Pada halaman ini administrator bisa melihat, menambahkan, mengubah, atau menghapus kalimat yang ada di *database*.



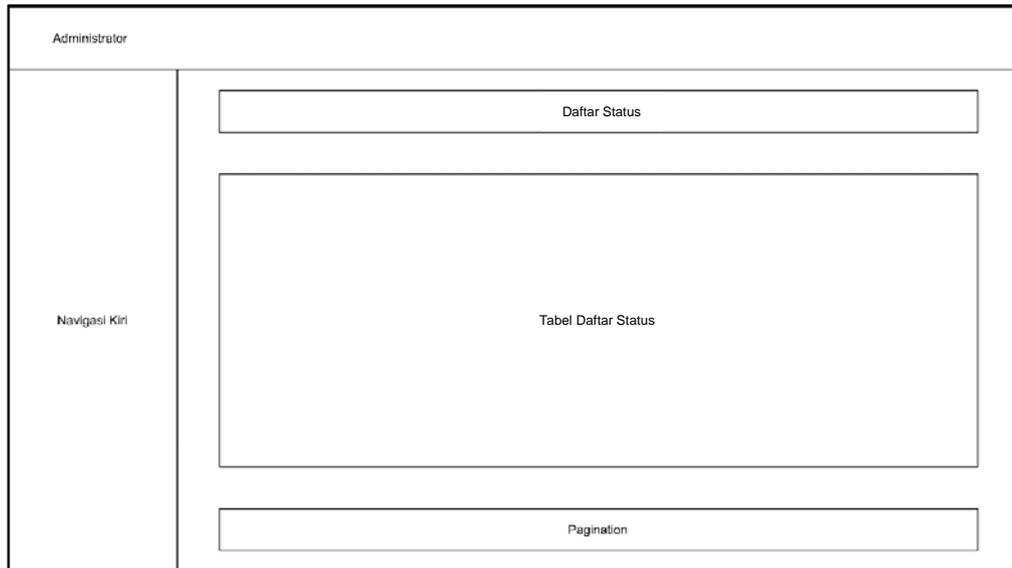
Gambar 3.40 Desain Antarmuka CMS Halaman Daftar Pertanyaan

Gambar 3.40 menunjukkan desain antarmuka halaman daftar pertanyaan untuk CMS. Pada halaman ini administrator bisa melihat, menambahkan, mengubah, atau menghapus kalimat yang ada di *database*.



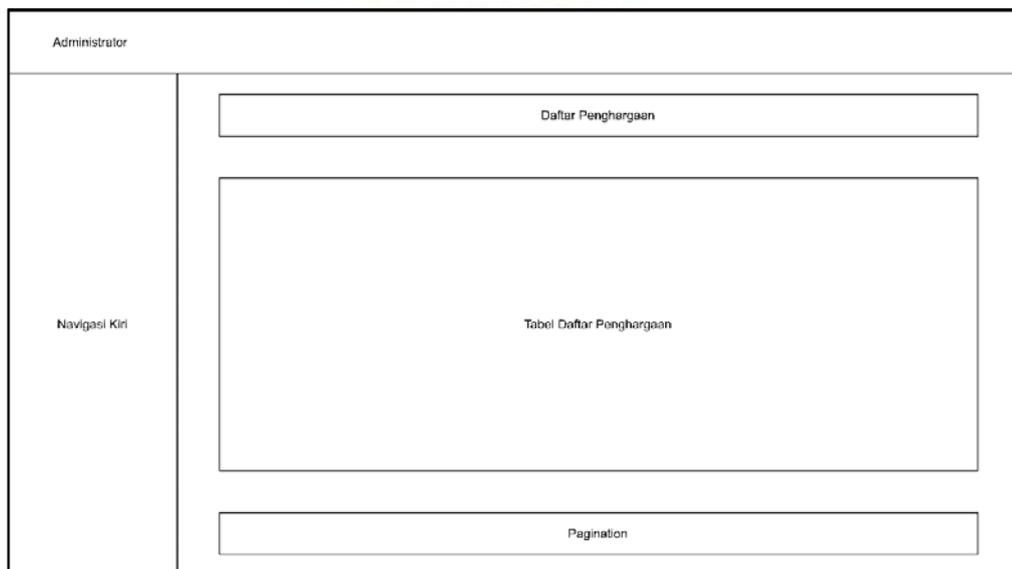
Gambar 3.41 Desain Antarmuka CMS Halaman Daftar Jawaban

Gambar 3.41 menunjukkan desain antarmuka halaman daftar jawaban untuk CMS. Pada halaman ini administrator bisa melihat, menambahkan, mengubah, atau menghapus jawaban yang ada di *database*.



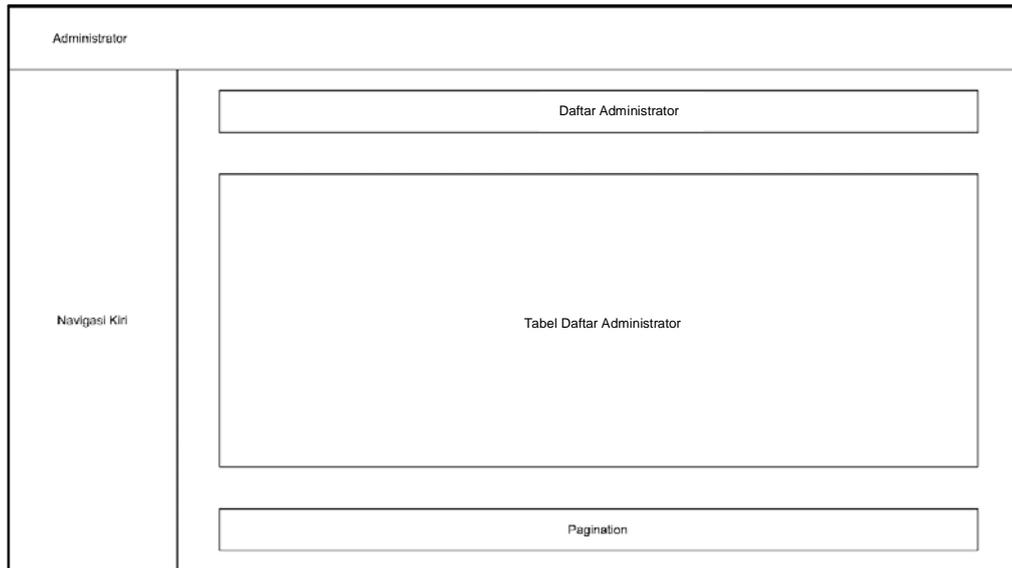
Gambar 3.42 Desain Antarmuka CMS Halaman Daftar Status

Gambar 3.42 menunjukkan desain antarmuka halaman daftar status untuk CMS. Pada halaman ini administrator bisa melihat, menambahkan, mengubah, atau menghapus status yang ada di *database*.



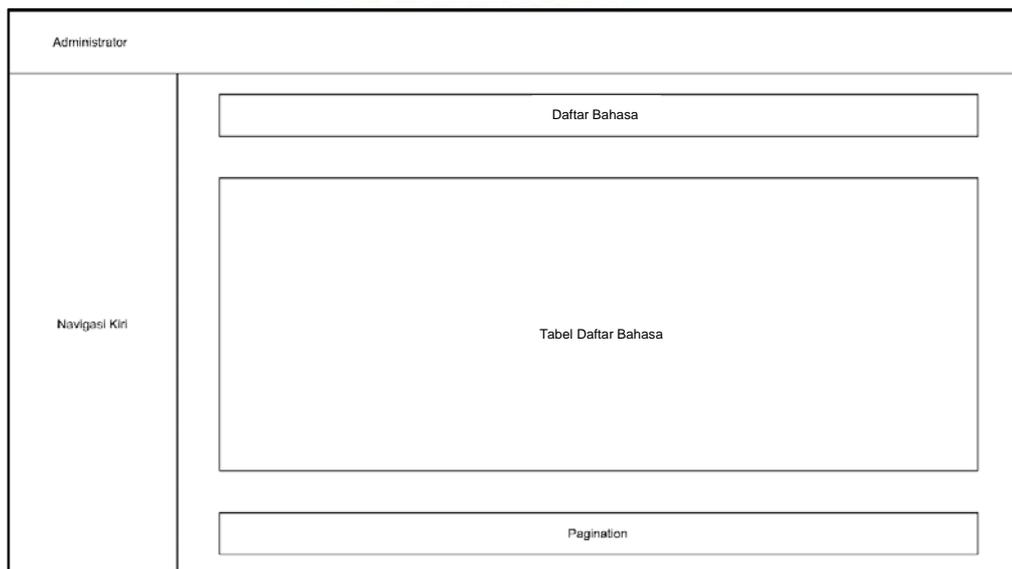
Gambar 3.43 Desain Antarmuka CMS Halaman Daftar Penghargaan

Gambar 3.43 menunjukkan desain antarmuka halaman daftar penghargaan untuk CMS. Pada halaman ini administrator bisa melihat, menambahkan, mengubah, atau menghapus penghargaan yang ada di *database*.



Gambar 3.44 Desain Antarmuka CMS Halaman Daftar Administrator

Gambar 3.44 menunjukkan desain antarmuka halaman daftar administrator untuk CMS. Pada halaman ini administrator bisa melihat, menambahkan, atau menghapus administrator yang ada di *database*.



Gambar 3.45 Desain Antarmuka CMS Halaman Daftar Bahasa

Gambar 3.45 menunjukkan desain antarmuka halaman daftar bahasa untuk CMS. Pada halaman ini administrator bisa melihat daftar bahasa yang ada di *database*.