# BAB III

**ANALISIS DAN PERANCANGAN**

## 3.1 Analisis

Dari beberapa permasalahan yang telah dipaparkan pada latar belakang, yang menjadi fokus peneliti adalah permasalahan pada kompleksitas tata bahasa (*Nahwu*) dan bentuk kata (*Sharaf*) Bahasa Arab. Kedua elemen linguistik inilah yang sering kali menjadi tantangan utama bagi pemelajar terlebih kalangan yang awam. Padahal, kemampuan fundamental ini sangatlah penting untuk memahami struktur suatu frasa dan kalimat.

Tantangan ini dibenarkan pula oleh institusi layanan bahasa asing di Amerika, yakni *The United States Foreign Service Institution* (FSI). Institusi ini menempatkan tingkat kompleksitas struktur Bahasa Arab sejajar dengan Bahasa

Mandarin, Korea, dan Jepang (Fatma Al-Busaidi, 2015).

### 3.1.1 Identifikasi Permasalahan

Dalam penguasaan bahasa terdapat permasalahan yang umum yang terjadi. Nurkholis dalam jurnal miliknya mengutip dari buku yang ditulis oleh Nanik Setyawati mengatakan klausul ini sebagai berikut. Kesalahan yang terjadi cenderung karena pengguna bahasa, bukan sistem bahasa yang ia gunakan. Lebih lanjut, dalam jurnal itu dikelompokkan dua jenis kesalahan. Pengelompokan tersebut yakni, pengaruh bahasa yang lebih dahulu dipelajari, dan pemahaman pemelajar bahasa terhadap bahasa yang sedang ia kuasai (Nurkholis, 2018).

Pemaparan di atas sangat relevan ketika disinggungkan dengan perbedaan struktur kalimat antara Bahasa Indonesia, Bahasa Arab, dan Bahasa Inggris.

Secara garis besar, dalam struktur kalimat Bahasa Indonesia dan Bahasa Inggris memilikikesatuan makna semantis. Maksudnya, dalam struktur verbal akan memiliki makna semantis yang verbal pula, lalu pada struktur nominal akan memiliki makna semantis berupa nomina. Sedangkan dalam Bahasa Arab tidak seperti kedua bahasa yang telah disebutkan. Ketika kalimat tersebut memiliki makna semantis nominal tidak selamanya disusun oleh struktur kata nomina sepenuhnya (Hidayatullah, 2017).

### 3.1.2 Pemecahan Masalah

Mengacu pada permasalahan yang telah dipaparkan maka diperlukan akselerasi yang dapat memberikan akselerasi dan keefektifan pemahaman bagi pemelajar pemula linguistik arab klasik. Serta sebagai sistem cerdas yang dapat mendeteksi klasifikasi kelas kata, kekeliruan gramatika, dan memberikan saran perbaikan olah frasa, klausa, dan kalimat dalam Bahasa Arab klasik.

Menurut Rahmad Iswanto, peran teknologi dalam proses pemahaman lingustik Arab yang dimaksudkan adalah cara teknologi akan menjawab permasalahan dalam pemahaman linguistik Arab itu sendiri dan seberapa ia dapat berperan dalam keberlanjutan pemelajaran yang ada di dalamnya dari fundamental hingga tingkat lanjutan menuju mahir (Iswanto, 2017).

### 3.1.3 Kelebihan Sistem yang Diusulkan

Kelebihan dari pemecahan masalah ini adalah:

1. Sistem akan memudahkan pengguna mengidentifikasi kesalahan gramatika dan bentuk kata bahasa Arab.
2. Pada sistem akan terdapat penganalisaan yang dipecah ke dalam sub bagian, yakni tingkat frasa (*Tarki:b*), klausa (*Jumlah*), dan kalimat (*Jumailah*). Sehingga lebih memudahkan pengguna dengan tingkat pemahaman pemula.
3. Sistem akan memberikan koreksi tata letak (kedudukan kata), kelas kata, kasus, gender, jumlah kata.

## 3.2 Perancangan

### 3.2.1 Perancangan Sistem

Tahap pembangunan sistem mencakup ide, konsep program, dan konten program yang semuanya akan menjadi struktur dari program. Pada penelitian ini, penulis akan merancang dan mengimplementasikan sebuah program dengan teknologi yang berkemampuan untuk mendeteksi ketidaktepatan susunan kata dari segi gramatika/tata bahasa dan bentuk kata. Selain untuk mendeteksi kesalahan, program ini juga akan dapat memberikan saran perbaikan susunan dan *reminder* kaidah-kaidah tata bahasa Arab.

Program akan memproses *input*-an *user* dengan mencocokkan bank kata dalam *database* yang telah dikelompokkan berdasarkan jenis-jenis bentuk kata-nya. Kemudian program akan menampilkan hasil sesuai olahan logika *rule* hasil adaptasi dari kaidah-kaidah tata bahasa Arab yang diimplementasikan di dalam program.

#### 3.2.1.1 Alur Aplikasi

Pada bagian navigasi halaman akan terdapat tiga pilihan menu, yakni *Home,* Bantuan, Tentang. Menu *Home* akan berisi konten utama berupa empat pilihan proses analisa, yakni analisa frasa, klausa, dan kalimat.

Ketika pengguna telah menginputkan kata, kemudian program akan memberikan hasil analisa sesuai opsi analisa yang dipilih. Selanjutnya pada menu Bantuan berisi materi dasar tata bahasa Arab, dan terminologi-terminologi asing yang dipakai di dalam program. Sedangkan pada menu Tentang berisi tentang tujuan pengembangan program dan versi program.

#### 3.2.1.2 Gambaran Umum

Secara umum pada aplikasi ini akan dirancang dengan tombol untuk memilih menu yang akan diarahkan sesuai fungsinya, Dalam aplikasi ini akan disediakan 3 menu di antara lain:

1. Menu Home

Ketika pengguna berada pada menu / halaman ini, maka akan tertampil tombol login untuk memulai analisa struktur. Pengguna akan dapat memilih analisa yang dikehendaki setelah melakukan inisisasi login/mendaftarkan akun. Jenis analisa terbagi menjadi analisa frasa, klausa, atau kalimat utuh. Setelah pengguna menginputkan kata sesuai opsi yang di pilih maka akan muncul hasil berupa deteksi dan koreksi setelah analisa diselesaikan oleh program.

1. Bantuan

Berisi tentang daftar terminologi-terminologi bahasa yang akan memudahkan bagi pengguna umum dan pemelajar linguistik Bahasa Arab pemula.

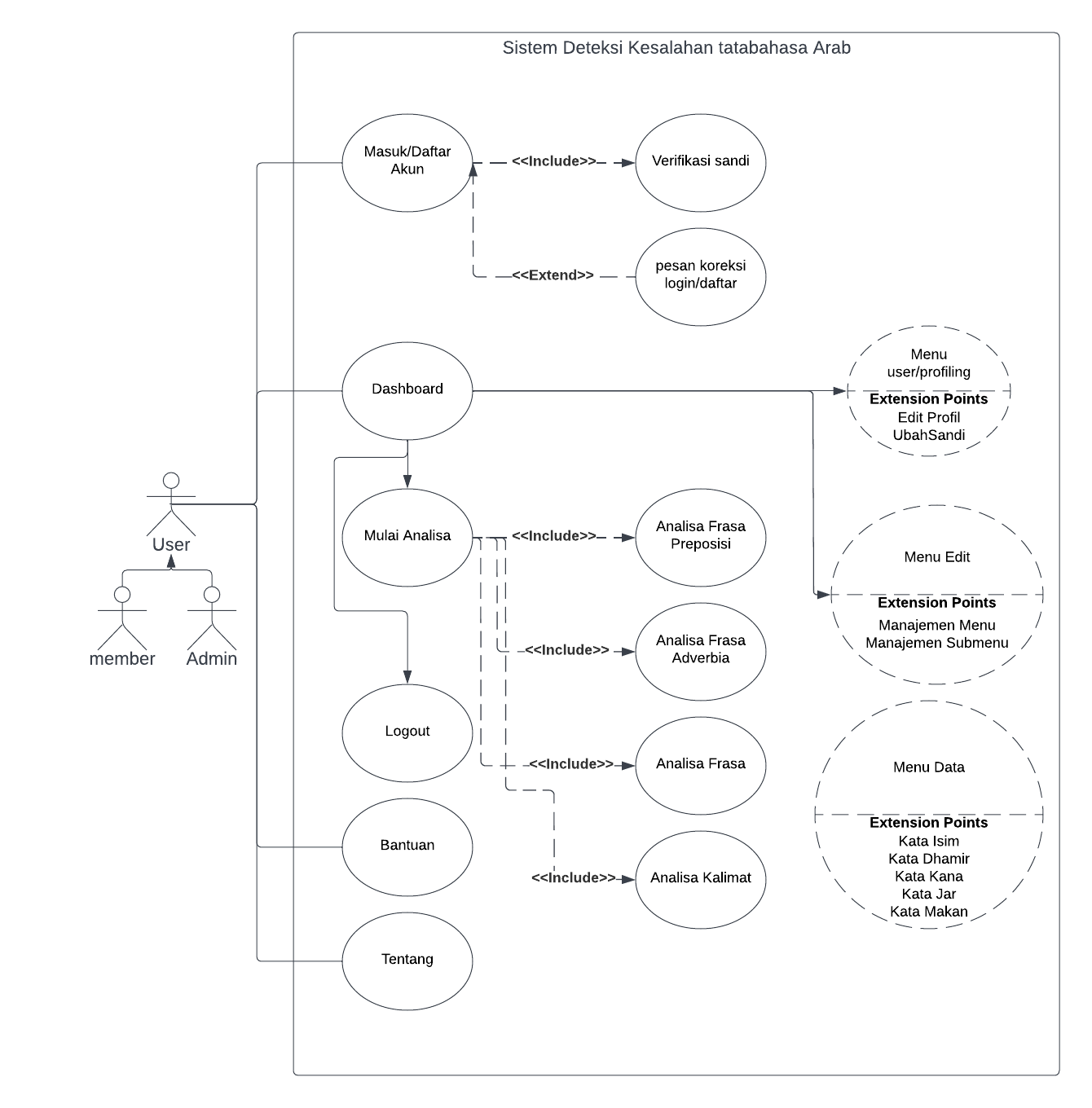
1. Tentang

Berisi tentang latar belakang pembangunan sistem dan versi.

#### 3.2.1.3 Unified Modelling Language Aplikasi

UML dapat juga diartikan sebuah bahasa grafik standar yang digunakan untuk memodelkan perangkat lunak berdasarkan pada objek dalam hal ini UML digunakan untuk menggambarkan alur dari aplikasi.

1. *Use Case* Diagram Aplikasi



**Gambar 3.1** Gambar UML use case diagram sistem

Pada Gambar 3.1 terdapat interaksi yang terjadi pada menu antara lain adalah Menu home yang akan menampilkan sub menu lain yaitu, Analisa kata, analisa frasa, analisa klausa, dan analisa kalimat. Pada tiap-tiap sub menu akan memiliki fungsi sama, yakni menampung inputan pengguna yang nanti akan diproses dan ditampilkan hasilnya tepat pada halaman yang sama.

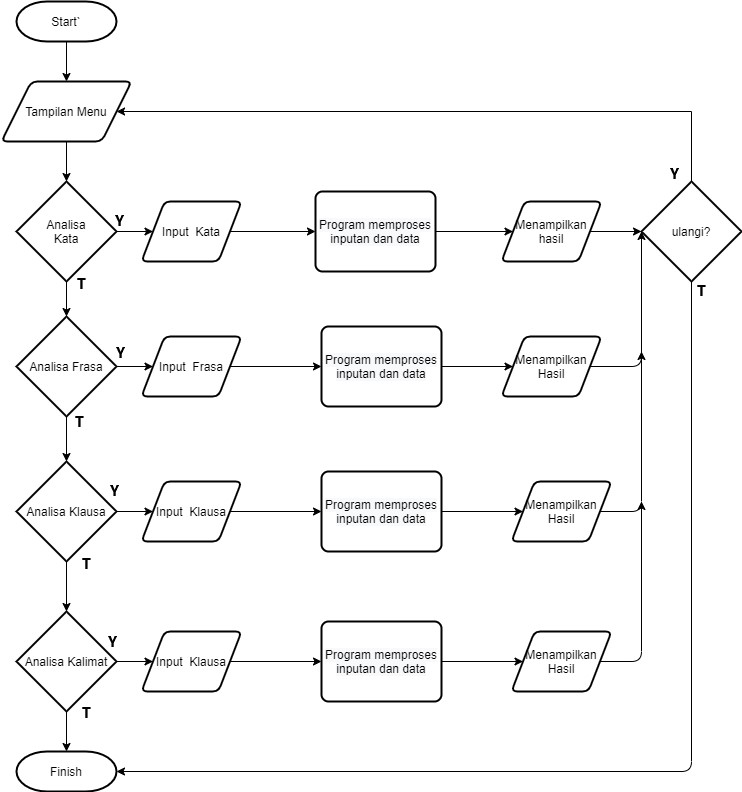
Selanjutnya adalah menu bantuan dan menu tentang. Menu Bantuan berisi dokumentasi cara-cara penggunaan sistem dan pengenalan linguistik dasar beserta indeks terminologi linguistik bagi pengguna. Sedangkan menu Tentang berisi informasi terkait latar belakang pembangunan dan versi sistem.

**Tabel 3.3** Keterangan *Use Case* Diagram *User*

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Keterangan** |
| 1 |  | *User* yang menggunakan aplikasi ini dapat menjalankan program, melakukan analisa kata, frasa, klausa, dan kalimat. Lalu mengakses menu bantuan. Terakhir membaca informasi tentang pengembangan program ini di menu Tentang. |
| 2 | Menu Home | Use case ini berfungsi untuk menampilkan pilihan opsi halaman jenis analisa yang |
|  |  | dikehendaki pengguna. |
| 3 | Analisa  frasa | *Use case* ini memiliki fungsi untuk melakukan analisa frasa. |
| 4 | Analisa  klausa | *Use case* ini memiliki fungsi untuk melakukan analisa klausa. |
| 5 | Analisa  kalimat | *Use case* ini memiliki fungsi untuk melakukan analisa kalimat. |
| 6 | Menu  Bantuan | *Use case* ini memiliki fungsi untuk menampilkan bantuan berupa dokumentasi, cara penggunaan program, pengetahuan fundamental lingustik beserta terminologi-terminologi yang digunakan. |
| 7 | Menu  Tentang | *Use case* ini memiliki fungsi untuk menampilkan informasi latar belakang  pembangunan sistem beserta versinya. |

#### 3.2.1.4 Flowchart Menu

Pada menu utama terdapat empat pilihan analisa yang dapat dipilih oleh pengguna. Pada setiap pilihan telah dibangun dengan memiliki fungsi melakukan analisa sesuai tingkatan dalam suatu susunan kata. Pilihan pertama adalah analisa kata yang dapat menganalisis susunan kata berdasarkan atribut yang menjadi parameter kelas dari kata tersebut. Pilihan selanjutnya adalah analisa frasa yang memungkinkan pengguna untuk mengetahui validitas susunan frasa dan jenisnya. Kemudian pilihan ketiga yaitu analisa klausa yang memungkinkan bagi pengguna untuk mengetahui jenis, kedudukan, dan fungsi dari klausa tersebut. Lalu pilihan yang keempat adalah analisa kalimat yang memungkinkan pengguna untuk mengetahui struktur kalimat dari sudut pandang tata bahasa Arab. Semua pilihan tersebut tentu dibekali dengan kemampuan untuk menampilkan koreksi terhadap struktur yang diakibatkan oleh kesalahan input yang dilakukan oleh pengguna.



**Gambar 3.11** *Flowchart* pilihan analisa

#### 3.2.1.5 *Rule Based*

Pada sistem ini akan diimplementasikan sejumlah *rule* yang akan menjadi acuan identifikasi sistem terhadap kesalahan inputan kata. Berikut ini merupakan rancangan *rule* dalam bentuk *pseudocode* yang terbagi dalam empat rancangan rule. Keempat rancangan ini terbagi ke dalam rancangan *pseudocode*  berikut.

1. **Fungsi Cek kedudukan Kata**

Fungsi ini berfungsi sebagai algoritma yang akan melakukan deteksi kedudukan kata. Pada tahap deteksi ini, rule akan melalukan cek input kata oleh user. Algoritma akan memastikan terlebih dahulu apakah kedudukan kata telah memnuhi syarat/kasus tertentu. Jika benar maka kedudukan kata telah benar. Namun jika belum makan kedudukan kata belum benar.

fungsi cek\_kedudukan\_awal(parameter A, parameter B)

{

If (data\_kata = [nama\_kata = parameter A], [kasus = parameter B]

)

{

return "kedudukan isim awal benar "

} else {

return "kedudukan isim awal salah"

}

}

fungsi cek\_kedudukan\_kedua(parameter A, parameter B)

{

If (data\_kata = [nama\_kata = parameter A], [kasus = parameter B]

)

{

return "kedudukan isim keda benar "

} else {

return "kedudukan isim kedua salah"

}

}

1. **Fungsi Cek Kelas kata**

Pada fungsi ini rule akan memastikan bahwa kata yang telah diinptakn oeh user telah memenuhi syarat tertentu. Algoritma akan memastikan input kata telah sesuai rule yang telah diadaptasi. Setiap kata harus berkasus nominatif, akusatif, atau genetif—yang tentu opsi ini sangat tergantung pada implementasi algoritma pada tingkat struktur kata tertentu pula.

fungsi cek\_kelas\_kata\_awal(parameter A, parameter B)

{

kasus\_kata = "Nominatif (Rafa)/Akusatif (Nashab)/Genetif(Jar)";

if (data\_kata['nama\_kata'] = parameter A && parameter B = kasus\_kata) {

return "kelas kata awal benar";

} else {

return "Kelas kata awal salah";

}

}

fungsi cek\_kelas\_kata\_kedua(parameter A, parameter B)

{

kasus\_kata = "Genetif (Jar)";

if (data\_kata['nama\_kata'] = parameter A && parameter B = kasus\_kata) {

return "kelas kata kedua benar";

} else {

return "Kelas kata kedua salah";

}

}

1. **Fungsi Cek Jumlah Kata**

Fungsi ini akan melakukan cek terhadap keserasian antarkata. Jika parameter pertama telah memiliki jenis jumlah yang sama maka logika algoritma akan bernilai benar. Jika tidak maka akan bernilai belum benar.

fungsi cek\_jumlah\_kata(parameter A, parameter B, parameter N)

{

if(parameter A = data(jumlah: "Jamak") dan parameter B = data(jumlah: "jamak")

{

return "Sama Jamak"

}

elseif (parameter A = data(jumlah: "dual") dan parameter B = data(jumlah: "dual")

{

return "Sama Dual"

}

elseif (parameter A = data(jumlah: "Tunggal") dan parameter B = data(jumlah: "Tunggal")

{

return "Sama Tunggal"

}

else {

return "Jumlah Kata Tidak Sama"

}

1. **Fungsi Cek Gender Kata**

Sama halnya seperti fungsi sebelumnya, fungsi ini akan memeastikan keserasian gender antarkata. Bila kata pada parameter pertama telah memiliki gender yang sama terhadap parameter kedua maka algoritma bernilai benar. Sebaliknya, jika gender kata belum sama makan bernilai salah.

fungsi cek\_gender\_kata(parameter A, parameter B, parameter N)

{

if(parameter A = data(jumlah: "Maskulin") atau parameter A = data(gender: "Maskulin/Feminin" ) dan

(parameter B = data(gender: "Maskulin") atau parameter B = data(gender: "Maskulin/Feminin")

{

return "Sama Maskulin"

}

elseif (parameter A = data(gender: "Feminin") atau parameter A = data(gender: "Maskulin/Feminin" ) dan

(parameter B = data(gender: "Feinin") atau parameter B = data(gender: "Maskulin/Feminin")

{

return "sama Feminin"

} else {

return "Gender Tidak Sama"

}

#### 3.2.1.6 Kebutuhan Teknologi

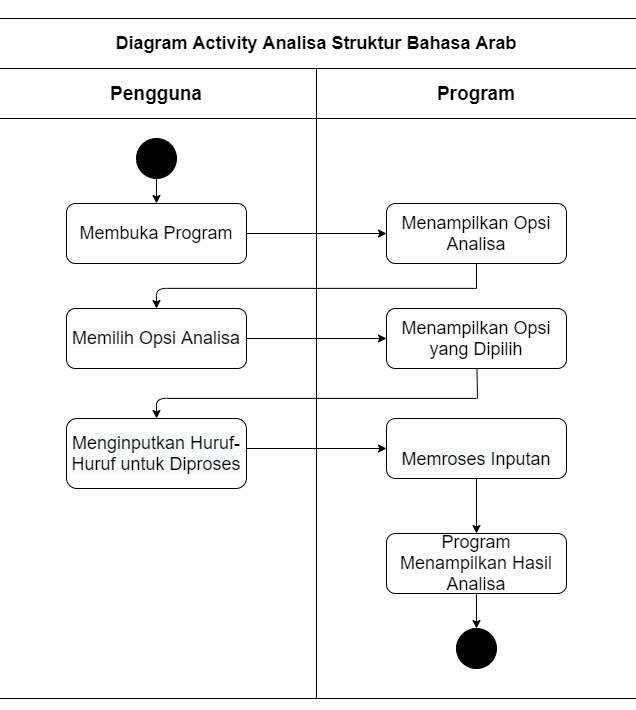
Daftar kebutuhan teknologi baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) untuk membangun aplikasi:

**Tabel 3.2** Kebutuhan *Hardware* dan *Software*

|  |  |  |
| --- | --- | --- |
| **No** | **Kebutuhan** | **Kegunaan** |
| 1. | Perangkat PC/Laptop | Sebagai perangkat yang digunakan untuk merancang program. |
| 2. | Localhost | Sebagai server uji coba |
| 3. | Text Editor | Sebagai perangkat lunak untuk mengkode program |

2.

*Activity Diagram*



**Gambar 3.2** *Activity* Diagram Analisa

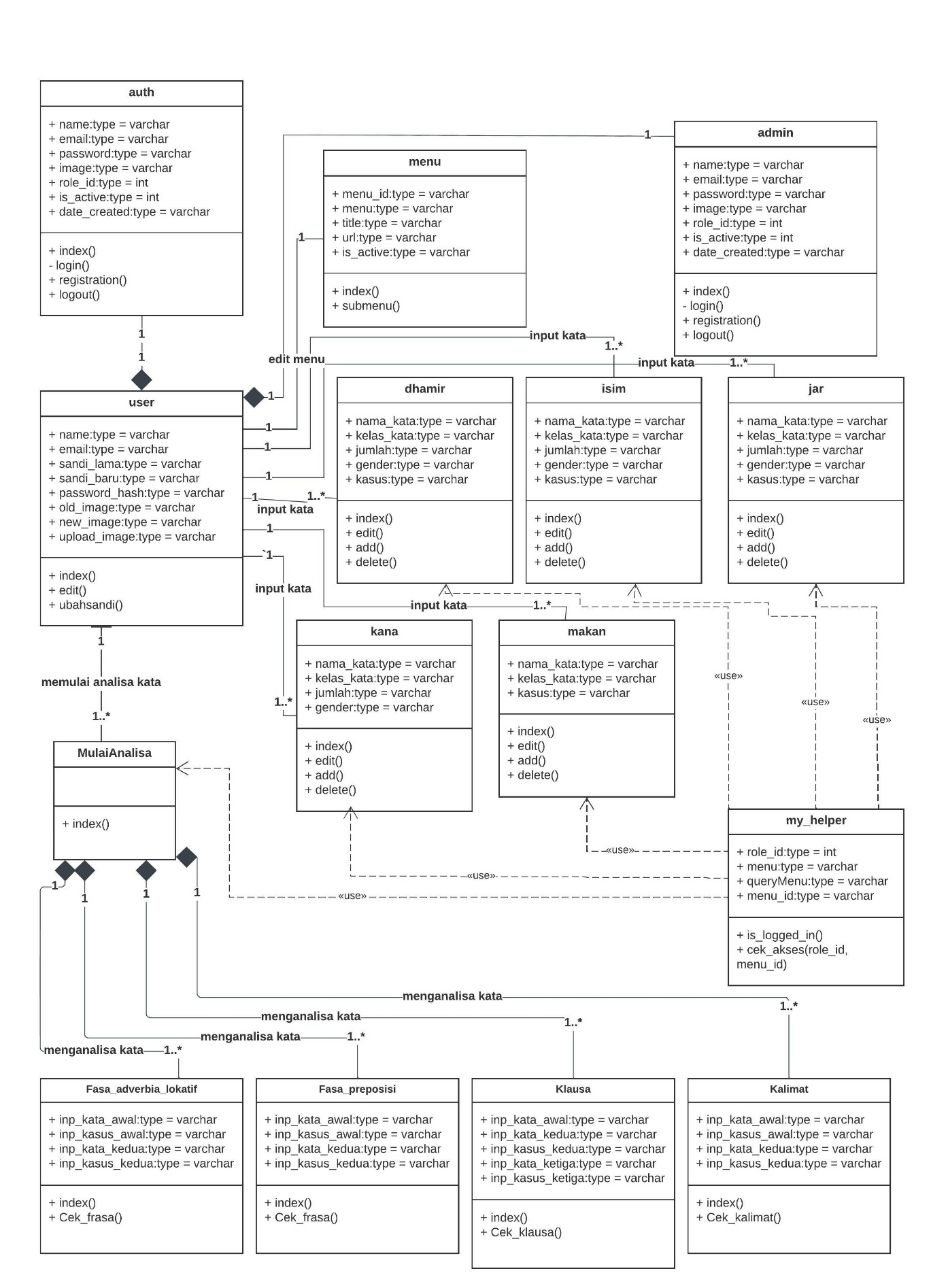
Pada gambar *activity* diagram di atas, alur prosesnya adalah sebagai berikut:

1. *User* mengakses/membuka program
2. Setelah pengguna membuka program maka sistem akan menampilkan opsi analisa yang bisa dipilih.
3. Pengguna memilih opsi analisa struktur yang dikehendaki
4. Sistem akan menampilkan opsi analisa struktur yang dipilih.
5. Pengguna akan melakukan analisa struktur yang diinputkan pada sistem
6. Sistem akan memproses inputan pengguna
7. Sistem akan menampilkan hasil analisa kepada pengguna.

### 3.2.2 Perancangan Data

#### 3.2.2.1 *Class* Diagram

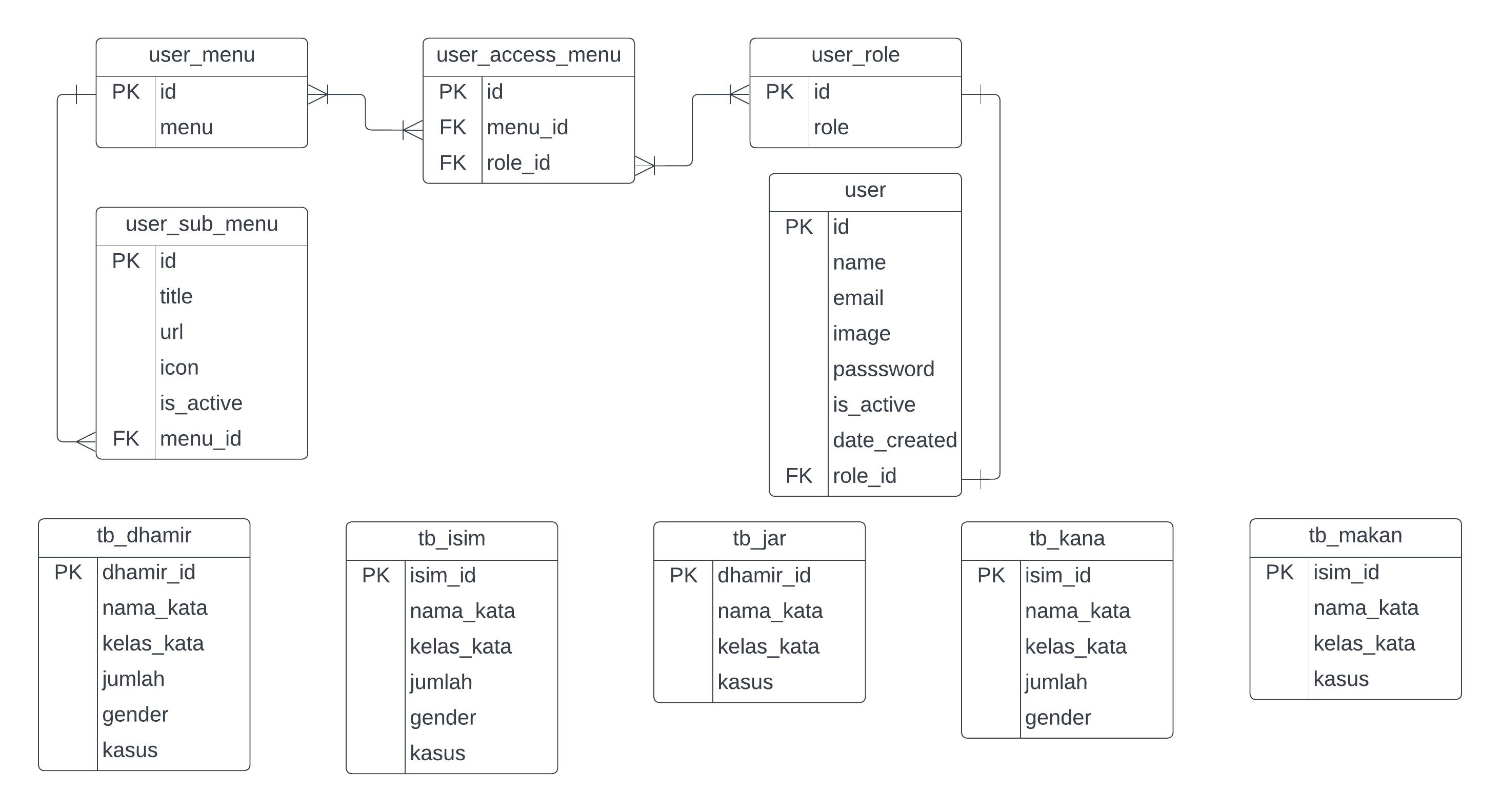
Penulis menyertakan rancang *Class Diagram* dari sistem yang akan dibuat. Secara garis besar, gambar konsep rancangan ini menjelaskan alur proses yang akan diimplementasikan pada sistem. Selanjutnya, pada saat peneliti membangun bentuk konseptual ini ke dalam bentuk fisikal, maka masih akan terdapat kemungkinan adanya perubahan proses dengan apa yang telah tervisualisasikan di dalam *Class Diagram* di bawah ini.



**Gambar 3.3** *Class Diagram*

#### 3.2.2.1 *Entity Relationship* *Diagram* (ERD)

Pada sub bab ini peneliti menyertakan rancang ERD yang menjelaskan relasi antar tabel di dalam database. Dalam rancang ERD ini terdapat Sembilan tabel master dan satu tabel *bridge* (tabel penghubung dua relasi *many to many*). Relasi tabel many to many terdapat pada tabel yang berkaitan dengan user manajemen. Sedangkan tabel yang berkaitan dengan data kata tidak memiliki relasi satu sama lain.



### 3.2.3 Perancangan Interface

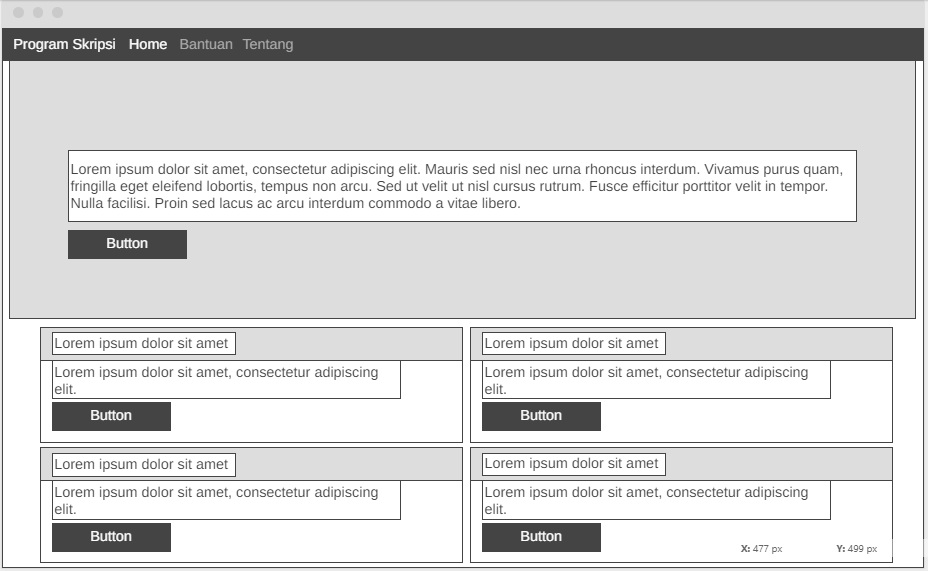
Konsep pada program ini akan dibangun untuk membantu *user* dalam mengoreki kesalahan dalam penulisan kata, frasa, klausa, dan kalimat Bahasa Arab. Di dalam konten program akan dibangun pula fitur yang memungkinkan pembahasan untuk dibagi sesuai konteks pembahasan. Sehingga akan memberikan pengalaman *user* yang berbeda dalam hal kemampuan tingkat pemahaman. Berikut ini merupakan tabel kelebihan pada program yang akan dibangun.

**Tabel 3.1** Konsep Program

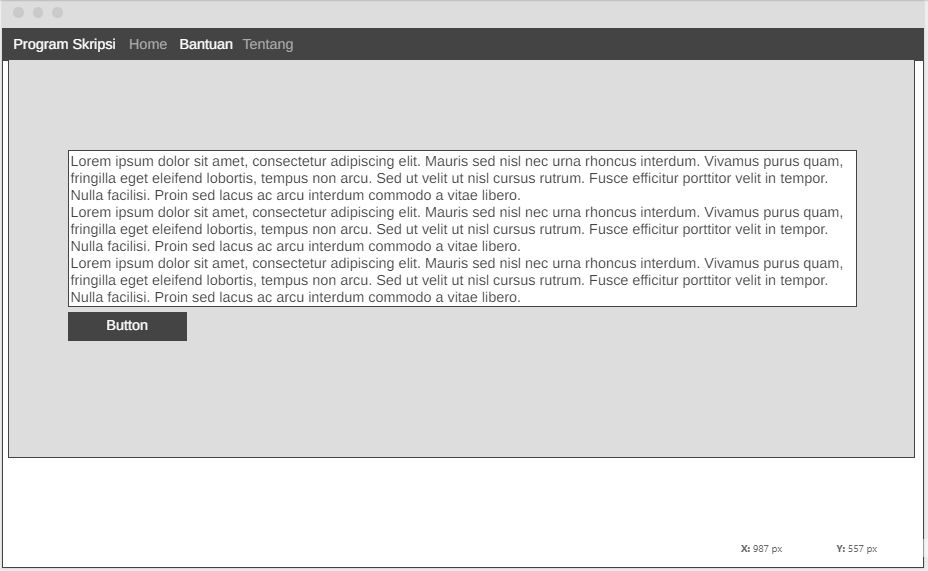
|  |  |  |
| --- | --- | --- |
| **No** | **Elemen Konsep** | **Keterangan** |
| 1. | Judul | Sistem pendeteksi kesalahan bahasa dengan menggunakan metode basis *rule* tata bahasa Arab klasik |
| 2. | Tujuan | Memanfaatkan teknologi NLP berdasarkan pada *rule* untuk memberikan pendeteksian struktur Bahasa Arab dan dampak akselerasi peningkatan pemahaman pemelajar linguistik Arab pemula dan tingkat lanjut. |
| 3. | Target Platform | *Daring* |
| 4. | Target Usia | 12 (Dua belas) tahun ke atas |
| 5. | Genre | Program Kecerdasan Buatan |
| 6. | Kelebihan | 1. Media Referensi 2. Disajikan pemahaman konteks pengoreksian yang komprehensif 3. Memiliki fitur tingkatan analisa dari frasa, klausa, dan kalimat. 4. Kebebasan bereksplorasi sambil memahami konteks-konteks koreksi susunan kata yang dipilih. |

#### 3.3.2 Desain *User Interface*

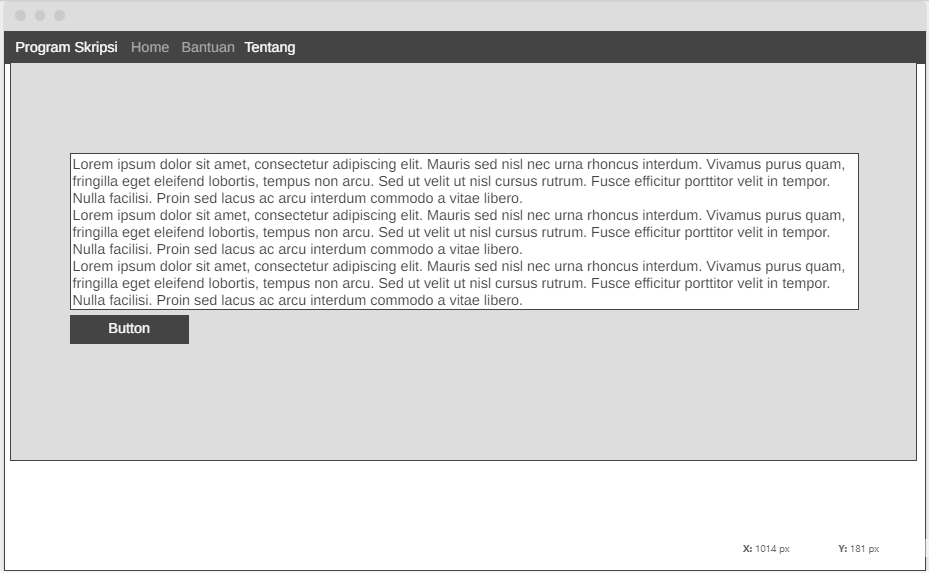
Rancangan halaman analisa dibagi menjadi empat opsi, yakni analisa kata, frasa, klausa, dan kalimat pada halaman *Home*. Lalu terdapat pula halaman untuk menampilkan Bantuan yang berisi dokumentasi dan terminologi linguistik. Terakhir terdapat halaman tentang program. Berikut adalah desain *user* *interface* secara jelas dalam bentuk *mockup*.

1. Halaman *Home*

**Gambar 3.4** Sketsa Menu Home

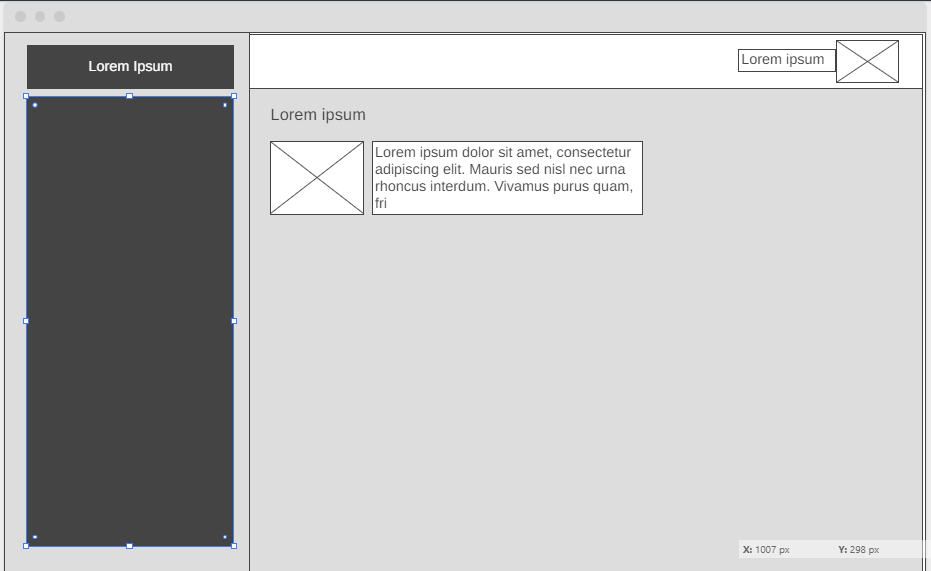
1. Halaman Menu Bantuan

**Gambar 3.4**  Tampilan Menu Bantuan

1. Halaman Tentang Program

**Gambar 3.5** Menu Tentang Program

1. Halaman Administrator Program

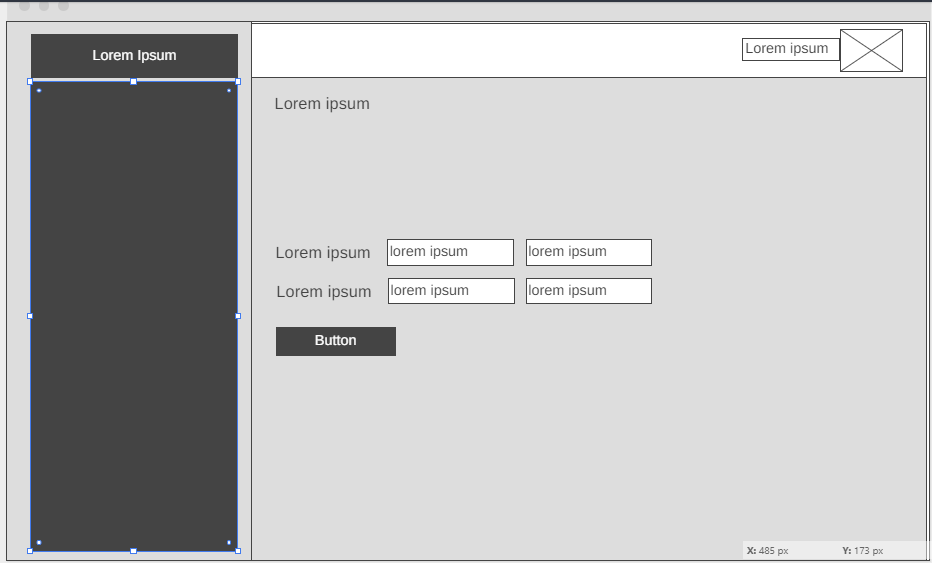


**Gambar 3.6** Menu Halaman Administrator

#### 3.3.3 Desain Menu

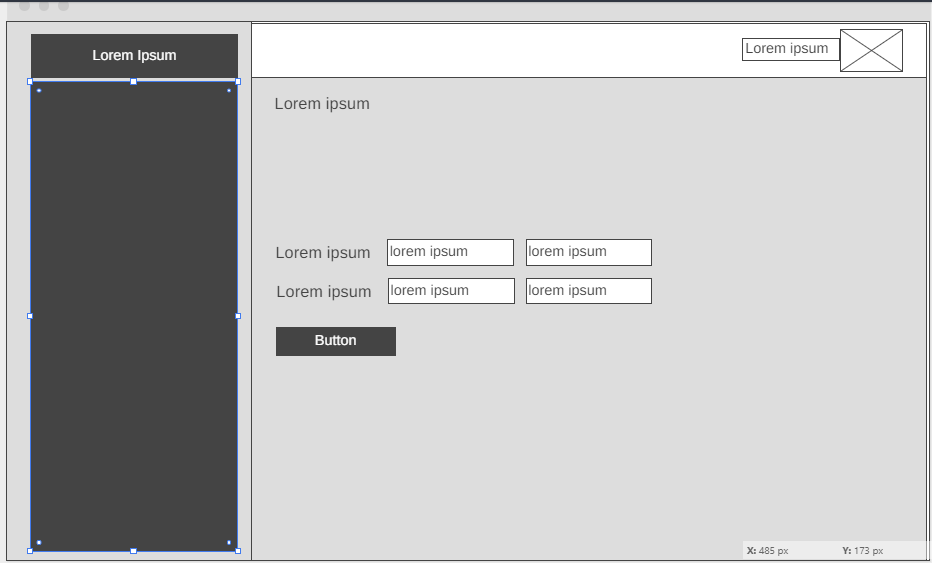
Pada sub pembahasan sebelumnya, telah diuraikan bahwasanya terdapat dua desain menu yang memiliki opsi halaman selanjutnya. Dalam hal ini yang dimaksud adalah halaman *Home* dan Bantuan. Pada *Home* memiliki empat opsi halaman analisa yang menuju halaman lanjutan. Demikian pula untuk halaman Bantuan yang terdiri dari empat sub menu pilihan, yakni Dokumentasi, Cara Menggunakan, Materi Fundamental Linguistik Arab, dan daftar terminologi bahasa.

1. **Menu Analisa Frasa Adverbia Lokatif**

Fitur ini berguna untuk memproses dan menampilkan jenis dan kelas dari frasa adverbia.

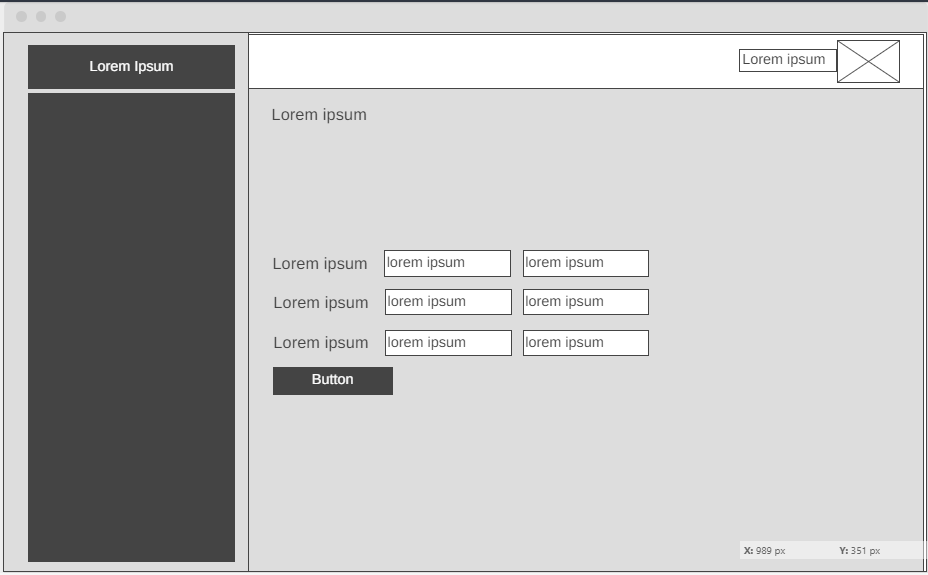
**Gambar 3.7** Menu Analisa Frasa adverbia lokatif

1. **Menu Analisa Frasa Preposisi**

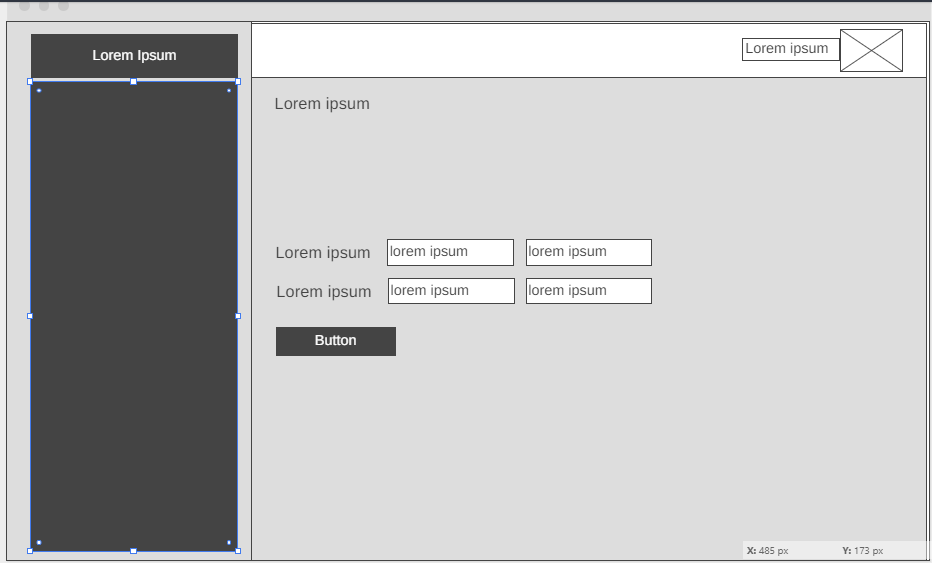


**Gambar 3.8** Sketsa Desain Menu Analisa Frasa preposisi

**Analisa Kata**

1. **Menu Analisa Klausa**

**Gambar 3.9** Sketsa Desain Menu Analisa Klausa

1. **Menu Analisa Kalimat**

**Gambar 3.10** Sketsa Desain Menu Analisa Kalimat

## 3.4 Rancangan Pengujian

Sebagai rancangan untuk Pengujian program maka peneliti menggunakan metode-metode sebagai berikut.

1. Menggunakan metode blackbox testing

Pada metode ini program akan diuji tingkat ketepatan fungsi dan Tampilan yang diharapkan menjadi hasil program yang semestinya.

1. Metode kuisioner hasil uji coba program oleh narasumber ahli.

Pada metode ini program akan diuji oleh narasumber ahli dengan diberikan kuisioner yang berkaitan dengan ketepatan sistem secara fungsi yang diharapkan.