BAB 3

ANALISIS DAN PERANCANGAN SISTEM

Analisis dan perancangan sistem merupakan suatu pendekatan yang sistematis untuk mengindentifikasi masalah, arus data dan arus informasi serta merancang sebuah sistem informasi. Analisis dan perancangan sistem memerlukan tahapan yang sistematis untuk mendapatkan rancangan sistem yang baik dan sesuai dengan kegunaan dan tujuannya. Tahap awal dari analisis adalah menganalisa kebutuhan-kebutuhan sistem mulai dari kebutuhan pengguna, kebutuhan non fungsional, dan kebutuhan fungsional. Sedangkan untuk tahap perancangan aplikasi yaitu perancangan database dan perancangan antarmuka.

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan sesuai dengan kebutuhan.

3.1.1 Analisis Masalah

Mendongeng merupakan kebiasaan yang telah membudaya bagi rakyat Indonesia. Namun kebiasaan yang telah membudaya itu menjadi luntur dan tidak

digemari lagi bagi anak-anak Indonesia khususnya. Budaya mendongeng (seni bertutur) telah di gantikan oleh peran televisi, media *mobile* atau media lainnya, anak-anak Indonesia dewasa ini lebih memilih untuk menonton kartun atau film animasi dari pada mendengarkan dongeng yang kaya akan nilai moral dalam setiap ceritanya tersebut.

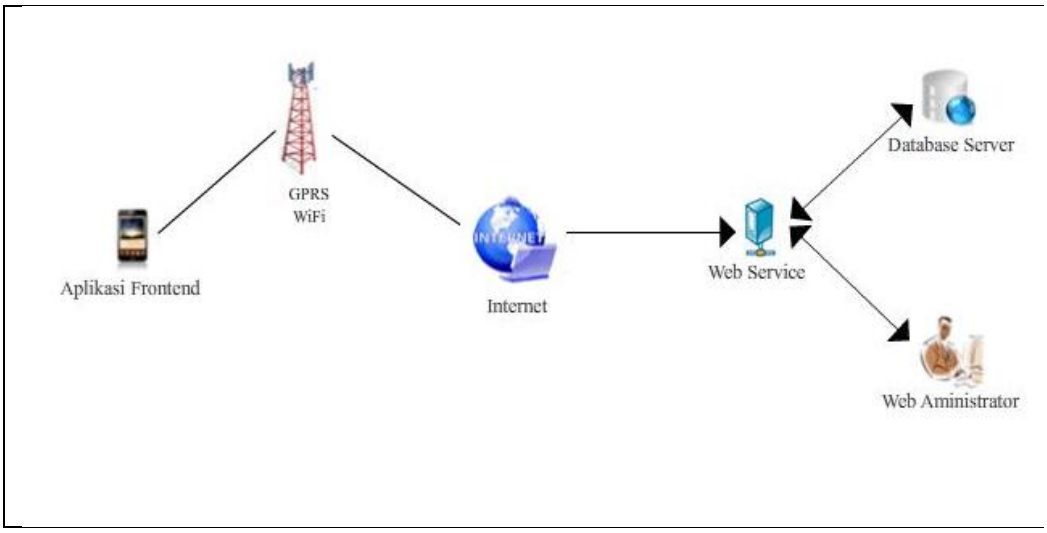
Agar tradisi mendongeng tidak hilang karena perkembangan teknologi yang semakin pesat, perlu di buat sebuah aplikasi yang dapat menampilkan cerita-cerita rakyat Indonesia yang selama ini didongengkan dari mulut ke mulut ke dalam sebuah media yang mewakili perkembangan teknologi tersebut. Salah satu media yang cukup popular dan mudah untuk di gunakan di kalangan masyarakat Indonesia adalah media *mobile*.

Aplikasi perpustakaan cerita rakyat merupakan aplikasi *mobile* yang bertujuan untuk memperkenalkan kembali budaya mendongeng dengan cerita-cerita yang di ambil dari cerita rakyat yang berasal dari seluruh rakyat Indonesia. Kelebihan dari aplikasi ini adalah menampilkan cerita dengan animasi teks yang disertai narasi, layaknya orang yang sedang membacakan dongeng tanpa kehilangan nilai moral yang terdapat didalam cerita-cerita tersebut.

3.1.2 Analisis Arsitektur Sistem

Arsitektur fisik sistem yang dibangun terdiri dari aplikasi Android (*frontend*) dan aplikasi web admin, web service, dan database server (*backend*). Arsitektur

sistem dalam pembangungan aplikasi perpustakaan cerita rakyat ini ditunjukkan pada gambar 3.1.



Gambar 3.1 Arsitektur system

Frontend adalah aplikasi perpustakaan cerita rakyat yang sudah terinstal dalam perangkat *mobile* yang digunakan (perangkat android). Aplikasi ini berfungsi sebagai antarmuka antara pengguna dengan sistem. Pengembangan antarmuka sistem, dibangun dengan menggunakan IDE Eclipse dan SDK Android dengan bahasa pemrograman java android.

Backend adalah aplikasi pendukung yang berfungsi sebagai penyuplai atau sumber data pada aplikasi *frontend*. Aplikasi ini terdiri dari tiga sub bagian utama yaitu web administrator, web service, dan database server. Web administrator merupakan web yang berfungsi sebagai pengolah sumber basis data yang terdapat pada database server yang dikelola oleh admin. Web administrator dibangun dengan menggunakan bahasa pemrograman PHP. Web service merupakan web yang

berfungsi sebagai media pertukaran data antar database server dan aplikasi. Web service dalam penelitian ini dibangun dengan menggunakan pemrograman PHP yang dikombinasikan dengan JSON (*Javascript Object Notation*). Database server merupakan aplikasi yang berfungsi sebagai sumber data (*data resources*) aplikasi. Pada bagian ini pengelolaan database menggunakan MySQL. Akses koneksi jaringan yang digunakan dari *frontend* untuk terhubung ke web service melalui perangkat *mobile* menggunakan jaringan seluler.

3.1.3 Analisis Sistem Multimedia

Seperti yang dijelaskan dalam bab 2, multimedia secara umum merupakan kombinasi tiga elemen, yaitu suara, gambar dan teks. Dalam pembangunan aplikasi ini sistem multimedia merupakan alat yang dapat menciptakan presentasi yang dinamis dan interaktif untuk mengkombinasikan animasi teks dan narasi dari cerita rakyat. Setiap kata yang di beri animasi teks akan di ikuti oleh narasi yang telah di rekam di sesuaikan dengan teks cerita rakyat tersebut.

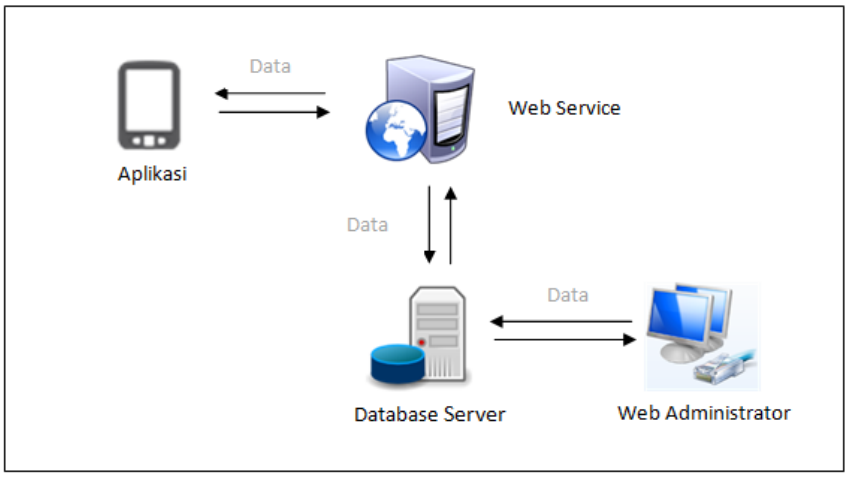
Sinkronisasi animasi teks dengan narasi merupakan inti dari aplikasi ini, dimana dengan sinkronisasi tersebut menjadikan aplikasi ini berfungsi sebagai sarana edukasi bagi anak untuk membaca. Dengan demikian setiap kata yang didengar akan sama dengan animasi teks yang dilihat (Visual + Audio = Audiovisual). Selain menampilkan animasi teks dan narasi aplikasi cerita rakyat ini juga dilengkapi dengan ilustrasi gambar yang menarik dan disesuaikan dengan tema cerita, dengan

demikian aplikasi cerita rakyat ini dapat disebut sebagai aplikasi multimedia karena didalam aplikasi ini terdapat kombinasi tiga elemen utama multimedia yaitu, suara, gambar dan teks.

Aplikasi ini menggunakan mekanisme streaming, dimana pada aplikasi ini memungkinkan untuk melakukan pendistribusian data audio, dan multimedia secara real-time melalui Internet. Pada aplikasi perpustakaan cerita rakyat ini dilakukan pengiriman media digital (berupa animasi, suara dan ilustrasi gambar) agar bisa diterima secara terus-menerus (stream). Data tersebut dikirim dari server dan diterima serta ditampilkan secara real- time pada aplikasi *frontend*(android).

3.1.4 Analisis Alur Data Sistem

Analisis alur data sistem merupakan analisis yang berfungsi untuk menggambarkan secara rinci bagaimana sistem bekerja.



Gambar 3.2 Alur Data Sistem

Analisis ini meliputi:

1. Aplikasi *mobile*

   Aplikasi *mobile* dalam hal ini adalah analisis fungsionalitas multimedia dimana pada aplikasi ini akan ditampilkan animasi teks disertai dengan narasi, pada aplikasi ini animasi teks harus mengikuti narasi/audio, baik saat aplikasi itu dalam posisi *play, pause* maupun posisi *stop*, animasi teks harus tetap mengikuti audio yang muncul. Secara keseluruhan aplikasi perpustakaan cerita rakyat yang dibangun pada *platform* android ini menggunakan *PhoneGap*, dimana *PhoneGap* merupakan sebuah framework *opensource* yang di pakai untuk membuat aplikasi *cross-platform mobile* dengan menggunakan bahasa pemrograman HTML, javascript dan CSS. Sedangkan untuk komunikasi data antar aplikasi dengan database server menggunakan parsing data dari web service yang bertipe JSON.

2. Web *service*

   Web service pada penelitian ini berfungsi sebagai *middle application* atau perantara aplikasi dengan database server. Cara kerja web service ini yaitu dengan mengambil data dari database server dan kemudian mengkonversi data tersebut kedalam format pertukaran data JSON.

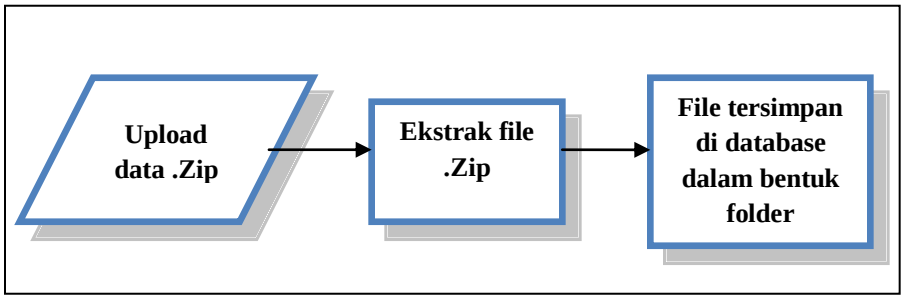
3. *Protocol* HTTP

   Protocol HTTP merupakan *protocol* jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan menggunakan hypermedia.

Pada penelitian ini HTTP digunakan aplikasi *mobile* dan web administrator sebagai protocol yang dapat men-distribusikan data yang bersumber dari database server.

3.1.5 Analisis Pertukaran Data *Backend* dan *Frontend*

Analisis penyimpanan data merupakan analisis yang berfungsi untuk menggambarkan bagaimana proses penyimpanan data cerita, mulai dari proses penambahan data cerita di halaman web hingga tersimpan di dalam database server.



Gambar 3.3 Analisis Penyimpanan Data

Proses penyimpanan data dilakukan pada aplikasi web(admin). Pada aplikasi web terjadi proses penambahan data cerita dalam format .Zip. Saat penambahan data selesai, file dalam format .Zip akan tersimpan dalam database server. Saat tersimpan ke dalam database akan ada proses ekstrak file dalam bentuk folder secara otomatis. File yang telah di ekstrak dalam bentuk folder tersebut yang akan dipanggil oleh aplikasi *mobile*.

3.1.6 Analisis Pertukaran Data *Backend* dan *Frontend*

Agar data cerita yang tersimpan dalam database server dapat tampil dalam aplikasi, maka digunakan JSON, format penulisan data untuk menampilkan atau mentransfer data yang tersimpan dalam database server ke aplikasi yang berada pada *client* yaitu pengguna aplikasi. Terdapat dua bagian utama data yang dipertukarkan yaitu data kategori dan data cerita.

Berikut adalah contoh format data yang dipertukarkan :

1) Format data kategori

```
{
    "kategori": [
        {
            "id_kategori": "K001",
            "nama_kategori": "Kisah",
            "image_kategori": "kategori.jpg"
        }
]
}
```

Format data kategori adalah data yang pertama kali di unduh oleh aplikasi ketika pengguna mengaktifkan aplikasi perpustakaan cerita rakyat. Data ini menjadi tampilan utama aplikasi atau menu utama dimana terdapat beberapa pilihan kategori cerita yang dapat dipilih oleh pengguna.

2) Format data cerita

```
{
    "cerita": [
        {
            "id_cerita": "C001",
            "id_kategori": "K001",
            "nama_cerita": "Timun Emas",
          "image_cerita": "timunemas.jpg",
            "upload_file": "timunemas.apk",

        }
]
}
```

Format data cerita adalah data yang berisi keterangan mengenai cerita-cerita rakyat berdasarkan kategori yang telah dipilih sebelumnya.

3.1.7 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang menitik-beratkan pada properti perilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras, perangkat lunak, serta *user* sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.1.7.1 Spesifikasi Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan kumpulan perintah-perintah yang diberikan kepada perangkat keras agar saling berinteraksi untuk melakukan suatu tugas. Perangkat lunak yang dibutuhkan untuk membangun aplikasi Perpustakaan Cerita Rakyat adalah sebagai berikut:

1) Sistem operasi Windows
2) JDK (*Java Development Kit*) 1.6, SDK (*Standard Developmenet Kit*), dan NDK (*Native Development Kit*) untuk Windows
3) ADT (*Android Development Tools*) 0.9.9
4) Eclipse versi Indigo
5) Android emulator versi 2.3 (*Gingerbread*).
6) Adobe Dreamweaver CS6.

3.1.7.2 Spesifikasi Perangkat Keras

Komputer dan *smartphone* terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat keras untuk melakukan tugas tertentu. Pembangunan aplikasi Perpustakaan Cerita Rakyat, perangkat keras yang digunakan yaitu yang terlihat pada tabel 3.3 berikut:

Tabel 3.1 Spesifikasi Perangkat Lunak

No	Perangakat keras	Spesifikasi perangkat keras Pembangun (Komputer)	Spesifikasi perangkat keras Pengguna (*Mobile*)
1	Prosesor	Prosesor 1,6 GHz	600 MHz
2	Resolusi layar	1024 x 800 pixel	320 x 480 pixel
3	Memori	2 Gb	128 MB RAM; 256 MB Flash External
4	Harddisk	250 Gb	-
5	Monitor/LCD	-	-
6	Keyboard dan Mouse	-	-

3.1.7.3 Spesifikasi Pengguna Sistem (*user*)

Spesifikasi pengguna sistem dimaksudkan untuk mengetahui siapa saja *actor* yang terlibat dalam menjalankan sistem. Aplikasi Perpustakaan Cerita Rakyat menggunakan *platform* teknologi *mobile* dan web.

Pengguna sistem dibagi menjadi dua bagian, yaitu :

1. Pengguna aplikasi perpustakaan cerita rakyat (*end user*) yaitu para pengguna *mobilephone* dengan *platform* android versi 2.3 atau lebih tinggi. Dalam menggunakan sistem ini, pengguna diharuskan terhubung dengan koneksi jaringan GPRS/EDGE/UMTS untuk bisa mengunduh aplikasi ini.

2. Admin sebagai pengelola konten mempunyai kapabilitas dalam mengelola database seperti menambah, mengubah, dan menghapus data melalui web administrator.

Berikut ini karakteristik pengguna sistem yang ditunjukkan oleh tabel berikut:

Tabel 3.2 Tabel Karakteristik Pengguna

Pengguna	Hak Akses	Pengalaman
User *Frontend*	Mengakses data konten perpustakaan cerita rakyat melalui aplikasi front-end.	Tidak dibutuhkan pengalaman karena masih melibatkan orang dewasa(ORTU) dalam penggunaan aplikasi.
User *BackEnd*	Mengakses, mengelola aplikasi back-end	Min. Mengatur dan memperbaiki data perpustakaan cerita rakyat pada web admin

3.1.8 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan. Analisis kebutuhan fungsional ini meliputi analisis kebutuhan sistem, analisis kebutuhan data, spesifikasi sistem dan pemodelan sistem.

Pemodelan sistem dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain *Use Case Diagram, Class Diagram, Activity Diagram, Sequence Diagram, Collaboration Diagram,* dan *Component Diagram.*

3.1.8.1 Analisis Kebutuhan Sistem

Dalam perancangan sistem pada aplikasi yang dibangun, sebelumnya dilakukan analisis kebutuhan sistem agar sistem yang dibangun sesuai dengan yang diharapkan atau dibutuhkan pengguna. Adapun kebutuhan sistem yang diperlukan antara lain:

1. Menampilkan informasi kategori yang terdapat di dalam perpustakaan cerita rakyat.
2. Menampilkan nama cerita rakyat berdasarkan kategori yang dipilih pengguna.
3. Menampilkan konten cerita rakyat yang berisikan animasi teks disertai narasi dan gambar.

3.1.8.2 Analisis Kebutuhan Data

Analisis kebutuhan data bertujuan untuk menganalisis kebutuhan data yang diperlukan dan memudahkan perancangan informasi setelah sistem yang dibutuhkan telah diketahui. Adapun analisis kebutuhan data dengan kriteria sebagai berikut:

1 Data Kategori cerita

Menampilkan data kategori yang meliputi nama kategori dan gambar dari kategori cerita.

2 Data Cerita

Menampilkan data Cerita yang meliputi nama cerita, gambar cerita, dan konten cerita rakyat berdasarkan kategori yang telah dipilih.

dan keterangan.

3.1.8.3 Spesifikasi Sistem

Adapun spesifikasi sistem yang dibutuhkan sebagai berikut :

1 Pengembangan sistem dibagi menjadi dua proses yaitu *frontend*, aplikasi yang berjalan dipihak *client* dan *backend*, berupa *database server* sebagai sumber informasi aplikasi.

2 Komunikasi antara aplikasi *client* dan *server* melalui koneksi protokol HTTP dengan memanfaatkan jaringan seluler.

3 Sistem yang berjalan pada *client* (*frontend*) sifatnya hanya menampilkan konten perpustakaan cerita rakyat tanpa keterlibatan pengguna dalam prosesnya.

4 Spesifikasi sistem pada sisi *frontend*, yaitu :

a. Lokasi saat mengunduh aplikasi *frontend* dianjurkan dalam lingkungan terbuka atau tidak sedang berada dalam bangunan tertutup agar proses unduh aplikasi berjalan lancar.

b. Pengguna berada dalam wilayah cakupan jaringan seluler.

5 Spesifikasi sistem pada sisi *server* (*backend*), yaitu :

a. Memfasilitasi kebutuhan data pada aplikasi *frontend*.

b. Dapat dilakukan pengelolaan data seperti tambah, ubah, dan hapus data.

3.1.8.4 Pemodelan Sistem

Pemodelan sistem berfungsi untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem pada aplikasi serta menentukan kelas yang dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

1) Use Case Diagram

Digunakan untuk memodelkan atau menggambarkan batasan sistem dan fungsi-fungsi utamanya. Mendiskripsikan fungsi dari sebuah sistem dari perspektif pengguna, use case bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* terdiri dari tiga bagian yaitu identifikasi *actor*, identifikasi *use case*, dan *scenario use case*.

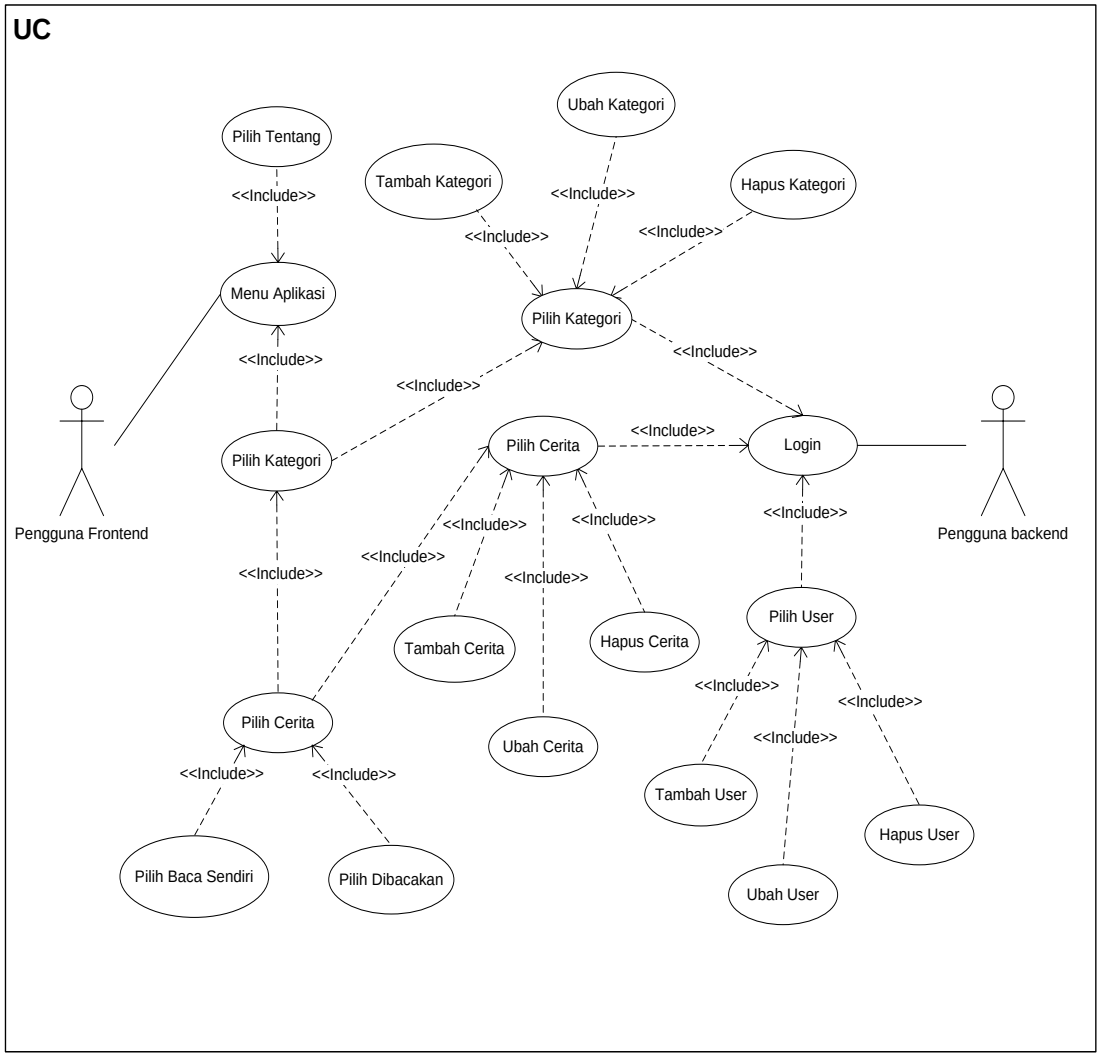
1. Identifikasi *Actor*

   Aktor adalah abstraksi dari orang dan sistem yang lain yang mengaktifkan fungsi dari target sistem. Berikut adalah aktor-aktor yang berperan dalam menjalan sistem yang dibangun.

Tabel 3.3 Tabel *Actor Use Case*

No	Aktor	Deskripsi
A-01	Pengguna *Frontend*	Merupakan *actor* atau pengguna akhir dari aplikasi *mobile*.
A-02	Pengguna *Backend*	Merupakan *actor* yang berperan mengelola data informasi dalam *database server*.

Pada gambar 3.3 penggambaran *use case diagram* sistem terdiri dari *frontend* dan *backend*. *Use case diagram* menunjukkan bagaimana interaksi antara aktor dan sistem, dimana terdapat dua aktor yang berperan dan sistem yang terdiri dari aplikasi *frontend* dan *backend*. Aktor pengguna *frontend* yaitu yang menjalankan aplikasi *frontend*. Sedangkan aktor pengguna *backend* berperan mengelola aplikasi *backend* berbasis web.



Gambar 3.4 *Use Case Diagram* aplikasi perpustakaan cerita rakyat

2. Identifikasi *Use Case Diagram*

Berikut tabel identifikasi use case yang terdapat dalam sistem :

Tabel 3.4 Identifikasi *Use Case Diagram*

Use Case* Aplikasi *Frontend		
No	***Use Case***	**Deskripsi**
UCF-01	Menu Aplikasi	Fungsionalitas untuk memulai aplikasi.

UCF-02	Pilih Tentang	Fungsionalitas yang memungkinkan pengguna untuk melihat tentang aplikasi (Pembuat dan cara menggunakan aplikasi).
UCF-03	Pilih Kategori	Fungsionalitas yang memungkinkan pengguna memilih kategori cerita terdiri dari kategori kisah, Asal Mula dan Legenda.
UCF-04	Pilih Cerita	Fungsionalitas yang memungkinkan pengguna memilih cerita berdasarkan jenis kategori.
UCF-05	Pilih Baca Sendiri	Fungsionalitas untuk menampilkan isi cerita yang terdiri dari tampilan teks dan ilustrasi gambar.
UCF-06	Pilih Dibacakan	Fungsionalitas untuk menampilkan isi cerita yang terdiri dari tampilan animasi teks, narasi dan ilustrasi gambar.
Use Case* Aplikasi *Backend		
No	***Use Case***	**Deskripsi**
UCB-01	Login	Fungsionalitas untuk proses autentifikasi hak akses kepada admin
UCB-02	Kategori	Fungsionalitas yang memungkinkan admin dapat mengelola data kategori.
UCB-03	Tambah Kategori	Fungsionalitas untuk menambahkan data kategori.
UCB-04	Ubah Kategori	Fungsionalitas untuk mengubah atau *update* data

		kategori.
UCB-05	Hapus Kategori	Fungsionalitas untuk menghapus data kategori.
UCB-06	Cerita	Fungsionalitas yang memungkinkan admin dapat mengelola data cerita.
UCB-07	Tambah Cerita	Fungsionalitas untuk menambahkan data Cerita.
UCB-08	Ubah Cerita	Fungsionalitas untuk mengubah atau *update* data Cerita.
UCB-09	Hapus Cerita	Fungsionalitas untuk menghapus data Cerita.
UCB-10	User	Fungsionalitas yang memungkinkan admin dapat mengelola data User.
UCB-11	Tambah User	Fungsionalitas untuk menambahkan data User.
UCB-12	Ubah User	Fungsionalitas untuk mengubah atau *update* data User.
UCB-13	Hapus User	Fungsionalitas untuk menghapus data User.

3. Skenario *Use Case*

A. Aplikasi *Frontend*

Skenario proses-proses yang terdapat dalam *use case diagram* aplikasi *frontend* dapat dilihat pada tabel-tabel berikut:

Tabel 3.5 Skenario Use Case Menu Aplikasi

Identifikasi	
Nama Use Case	Kategori
Aktor	Pengguna *Frontend*
Tujuan	Memulai menggunakan aplikasi
Keadaan awal	Sistem menampilkan menu *Aplikasi*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan Daftar menu yang terdapat di dalam aplikasi perpustakaan cerita rakyat.
2. Melihat tampilan menu aplikasi	
Kondisi Akhir	Menampilkan menu aplikasi

Tabel 3.6 Skenario Use Case Pilih Tentang

Identifikasi	
Nama Use Case	Kategori
Aktor	Pengguna *Frontend*
Tujuan	Memilih menu tentang
Keadaan awal	Sistem menampilkan menu *tentang*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**

1. Memilih menu tentang aplikasi	
	2. Menampilkan isi dari menu tentang aplikasi.
Kondisi Akhir	Menampilkan Isi tentang Aplikasi

Tabel 3.7 Skenario Use Case Pilih Kategori

Identifikasi	
Nama Use Case	Kategori
Aktor	Pengguna *Frontend*
Tujuan	Memilih jenis (kategori) Cerita Rakyat
Keadaan awal	Sistem menampilkan menu *kategori*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan jenis-jenis kategori (Kisah, Asal Mula, Legenda)
2. Memilih menu pada kategori	
	3. Aplikasi merespon dengan cara memanggil dan melihat kelas kategori yang dipilih.
Kondisi Akhir	Menampilkan nama cerita berdasarkan kategori yang

	dipilih.

Tabel 3.8 Skenario Use Case Pilih Cerita

Identifikasi	
Nama *Use Case*	Konten (Cerita)
Aktor	Pengguna *Frontend*
Tujuan	Memilih jenis cerita (Perpustakaan Cerita Rakyat) berdasarkan kategori
Keadaan Awal	Sistem menampilkan nama cerita berdasarkan kategori.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menemukan nama cerita berdasarkan kategori.	
	2. Aplikasi merespon dengan cara memanggil dan melihat kelas cerita yang dipilih.
Keadaan Akhir	Menampilkan menu cerita yang terdiri dari dibaca sendiri atau dibacakan.

B. Skenario *Use Case* Aplikasi *Backend*

Skenario proses-proses yang terdapat dalam *use case diagram* aplikasi *backend* dapat dilihat pada tabel-tabel berikut :

Tabel 3.9 Skenario *Use Case* Login

Identifikasi	
Nama Use Case	*Login*
Aktor	Pengguna *Backend*
Tujuan	Masuk kedalam sistem
Keadaan Awal	Sistem menampilkan halaman login admin
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Mengisi *username* dan *password* pada halaman web *login* admin	
	2. Sistem melakukan autentikasi *username* dan *password* yang diiunputkan.
	3. Bila sesuai maka menampilan halaman web menu utama.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan pesan "*Error : Invalid username or password. Please try again*"
2. Melakukan input *username* dan	

password pada halaman web login admin	
Kondisi Akhir	Halaman menu utama ditampilkan

Tabel 3.10 Skenario Use Case Kategori

Identifikasi	
Nama Use Case	Kategori
Aktor	Pengguna *Backend*
Tujuan	Menampilkan data kategori secara keseluruhan serta fungsionalitas pengolahan data
Keadaan Awal	Menampilkan halaman utama Web Admin
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih Kategori pada menu navigator.	
	2. Menampilkan halaman pengolahan kategori
Kondisi Akhir	Menampilkan data kategori.

Tabel 3.11 Skenario *Use Case* Tambah Kategori

Identifikasi	
Nama Use Case	Tambah Kategori
Aktor	Pengguna *Backend*

Tujuan	Menambah data kategori
Keadaan Awal	Sistem menampilkan halaman pengelohan kategori.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih *Tambah* pada pilihan halaman Kategori	
	2. Menampilkan halaman pengisian tambah kategori.
3. Mengisi *field-field* pada pengisian tambah *kategori*.	
	4. Melakukan proses tambah data pada basis data.
	5. Memunculkan informasi "Tambah Data Sukses"
6. Menyetujui penambahan dengan memilih atau menekan tombol "OK"	
	7. Menyimpan hasil tambah data.
Kondisi Akhir	Data kategori bertambah sesuai dengan input data.

Tabel 3.12 Skenario *Use Case* Ubah Kategori

Identifikasi	
Nama Use Case	Ubah Kategori
Aktor	Pengguna *Backend*
Tujuan	Mengubah data kategori
Keadaan Awal	Sistem menampilan halaman pengolahan kategori
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih tombol ubah pada kolom *action* tabel Kategori	
	2. Menampilkan halaman ubah kategori.
3. Mengisi *field-field* pada halaman ubah kategori	
	4. Melakukan proses ubah data pada basis data dan menyimpannya.
	5. Memunculkan informasi “Ubah Data Sukses”
6. Menyetujui pengubahan data dengan memilih atau menekan tombol “OK”	

	7. Menyimpan hasil Ubah data.
Keadaan Akhir	Perubahan data kategori sesuai dengan ubah data.

Tabel 3.13 Skenario *Use Case* Hapus Kategori

Identifikasi	
Nama Use Case	Hapus Kategori
Aktor	Pengguna *backend*
Tujuan	Menghapus data kategori
Keadaan Awal	Sistem menampilkan halaman pengolahan kategori
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih tombol hapus pada kolom *action* tabel Kategori	
	2. Menampilkan pesan konfirmasi *hapus kategori*
3. Menyetujui penghapusan dengan memilih atau menekan tombol “OK”	
	4. Menghapus data kategori.
	5. Menyimpan hasil perubahan data

Kondisi Akhir	Perubahan data kategori sesuai dengan data yang dihapus.

Tabel 3.14 Skenario Use Case Cerita

Identifikasi	
Nama Use Case	Cerita
Aktor	Pengguna *Backend*
Tujuan	Menampilkan data cerita secara keseluruhan serta fungsionalitas pengolahan data
Keadaan Awal	Menampilkan halaman utama Web Admin
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih Cerita pada menu navigator.	
	2. Menampilkan halaman pengolahan cerita.
Kondisi Akhir	Menampilkan data cerita.

Tabel 3.15 Skenario *Use Case* Tambah cerita

Identifikasi	
Nama Use Case	Tambah Cerita
Aktor	Pengguna *Backend*
Tujuan	Menambah data cerita.

Keadaan Awal	Sistem menampilkan halaman pengelohan cerita.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih *Tambah* pada pilihan halaman cerita.	
	2. Menampilkan halaman pengisian tambah cerita.
3. Mengisi *field-field* pada pengisian tambah cerita.	
	4. Melakukan proses tambah data pada basis data.
	5. Memunculkan informasi "Tambah Data Sukses"
6. Menyetujui penambahan data dengan memilih atau menekan tombol "OK"	
	7. Menyimpan hasil tambah data.
Kondisi Akhir	Data Cerita bertambah sesuai dengan input data.

Tabel 3.16 Skenario *Use Case* Ubah Cerita

Identifikasi	
Nama Use Case	Ubah Cerita
Aktor	Pengguna *Backend*
Tujuan	Mengubah data kategori
Keadaan Awal	Sistem menampilan halaman pengolahan kategori
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih tombol ubah pada kolom *action* tabel Cerita.	
	2. Menampilkan halaman ubah Cerita.
3. Mengisi *field-field* pada halaman ubah Cerita.	
	4. Melakukan proses ubah data pada basis data dan menyimpannya.
	5. Memunculkan informasi “Ubah Data Sukses”
6. Menyetujui pengubahan data dengan memilih atau menekan tombol “OK”	
	7. Menyimpan hasil Ubah data.

Keadaan Akhir	Perubahan data Cerita sesuai dengan ubah data.

Tabel 3.17 Skenario *Use Case* Hapus Cerita

Identifikasi	
Nama Use Case	Hapus Cerita.
Aktor	Pengguna *backend*
Tujuan	Menghapus data Cerita.
Keadaan Awal	Sistem menampilkan halaman pengolahan Cerita.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih tombol hapus pada kolom *action* tabel Cerita.	
	2. Menampilkan pesan konfirmasi *hapus Cerita*.
3. Menyetujui penghapusan dengan memilih atau menekan tombol “OK”	
	4. Menghapus data Cerita.
	5. Menyimpan hasil perubahan data

Kondisi Akhir	Perubahan data Cerita sesuai dengan data yang dihapus.

Tabel 3.18 Skenario Use Case User

Identifikasi	
Nama Use Case	User
Aktor	Pengguna *Backend*
Tujuan	Menampilkan data User secara keseluruhan serta fungsionalitas pengolahan data
Keadaan Awal	Menampilkan halaman utama Web Admin
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih User pada menu navigator.	
	2. Menampilkan halaman pengolahan User.
Kondisi Akhir	Menampilkan data User.

Tabel 3.19 Skenario *Use Case* Tambah User

Identifikasi	
Nama Use Case	Tambah User
Aktor	Pengguna *Backend*
Tujuan	Menambah data user

Keadaan Awal	Sistem menampilkan halaman pengelohan user
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih *Tambah* pada pilihan halaman ce user.	
	2. Menampilkan halaman pengisian tambah user.
3. Mengisi *field-field* pada pengisian tambah user.	
	4. Melakukan proses tambah data pada basis data.
	5. Memunculkan informasi "Tambah Data Sukses"
6. Menyetujui penambahan data dengan memilih atau menekan tombol "OK"	
	7. Menyimpan hasil tambah data.
Kondisi Akhir	Data user bertambah sesuai dengan tambah data.

Tabel 3.20 Skenario *Use Case* Ubah User

Identifikasi	
Nama Use Case	Ubah User
Aktor	Pengguna *Backend*
Tujuan	Mengubah data user
Keadaan Awal	Sistem menampilan halaman pengolahan user
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih tombol ubah pada kolom *action* tabel user.	
	2. Menampilkan halaman ubah user.
3. Mengisi *field-field* pada halaman ubah user.	
	4. Melakukan proses ubah data pada basis data dan menyimpannya.
	5. Memunculkan informasi “Ubah Data Sukses”
6. Menyetujui pengubahan data dengan memilih atau menekan tombol “OK”	
	7. Menyimpan hasil Ubah data.

Keadaan Akhir	Perubahan data user sesuai dengan ubah data.

Tabel 3.21 Skenario *Use Case* Hapus User

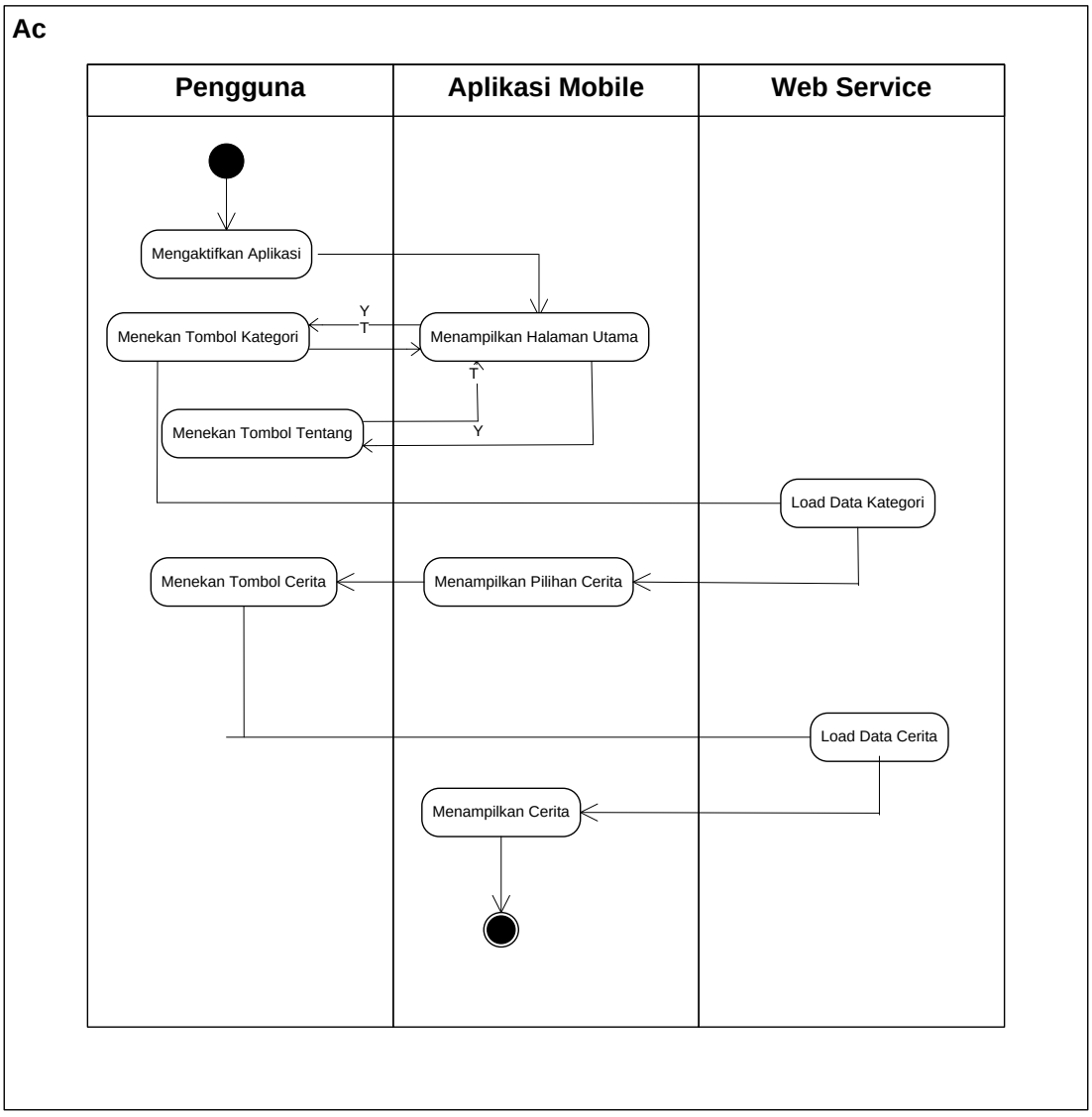
Identifikasi	
Nama Use Case	Hapus User
Aktor	Pengguna *backend*
Tujuan	Menghapus data user
Keadaan Awal	Sistem menampilkan halaman pengolahan user.
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memilih tombol hapus pada kolom *action* tabel user.	
	2. Menampilkan pesan konfirmasi *hapus user*.
3. Menyetujui penghapusan dengan memilih atau menekan tombol “OK”	
	4. Menghapus data user.
	5. Menyimpan hasil perubahan data.

Kondisi Akhir	Perubahan data user sesuai dengan data yang dihapus.

2) Activity Diagram Perpustakaan cerita Rakyat

Diagram activity menggambarkan berbagai alir aktivitas dalam sistem yang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. Penggambaran *activity* diagram memiliki kemiripan dengan *flowchart diagram*. *Activity diagram* memodelkan *event-event* yang terjadi pada *Use Case* dan digunakan untuk pemodelan aspek dinamis dari sistem.

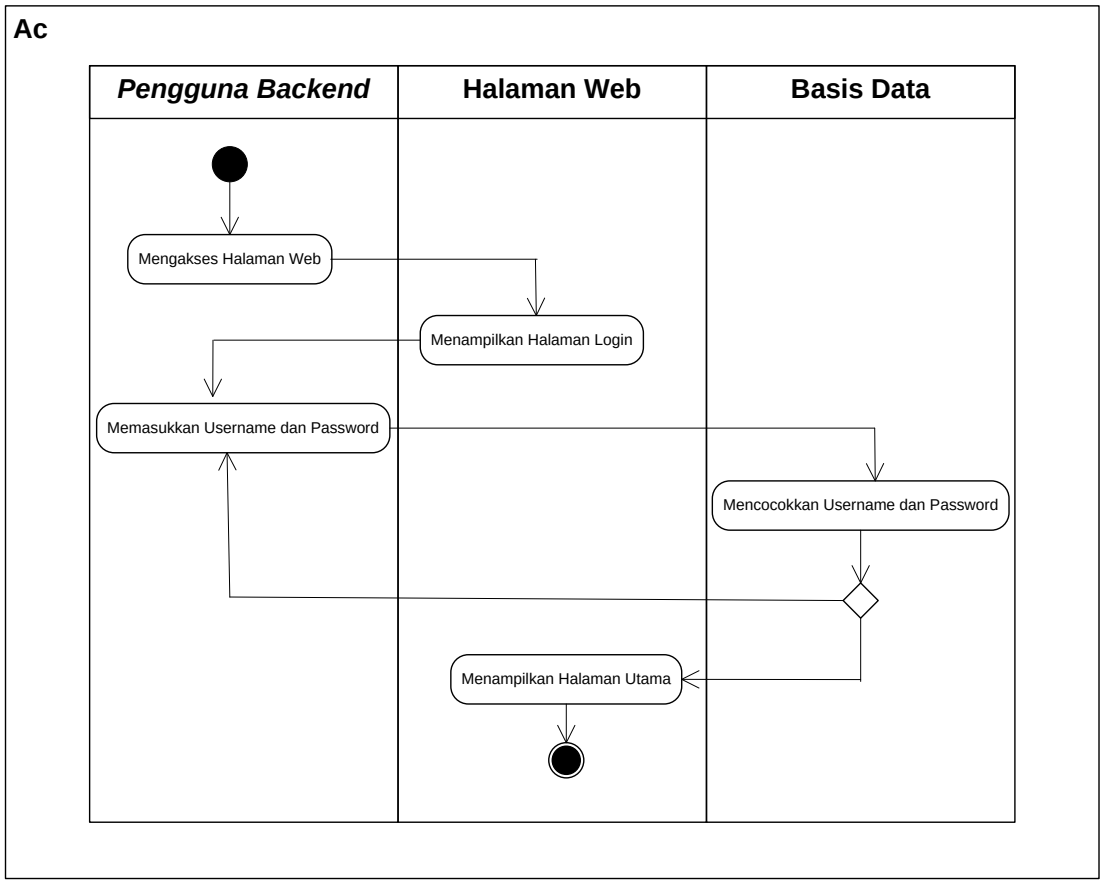
1. *Activity Diagram* Aplikasi *Frontend*



Gambar 3.5 *Activity Diagram* Aplikasi *Frontend*

Gambar 3.5, menggambarkan aktivitas yang terjadi pada aplikasi saat pengguna pertama kali mengaktifkan aplikasi, Dimana dalam aplikasi akan menyajikan beberapa jenis kategori dan beberapa pilihan cerita yang dapat dipilih pengguna.

2. *Activity Diagram* Aplikasi *BackEnd*-Login

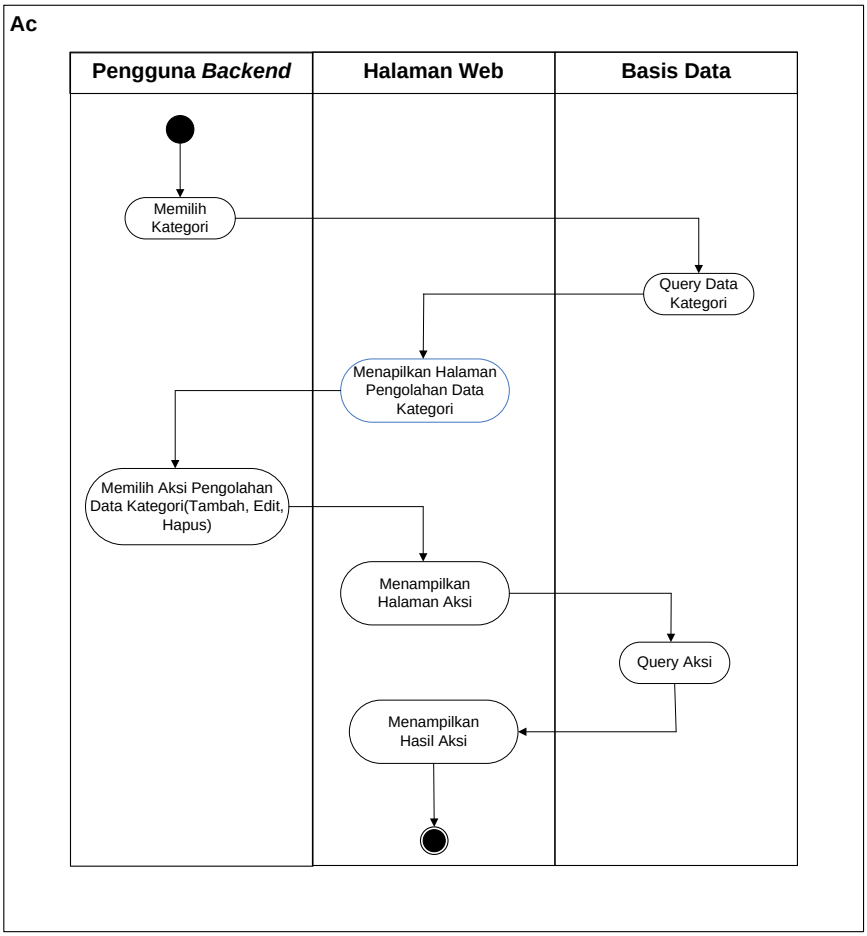


Gambar 3.6 *Activity Diagram* Aplikasi *BackEnd*-Login

Gambar 3.6, menggambarkan aktivitas login untuk masuk ke halaman utama Web. Admin mengakses halaman Web, kemudian Web akan menampilkan halaman login, kemudian admin melakukan input *username* dan *password*. Server akan melakukan verifikasi, jika *username* dan *password* tidak sesuai maka admin akan diarahkan kembali ke halaman login untuk melakukan input *username* dan *password* yang

valid. Jika *username* dan *password* yang di input valid, maka akan ditampilkan halaman utama web Admin.

3. *Activity Diagram* Aplikasi *BackEnd*-Kategori

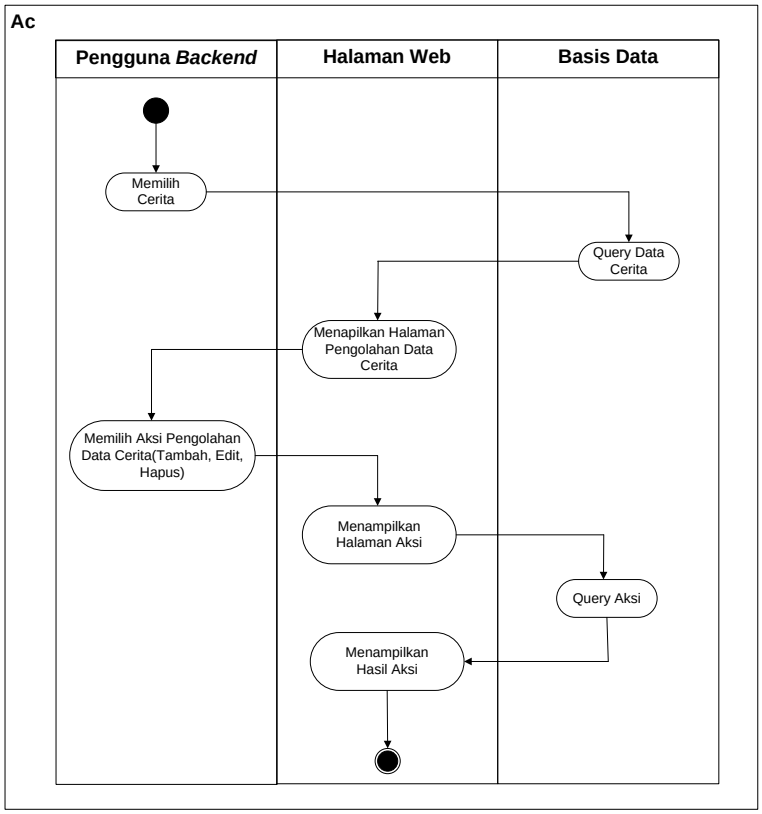


Gambar 3.7 *Activity Diagram* Aplikasi *BackEnd*-Kategori

Gambar 3.7, menggambarkan aktivitas pengolahan data kategori. Proses ini berjalan setelah admin berhasil melakukan login. Admin memilih menu kategori, selanjutnya Web akan menampilkan halaman kategory. Admin dapat mengelola data kategori dengan menggunakan

aksi Tambah, Ubah, atau Hapus. Web akan menampilkan halaman berdasarkan aksi yang dilakukan admin, saat bersamaan *query* akan dipanggil dan dieksekusi pada database server kemudian hasilnya ditampilkan pada web.

4. *Activity Diagram* Aplikasi *BackEnd*-Cerita

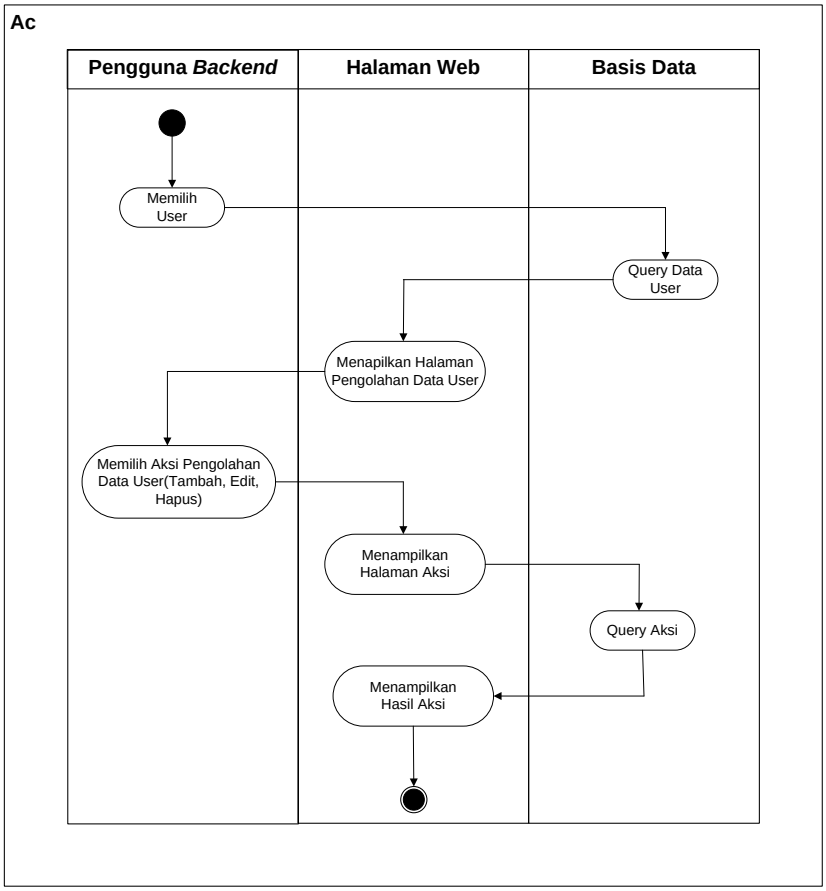


Gambar 3.8 *Activity Diagram* Aplikasi *BackEnd*-Cerita

Gambar 3.8, menggambarkan aktivitas pengolahan data cerita. Proses ini berjalan setelah admin berhasil melakukan login. Admin memilih menu cerita, selanjutnya Web akan menampilkan halaman cerita.

Admin dapat mengelola data kategori dengan menggunakan aksi Tambah, Ubah, atau Hapus. Web akan menampilkan halaman berdasarkan aksi yang dilakukan admin, saat bersamaan *query* akan dipanggil dan dieksekusi pada database server kemudian hasilnya ditampilkan pada web.

5. *Activity Diagram* Aplikasi *BackEnd*-User



Gambar 3.9 *Activity Diagram* Aplikasi *BackEnd*-User

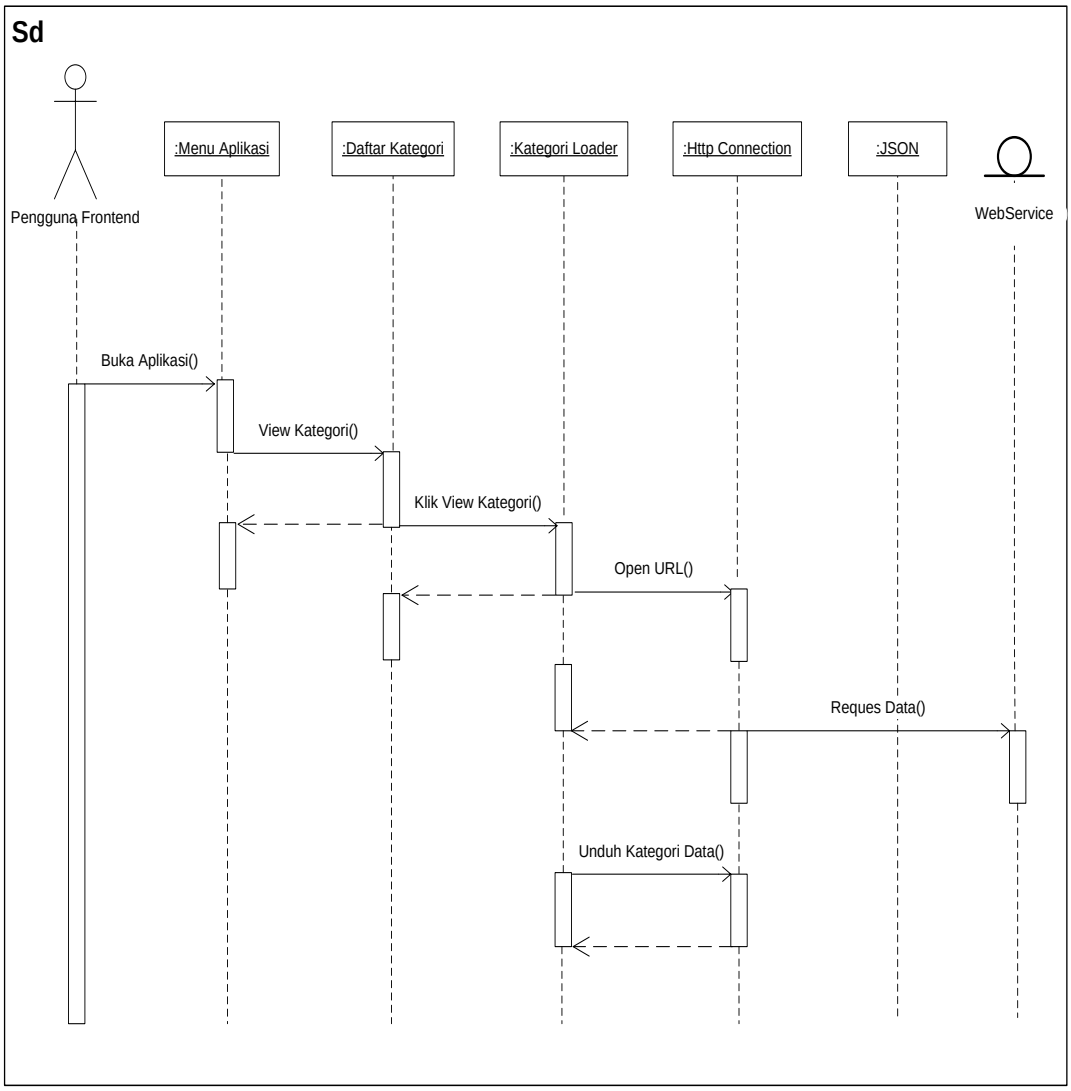
Gambar 3.9, menggambarkan aktivitas pengolahan data user. Proses ini berjalan setelah admin berhasil melakukan login. Admin memilih menu user, selanjutnya Web akan menampilkan halaman user. Admin dapat mengelola data kategori dengan menggunakan aksi Tambah, Ubah, atau Hapus. Web akan menampilkan halaman berdasarkan aksi yang dilakukan admin, saat bersamaan *query* akan dipanggil dan dieksekusi pada database server kemudian hasilnya ditampilkan pada web.

3) *Sequence* Diagram

Sequence diagram menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam use case.

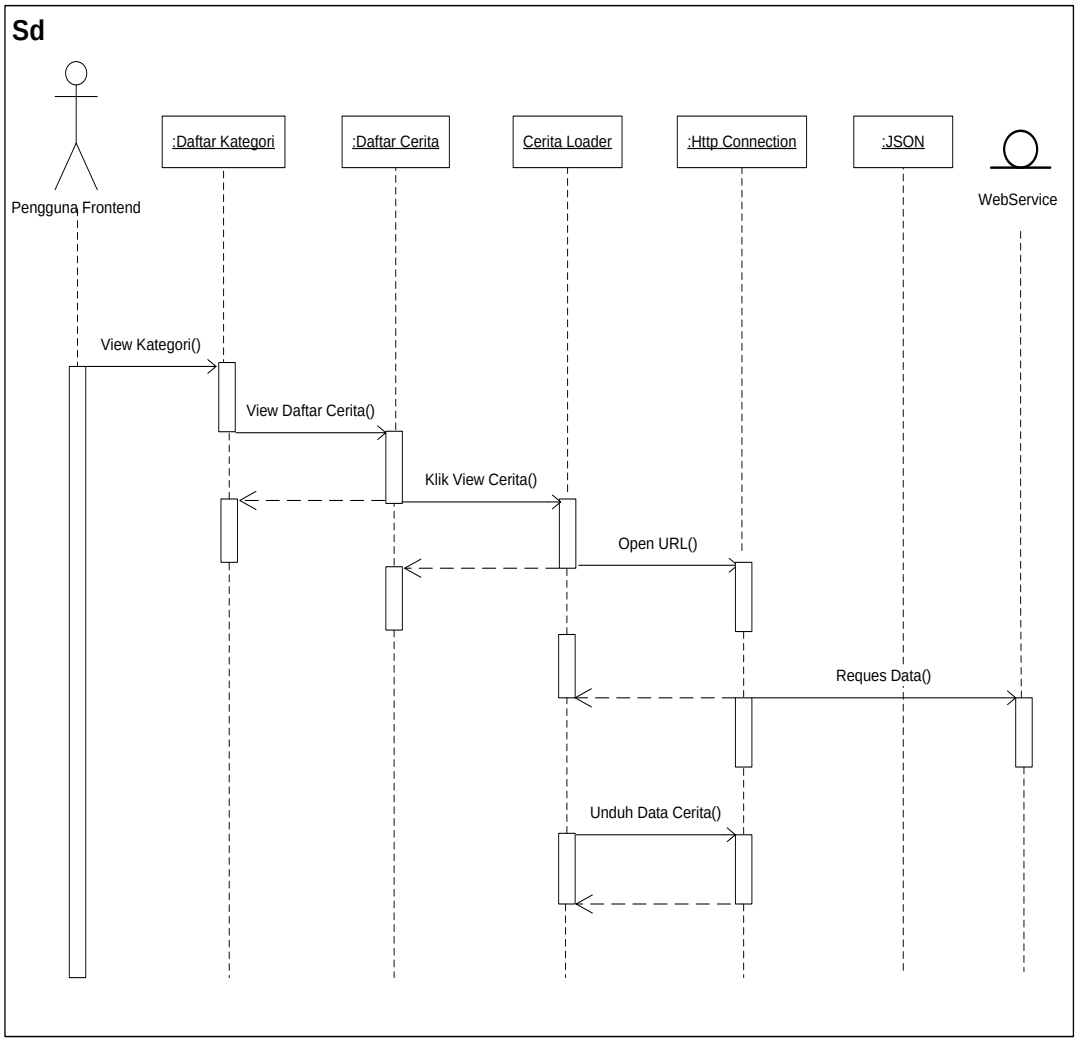
Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progres vertikal.

1. *Sequence Diagram Frotend*- Pilih Kategori



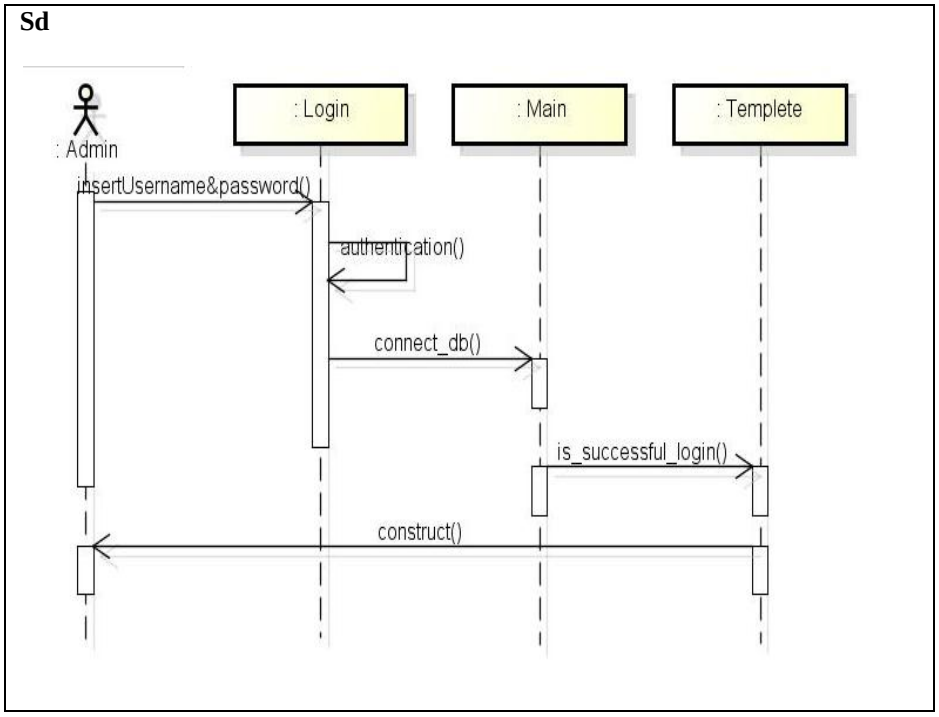
Gambar 3.10 *Sequence Diagram Frotend*- Pilih Kategori

2. *Sequence Diagram Frontend*-Pilih Cerita



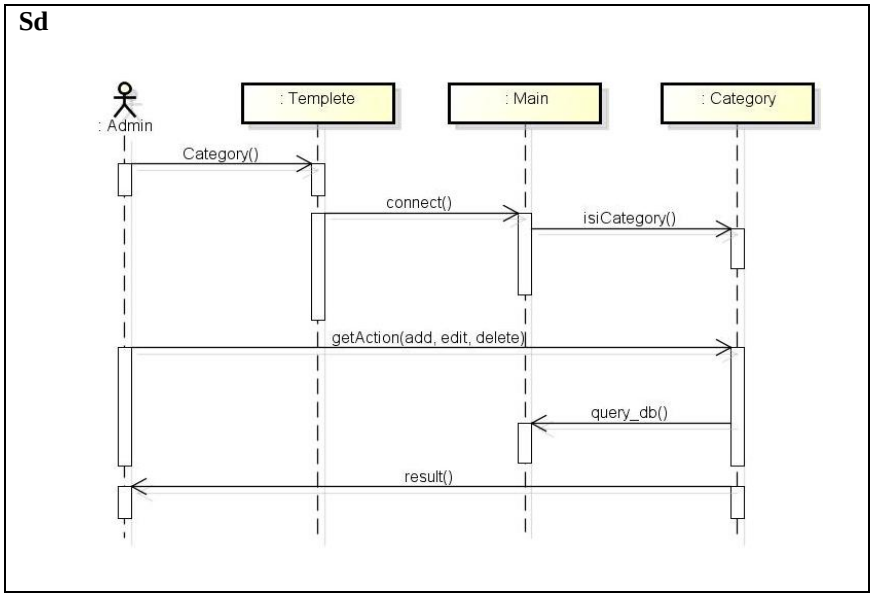
Gambar 3.11 Sequence Diagram *Frontend*-Pilih Cerita

3. *Sequence Diagram Backend*-Login



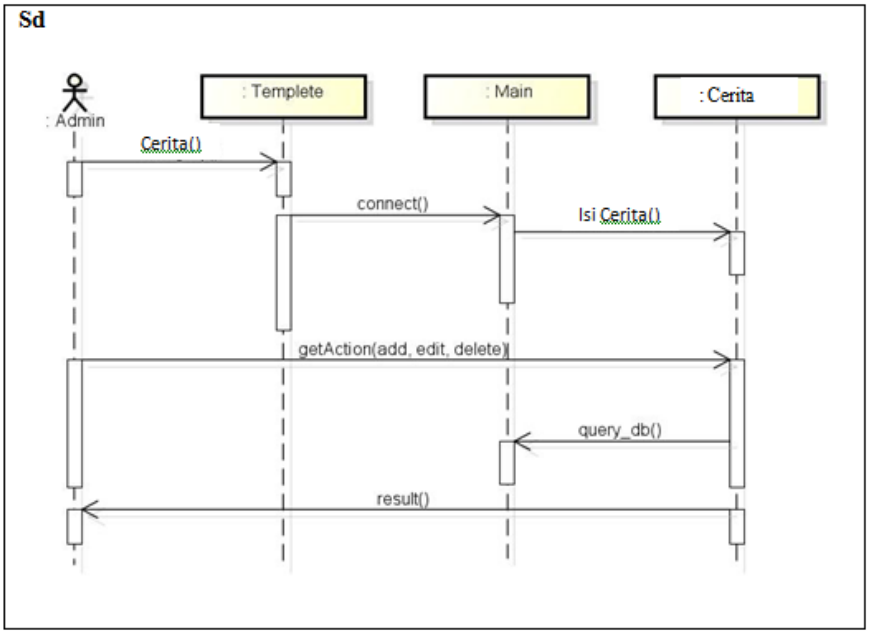
Gambar 3.12 *Sequence Diagram Backend*-Login

4. *Sequence Diagram Backend*-Kategori



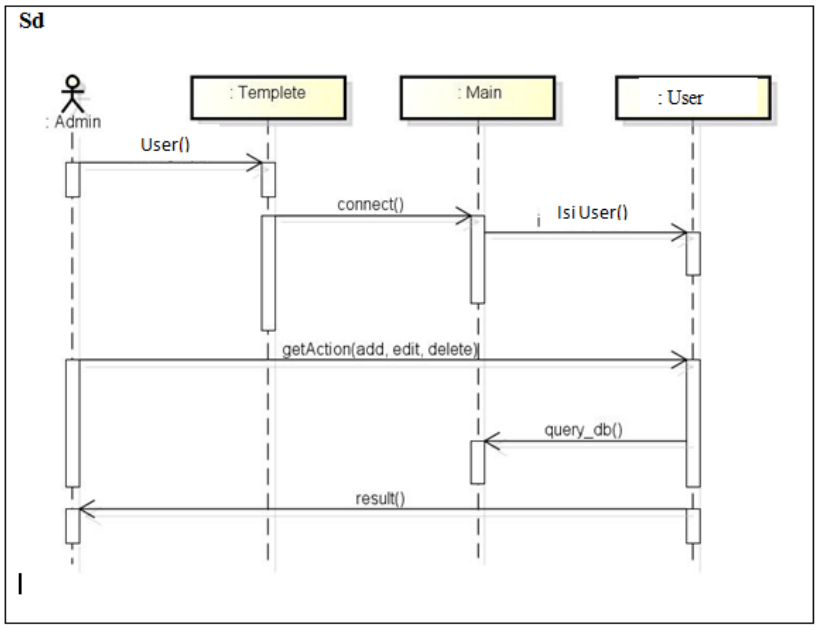
Gambar 3.13 *Sequence Diagram Backend*- Kategori

5. *Sequence Diagram Backend*-Cerita



Gambar 3.14 *Sequence Diagram Backend*- Cerita

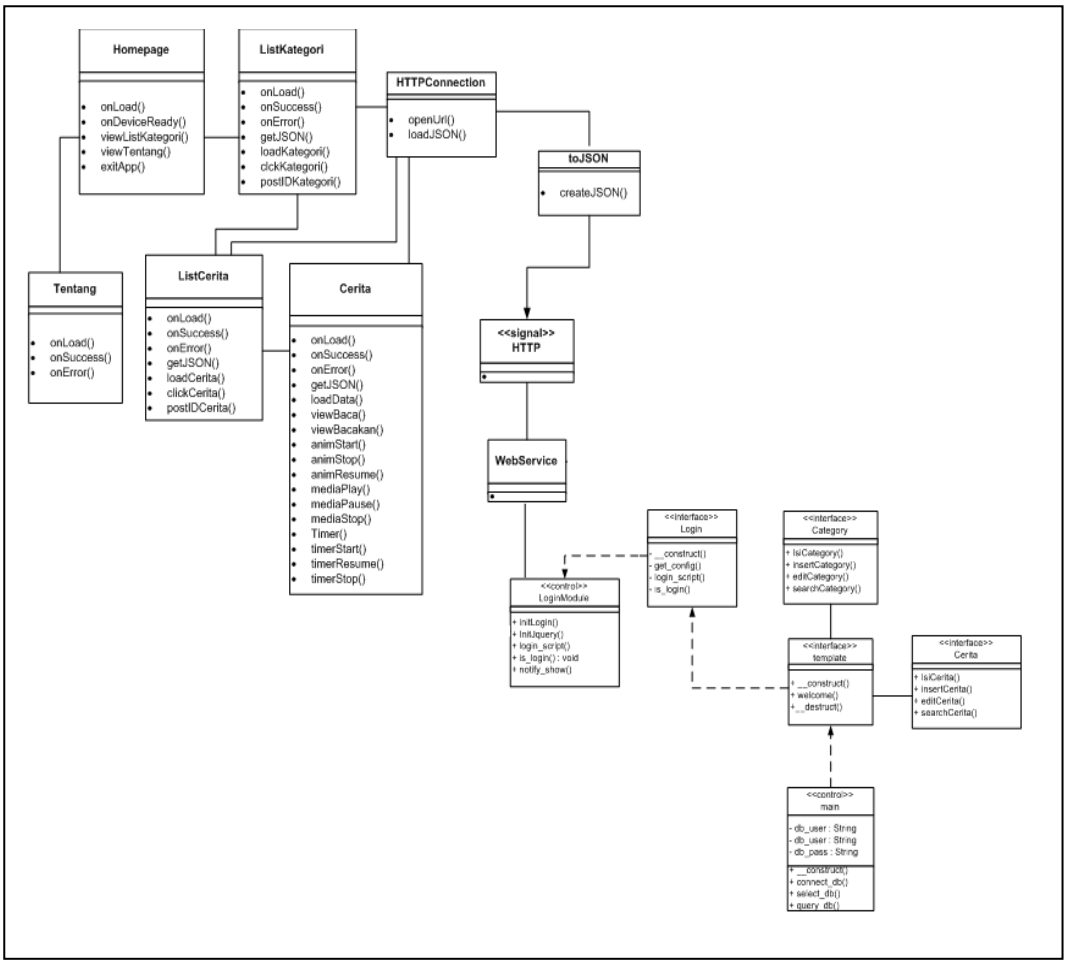
6. *Sequence Diagram Backend*-User



Gambar 3.15 *Sequence Diagram Backend*- User

4) Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*attribut* atau *property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi).



Gambar 3.16 *Class Diagram*

Tabel 3.22 Deskripsi *Class Diagram Perpustakaan Cerita Rakyat*

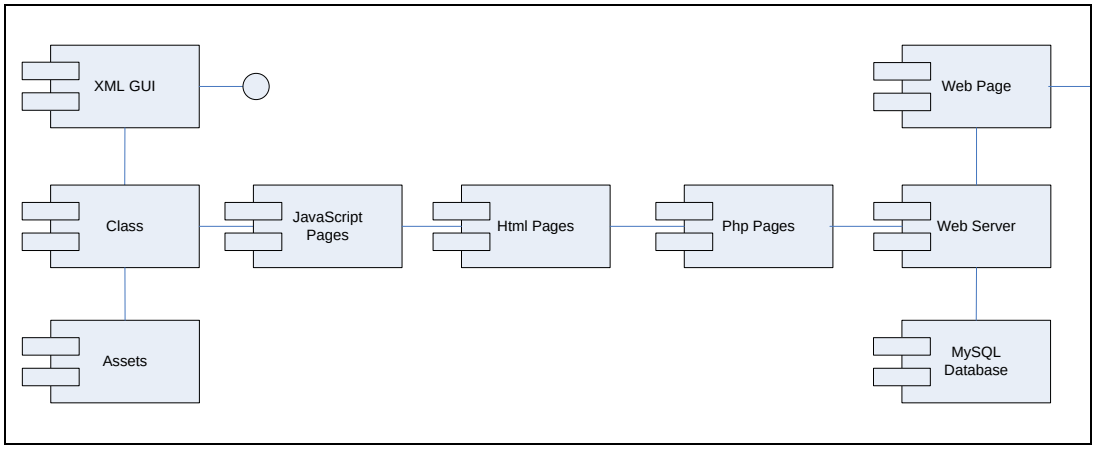
Backend Application		
Class	**Jenis Class**	**Deskripsi**
Category	*Interface*	*Class* yang berisikan tampilan data kategori
Cerita	*Interface*	*Class* yang berisikan tampilan data cerita
Template	*Interface*	*Class* yang berisikan menu navigator web
Login	*Interface*	*Class* yang berisikan tampilan login web
Main	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class interface*
LoginModule	*Control*	*Class* yang berisikan *method-method* untuk mengontrol *class interface* Login
Frontend Application		
Class	**Jenis Class**	**Deskripsi**
HTTPConnection	*Control*	Class yang berisikan *method-method* untuk menghubungkan aplikasi dengan webservice
JSON	*Control*	Class yang berisikan *method-method* untuk fungsionalitas menyusun urutan penanda.
Homepage	*Interface*	Class yang berisikan *method-method* untuk

		fungsionalitas menampilkan halaman utama
List Kategori	*Interface*	Class yang berisikan *method-method* untuk fungsionalitas menampilkan list Kategori
List Cerita	*Interface*	Class yang berisikan *method-method* untuk fungsionalitas menampilkan list Cerita
Cerita	*Interface*	Class yang berisikan *method-method* untuk fungsionalitas menampilkan isi Cerita
Tentang	*Interface*	Class yang berisikan *method-method* untuk fungsionalitas menampilkan halaman tentang aplikasi

5) *Component* Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) diantaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

Komponen dapat juga berupa *interface*, yaitu kompulan layanan yang disediakan sebuah komponen untuk komponen lain.

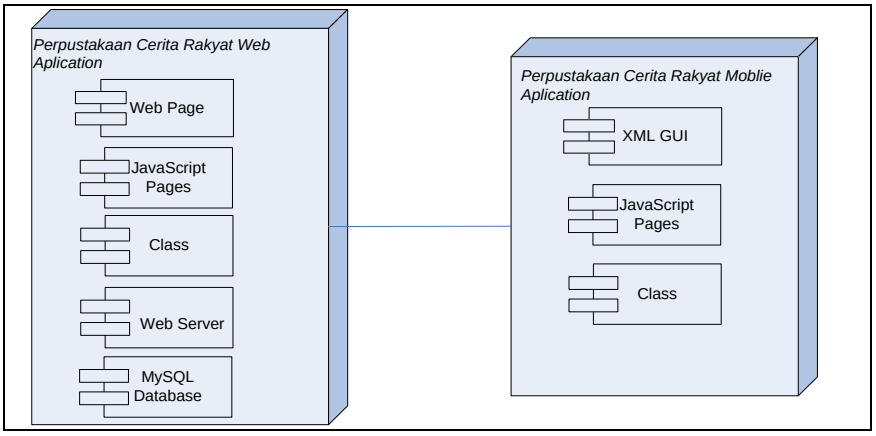


Gambar 3.17 *Component* Diagram

6) Deployment Diagram

Deployment diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal.

Sebuah *node* adalah server, *workstation* , atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.



Gambar 3.18 *Deployment Diagram*

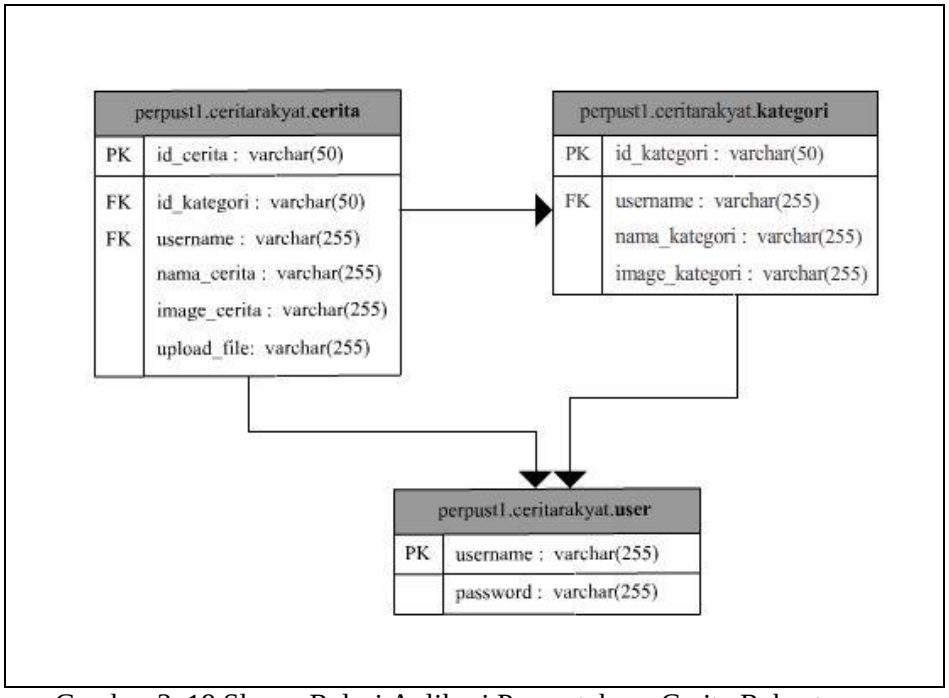
3.2 Perancangan Sistem

Perancangan sistem bertujuan untuk menspesifikasikan aspek-aspek teknik yang menjadi solusi dalam perancangan. Pada tahap ini perancangan akan didefinisikan secara detail untuk mengatasi masalah-masalah yang lebih teknik, berkaitan dengan kegiatan implementasi seperti perancangan arsitektur sistem, perancangan basis data, dan perancangan antarmuka

3.2.1 Skema Relasi

Skema relasi atau relasi antar tabel merupakan gabungan antar file yang mempunyai kunci utama yang sama, sehingga file-file tersebut menjadi satu kesatuan yang dihubungkan oleh field kunci (*primary key*). Pada proses ini elemen-elemen data dikelompokkan menjadi satu file database beserta entitas dan hubungan.

Berikut adalah gambar skema relasi :



Gambar 3. 19 Skema Relasi Aplikasi Perpustakaan Cerita Rakyat

3.2.2 Struktur Tabel

1) Tabel Kategori

Tabel 3.23 Struktur Tabel Kategori

Nama Field	Tipe Data	Panjang	Keterangan
Id_kategori	Varchar	50	Primary Key
nama_kategori	Varchar	255	
image_kategori	Varchar	255	

2) Tabel Cerita

Tabel 3.24 Struktur Tabel Cerita

Nama Field	Tipe Data	Panjang	Keterangan
Id_cerita	Varchar	50	Primary Key
Id_kategori	Varchar	50	Foreign Key references Category(category_id)
nama_cerita	Varchar	255	
image_cerita	Varchar	255	
Upload_file	Varchar	255	

3) Tabel *User*

Tabel 3.25 Struktur Tabel *User*

Nama Field	Tipe Data	Panjang	Keterangan
username	Varchar	255	Primary Key
password	Varchar	255	

3.2.3 Perancangan Grafis dan Antarmuka

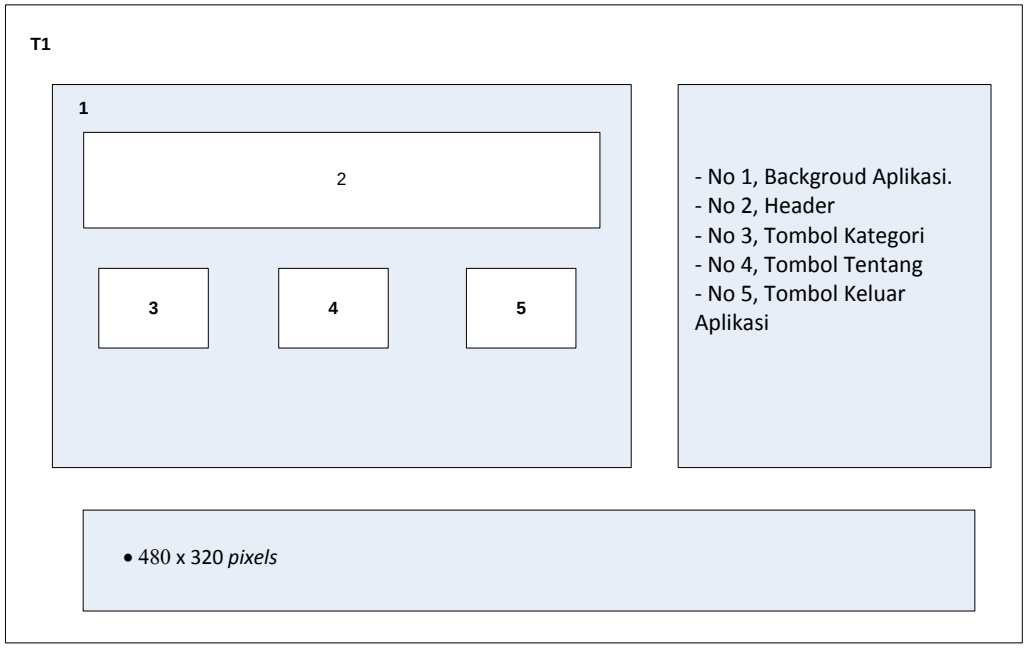
Tahap perancangan desain bertujuan untuk mencari bentuk yang optimal dari aplikasi yang akan dibangun dengan pertimbangan faktor-faktor permasalahan dan kebutuhan yang ada pada sistem seperti yang telah ditetapkan pada tahap analisis. Dalam tahap ini upaya yang dilakukan yaitu dengan cara mengkombinasikan

penggunaan teknologi perangkat keras dan perangkat lunak yang tepat sehingga diperoleh desain yang optimal dan mudah di implementasikan.

Perancangan grafis dan antarmuka terdiri dari dua bagian utama yaitu desain grafis dan antarmuka aplikasi *frontend* dan *backend.*

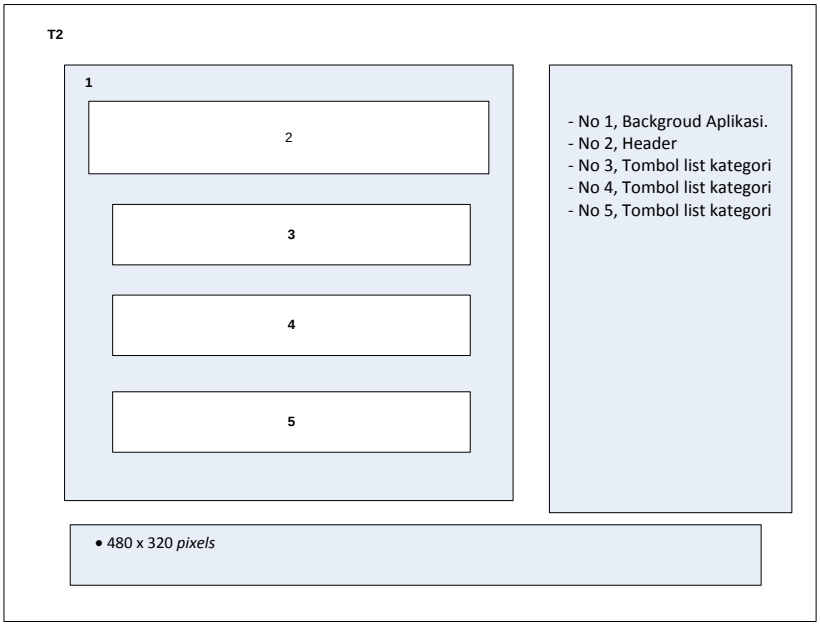
3.2.3.1 Perancangan Antarmuka Aplikasi *Frontend*

1) Perancangan Antarmuka Halaman Utama



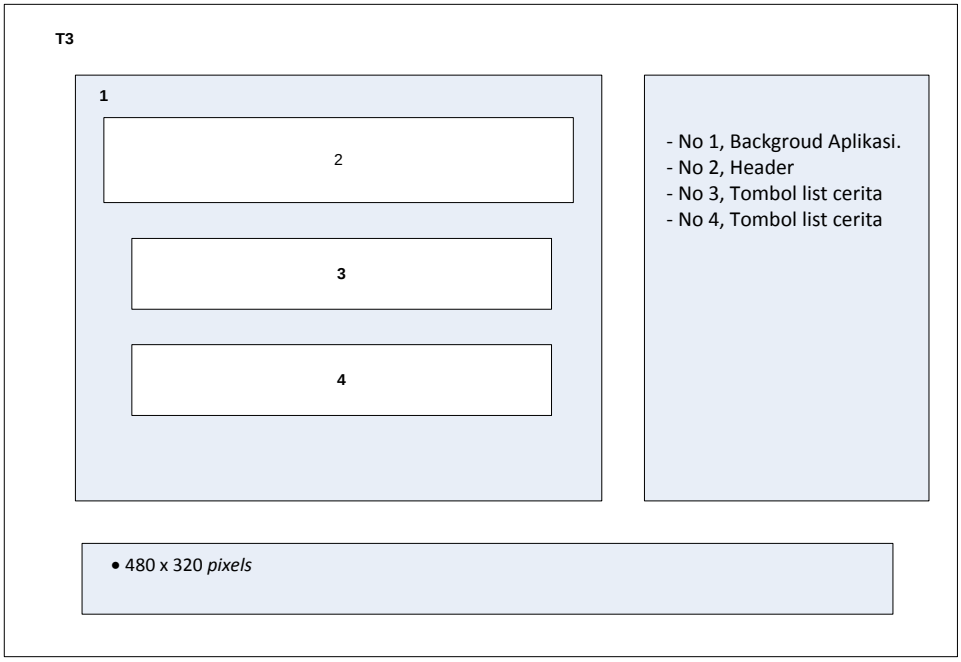
Gambar 3.20 Perancangan Antarmuka Halaman Utama

2) Perancangan Antarmuka Pilih Kategori



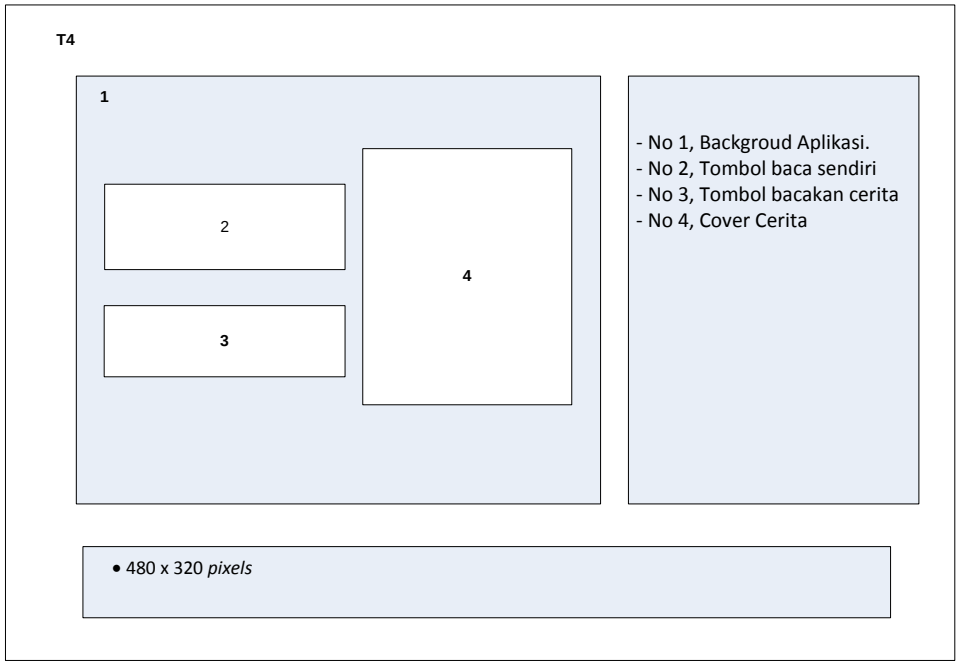
Gambar 3.21 Perancangan Antarmuka Pilih Kategori

3) Perancangan Antarmuka Pilih Cerita



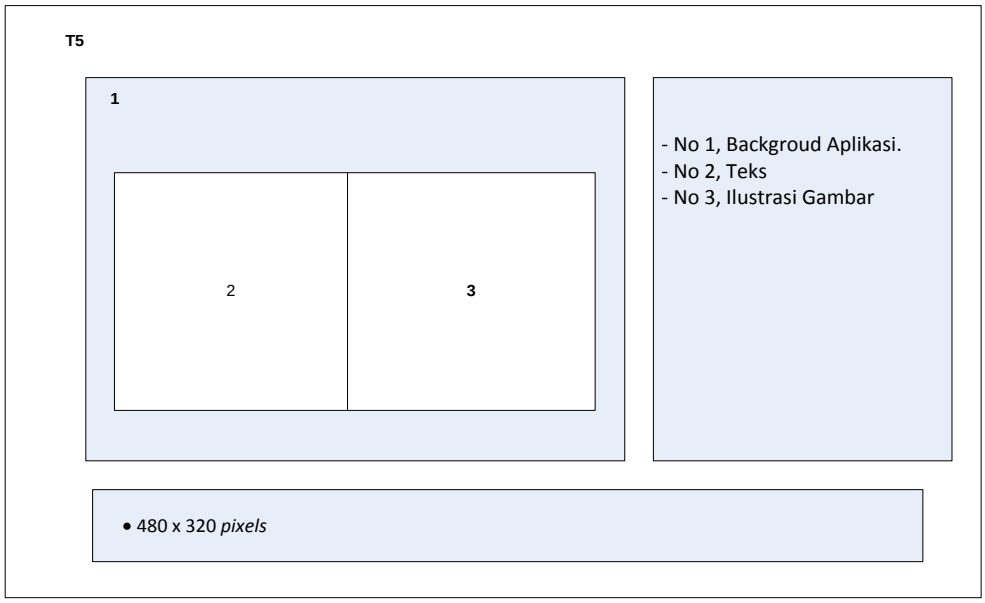
Gambar 3.22 Perancangan Antarmuka Pilih Cerita

4) Perancangan Antarmuka Lihat Cerita



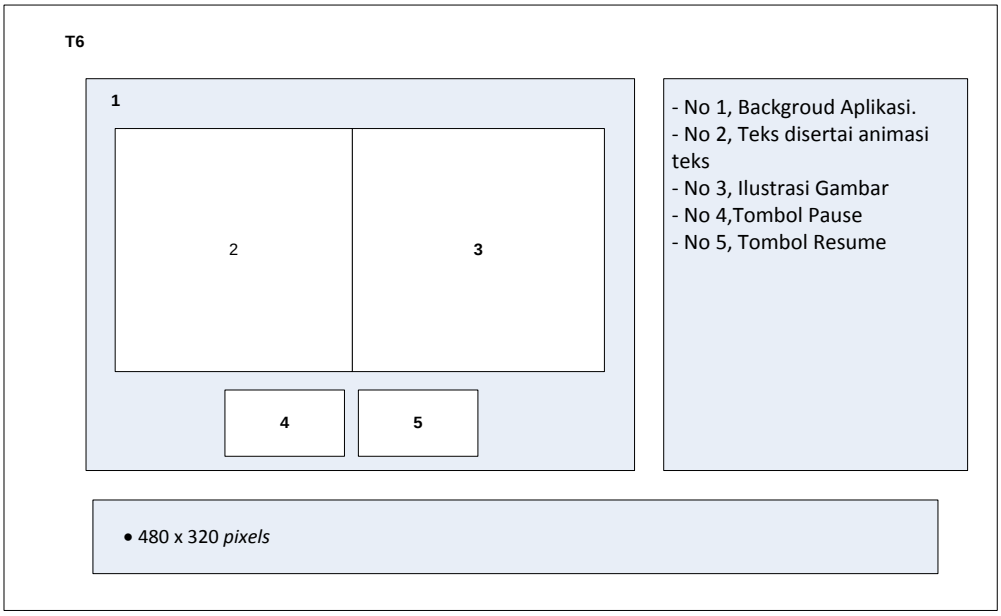
Gambar 3.23 Perancangan Antarmuka Lihat Cerita

5) Perancangan Antarmuka Isi Cerita-Baca Sendiri



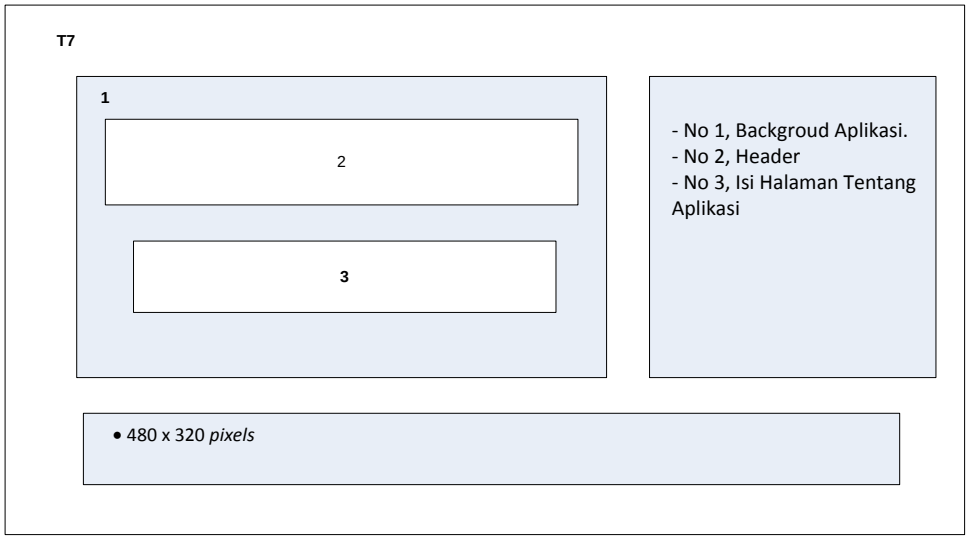
Gambar 3.24 Perancangan Antarmuka Isi Cerita-Baca Sendiri

6) Perancangan Antarmuka Isi Cerita-Bacakan



Gambar 3.25 Perancangan Antarmuka Isi Cerita-Bacakan

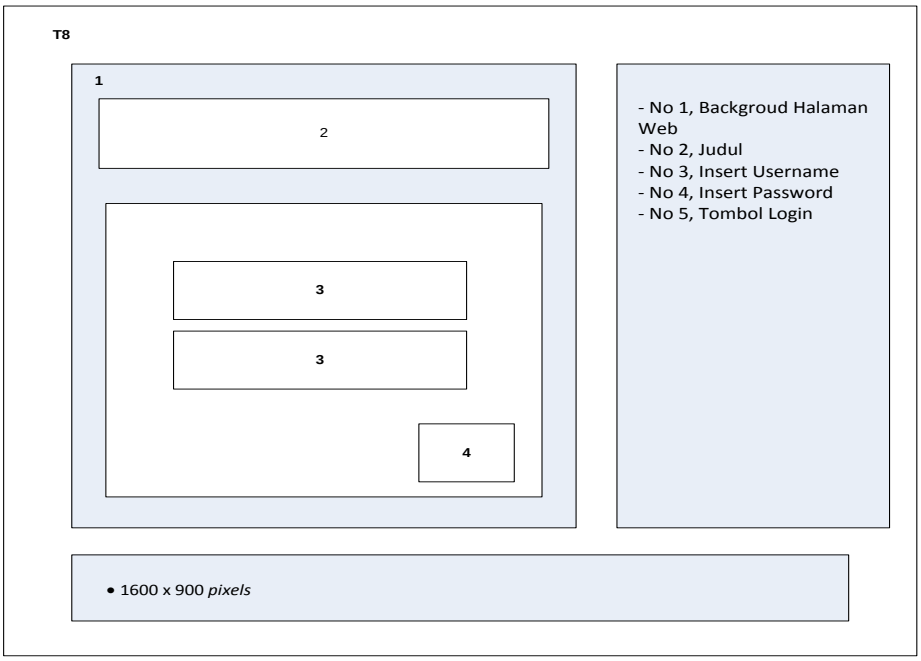
7) Perancangan Antarmuka Pilih Tentang



Gambar 3.26 Perancangan Antarmuka Pilih Tentang

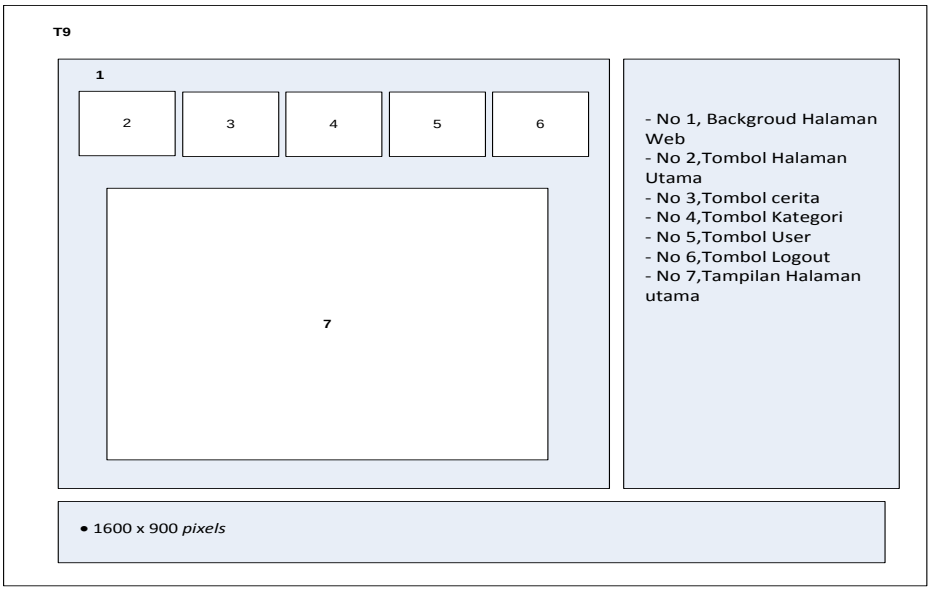
3.2.3.2 Perancangan Antarmuka Aplikasi *Backtend*

1) Perancangan Antarmuka Halaman Login



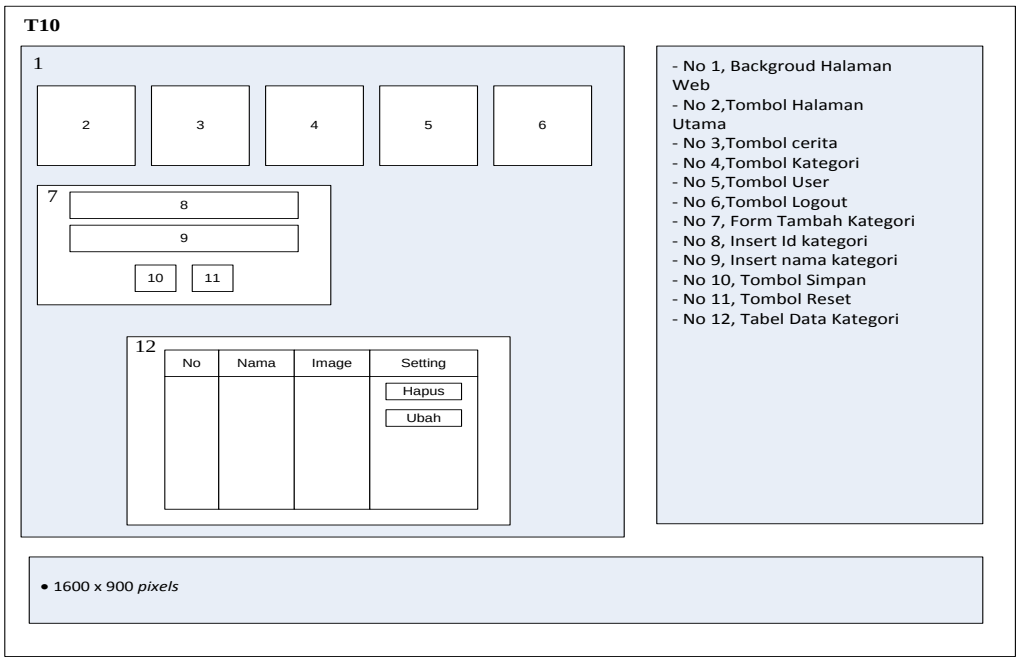
Gambar 3.27 Perancangan Antarmuka Halaman Login

2) Perancangan Antarmuka Halaman Utama



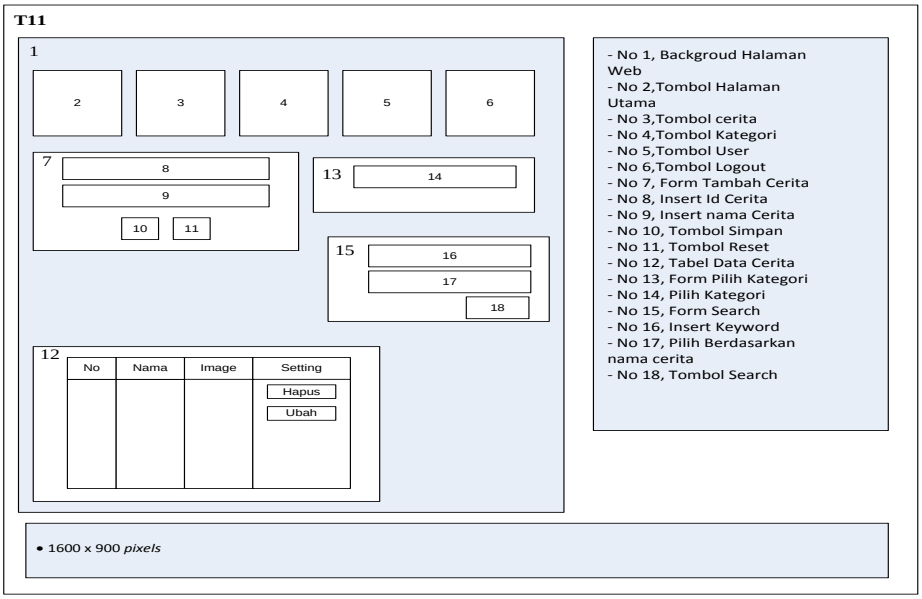
Gambar 3.28 Perancangan Antarmuka Halaman Utama

3) Perancangan Antarmuka Pilih Kategori



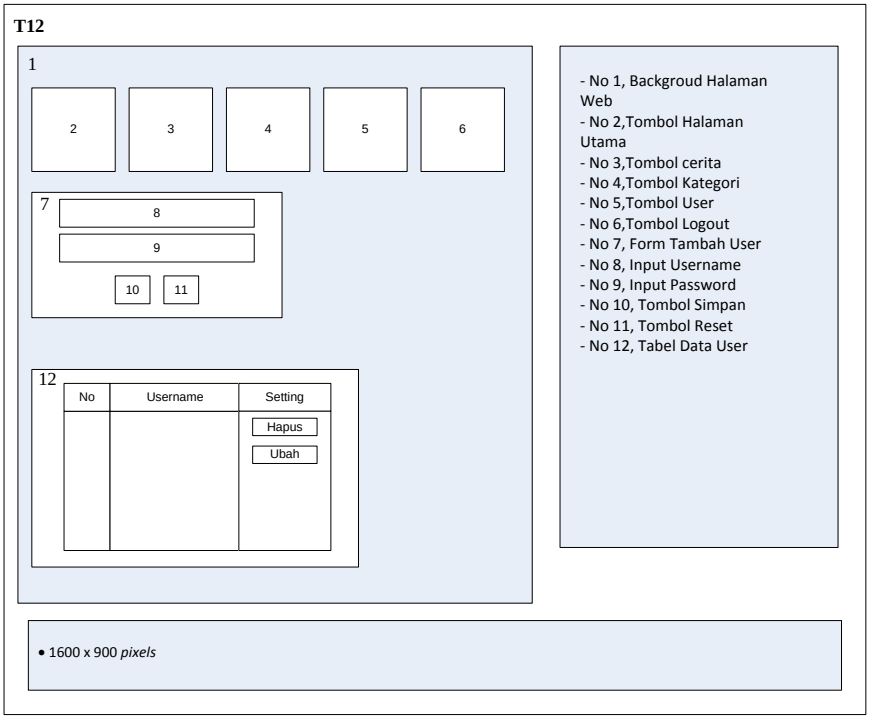
Gambar 3.29 Perancangan Antarmuka Pilih Kategori

4) Perancangan Antarmuka Pilih Cerita



Gambar 3.30 Perancangan Antarmuka Pilih Cerita

5) Perancangan Antarmuka Pilih User



Gambar 3.31 Perancangan Antarmuka Pilih User