# BAB III ANALISIS DAN PERANCANGAN

## Analisis

### Identifikasi Masalah

Dalam pengelolaan konten digital signage, ditemukan sejumlah permasalahan yang mempengaruhi penggunaan efektif aplikasi Xibo. Salah satu permasalahan utama terletak pada kompleksitas alur kerja yang diperlukan dalam proses pengelolaan konten. Alur kerja ini melibatkan serangkaian langkah yang kompleks, seperti mengunggah manual konten seperti gambar atau video, menyusun konten dalam playlist, mendesain layout yang menarik, menghubungkan layout dengan playlist yang sesuai, dan menjadwalkan tampilan pada perangkat pemutar Xibo. Pengguna juga menghadapi kesulitan dalam menjelajahi antarmuka Xibo, terutama karena struktur menu dan submenu yang rumit, sehingga menghambat kemudahan dalam menemukan fungsi yang diinginkan. Dampak dari permasalahan ini termasuk penurunan efisiensi dalam pengelolaan konten, potensi terjadinya kesalahan dalam mengatur tampilan, dan adanya rasa frustrasi di pihak pengguna.

### Pemecahan Masalah

Dalam menghadapi kerumitan alur kerja di aplikasi Xibo untuk pengelolaan konten digital signage, solusi yang ditawarkan meliputi pengembangan sebuah aplikasi middleware berbasis website. Aplikasi ini bertujuan untuk mengatasi kesulitan yang ditemui pengguna dalam langkah-langkah kompleks dalam Xibo. Dalam perbandingan dengan alur kerja pada aplikasi Xibo, berikut adalah perbandingan langkah-langkah utama antara kedua aplikasi:

1. Xibo



1. Middleware

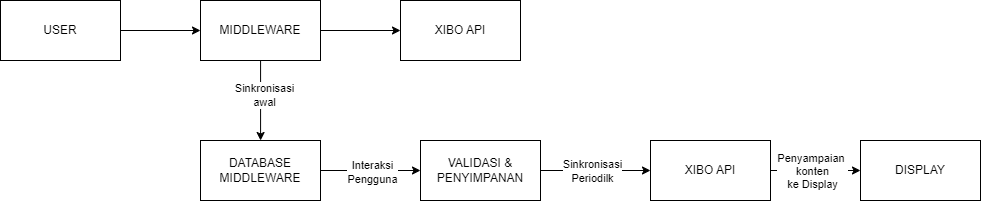


Dalam perbandingan diatas , dapat dilihat bahwa alur kerja yang diterapkan oleh Aplikasi Middleware terlihat lebih sederhana dan terpusat pada tahapan-tahapan inti. Hal ini berpotensi mengurangi kompleksitas yang dihadapi oleh pengguna dalam mengelola konten digital signage. Dengan mengeliminasi beberapa tahap yang ditemukan dalam Aplikasi Xibo.

## Perancangan

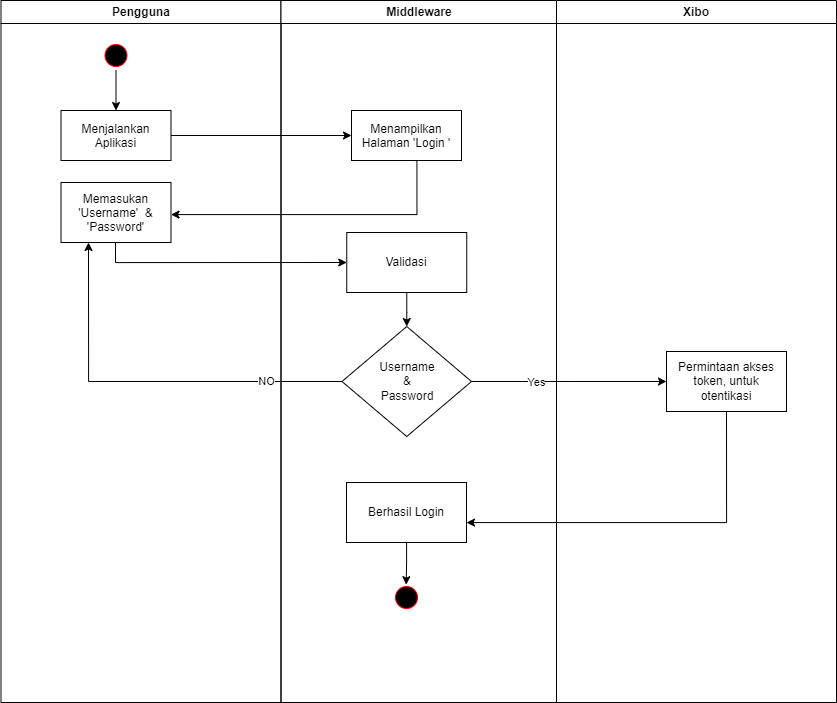
### Perancangan Sistem

1. **Alur Program**



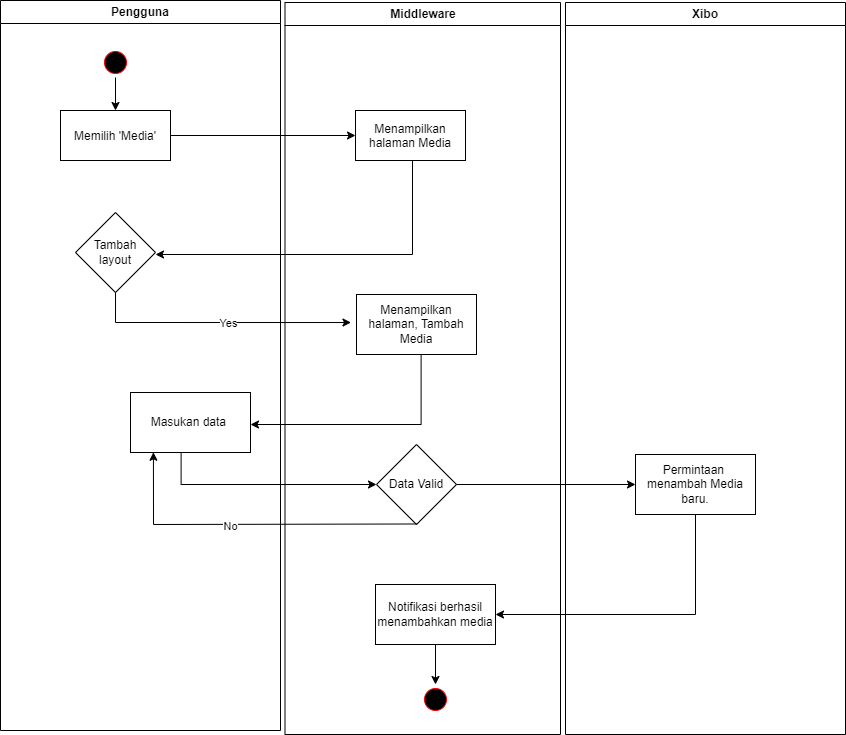
Gambar 3. Gambar Cara Kerja Middlware

Gambar diatas merupakan gambaran dari pengguna berinteraksi dengan *middleware* melalui antarmuka pengguna. *Middleware* melakukan sinkronisasi awal dengan Xibo API untuk mengambil data awal dan menyimpannya dalam database middleware. Pengguna memberikan *input* yang akan divalidasi dan disimpan dalam database. *Middleware* melakukan sinkronisasi periodik (perbahuruan berkala) dengan Xibo API untuk memperbarui data di *middleware* berdasarkan perubahan yang terjadi di Xibo. Konten layout dikirimkan ke *display* sesuai dengan jadwal yang telah ditentukan. *Middleware* juga dapat melacak dan memantau penggunaan *display*. Data yang terkait dengan manajemen dan pengaturan lainnya juga disimpan dalam database *middleware*.

1. **Perancangan Alur Program**
   * **Login Activity**

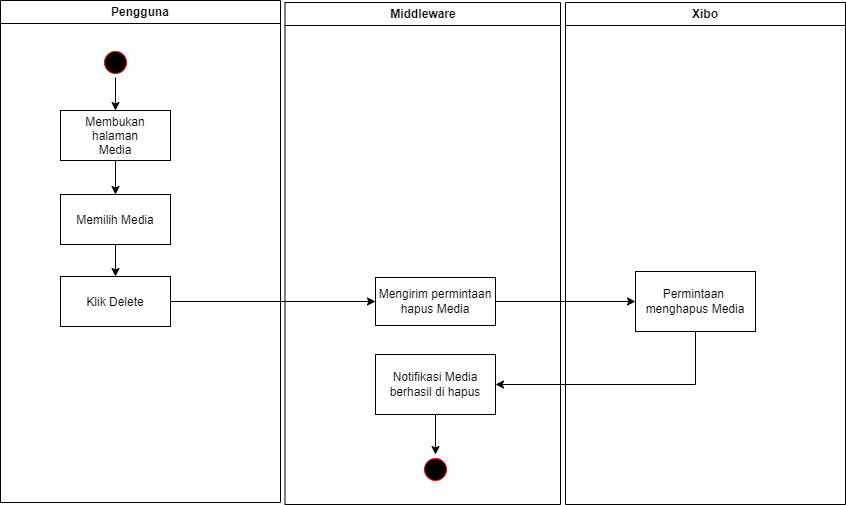
* **Media**

Gambar Diagram aktivitas Login dan permintaan access token

****

Gambar 3.3 Diagram aktivitas penambahan media

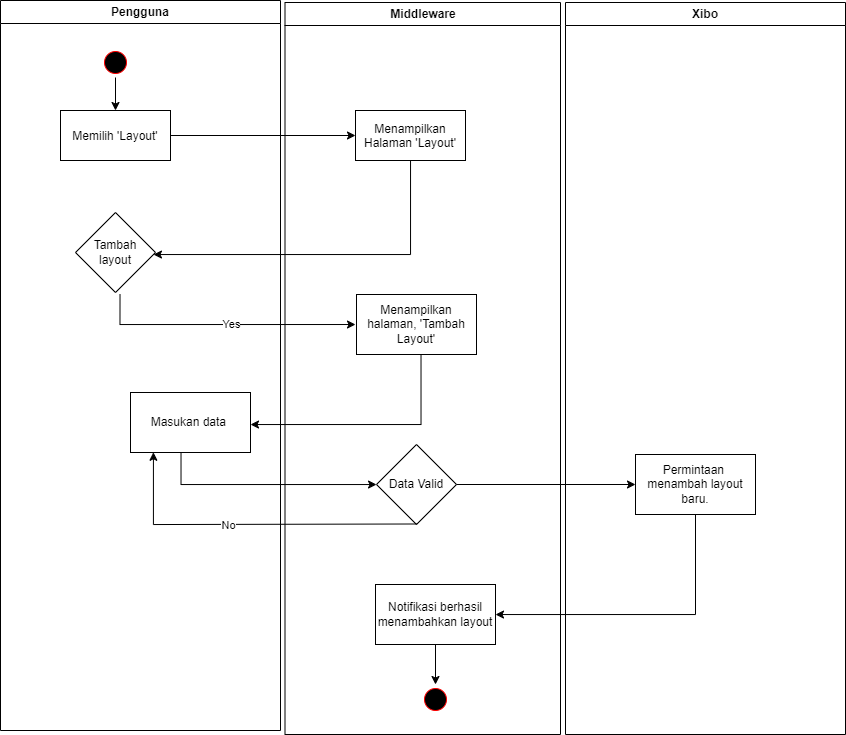
**Hapus Media**

****

Gambar 1. ‑ Diagram aktivitas untuk menghapus media

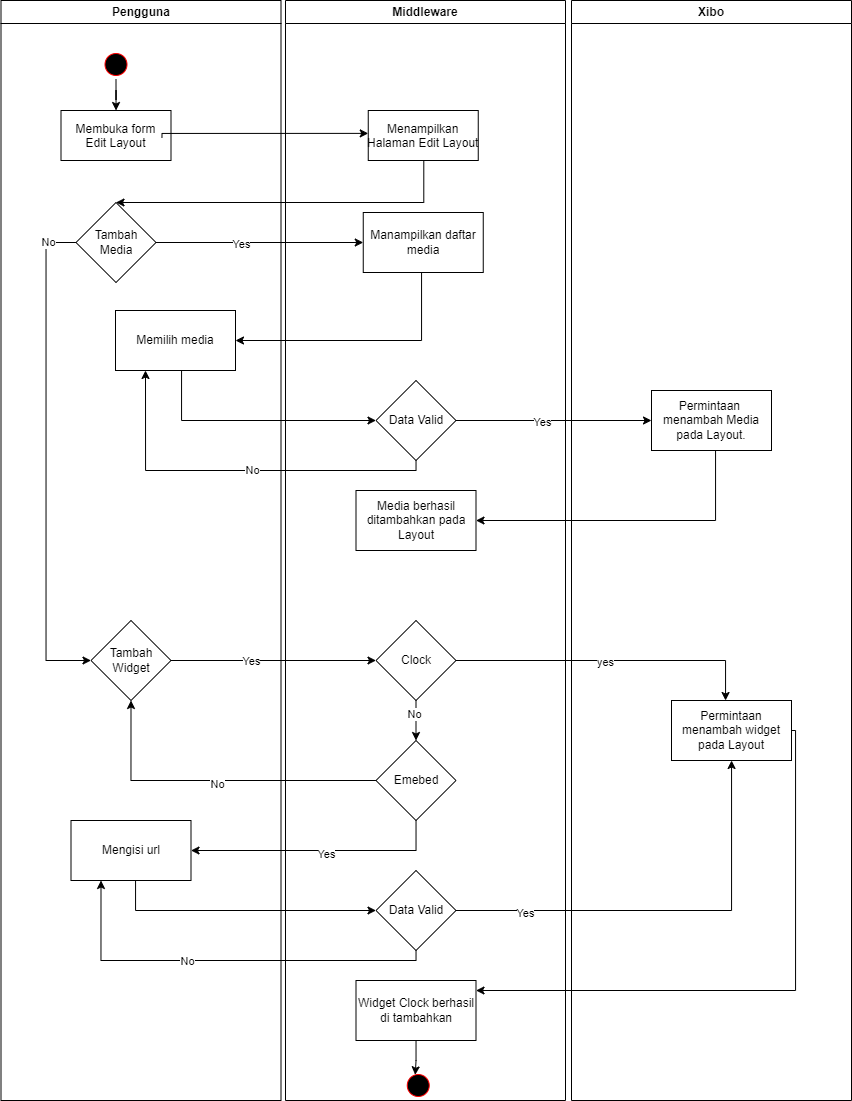
Gambar 1. ‑

Gambar 3.4 Diagram aktivitas menghapus media

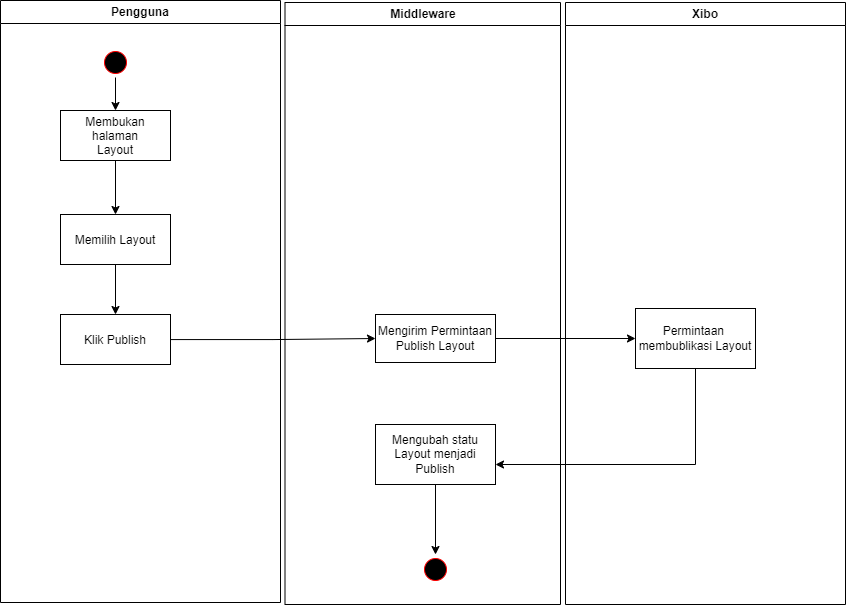
* **Layout**

Gambar 3.5 Diagram aktivitas penambahan layout

**Edit Layout**

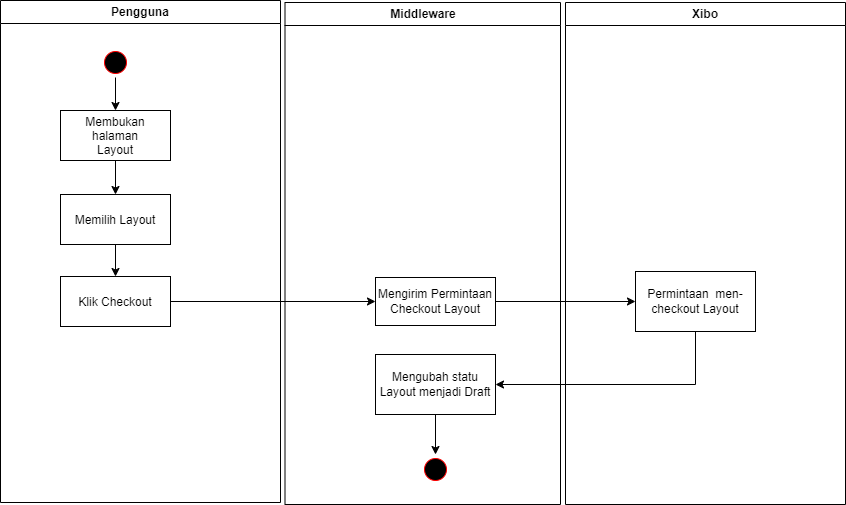
****

Gambar 3.6 Diagram aktivitas pengeditan layout

**Publish Layout**

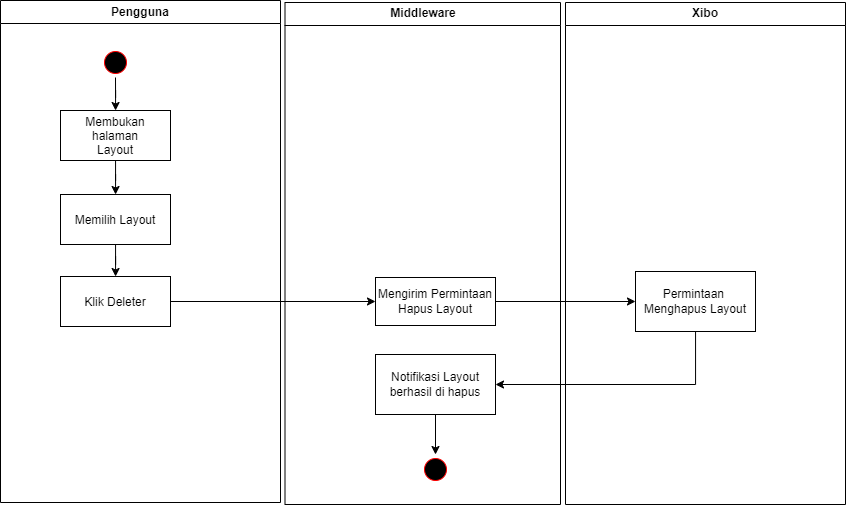
Gambar 3.7 Diagram aktivitas untuk publikasi layout

**Checkout Layout**

****

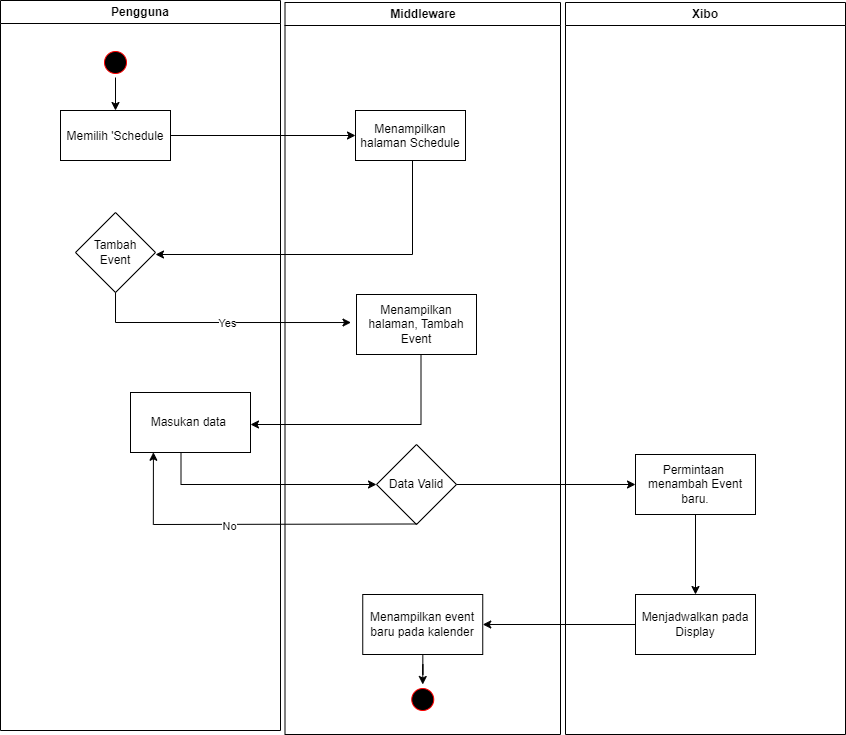
Gambar 3.8 Diagram aktivitas untuk checkout layout

**Delete Layout**

****

Gambar 3.9 Diagram aktivitas penghapusan layout

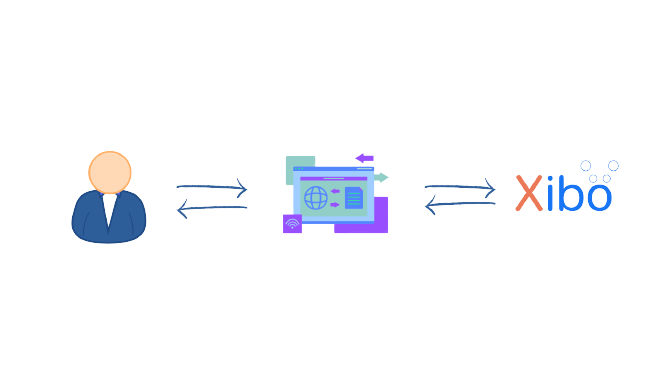
* + **Schedule**

****

Gambar 3.10 Diagram aktivtas pembuatan event baru

### 3.2.2 Perancangan Data

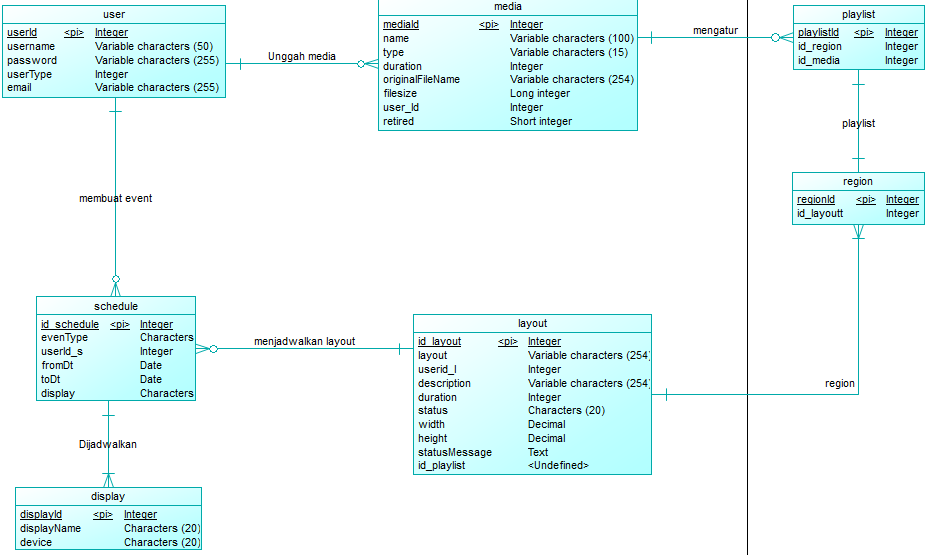
#### **3.2.2.1 Diagram Aliran data.**



Gambar 3.11 Diagram aliran data anatar Client(user), Middleware, dan Xibo

Aliran data dimulai dengan pengguna yang berinteraksi melalui antarmuka pengguna Middleware. Pengguna memberikan input, seperti permintaan pengelolaan konten atau pembuatan layout. Middleware menerima permintaan, memprosesnya, dan berkomunikasi dengan Xibo API menggunakan cURL. Xibo API mengambil data dari basis data Xibo dan merespons kembali ke Middleware. Middleware kemudian memproses respons, melakukan manipulasi data jika diperlukan, dan mengirimkan hasil akhir ke antarmuka pengguna. Pengguna menerima hasil akhir dari permintaan mereka melalui antarmuka pengguna Middleware. Ini menciptakan aliran data yang mulus antara pengguna, Middleware, dan Xibo API dalam mengelola konten dan fungsionalitas Xibo.

***3.2.2.1 Conseptual Data Model***



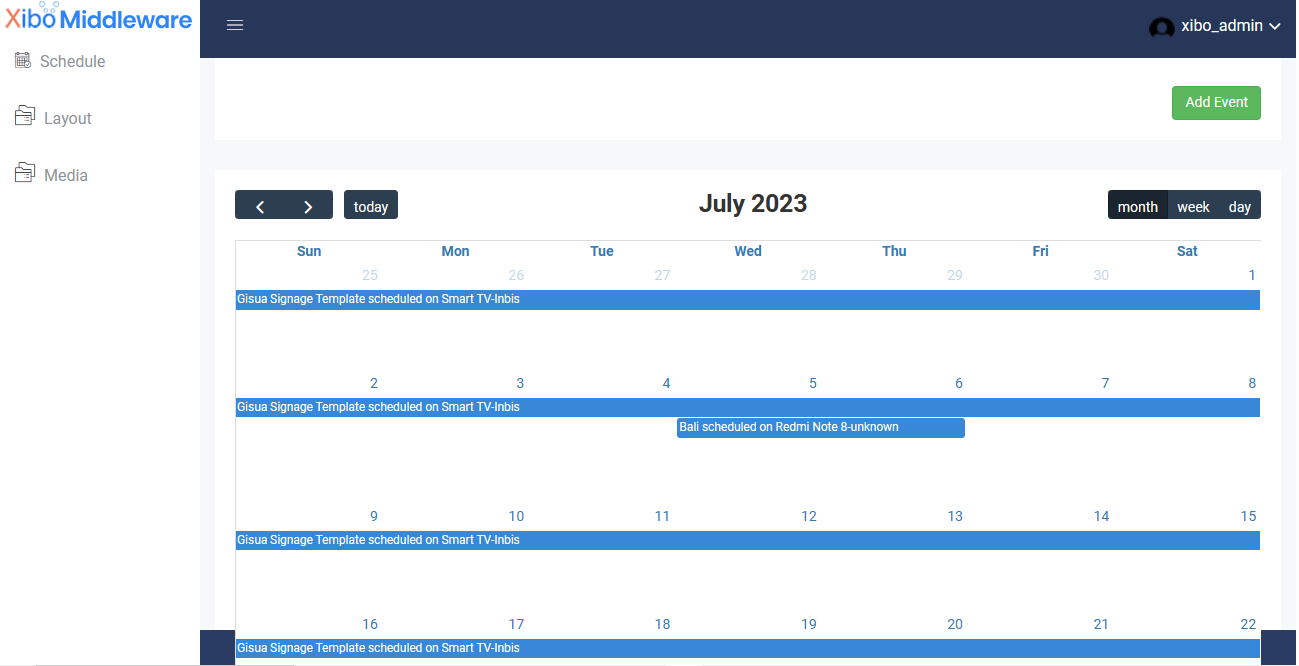
Gambar 3.12 Diagram Conseptual Data Model

### 3.2.3 Perancangan User Interface / *Mock-up* aplikasi



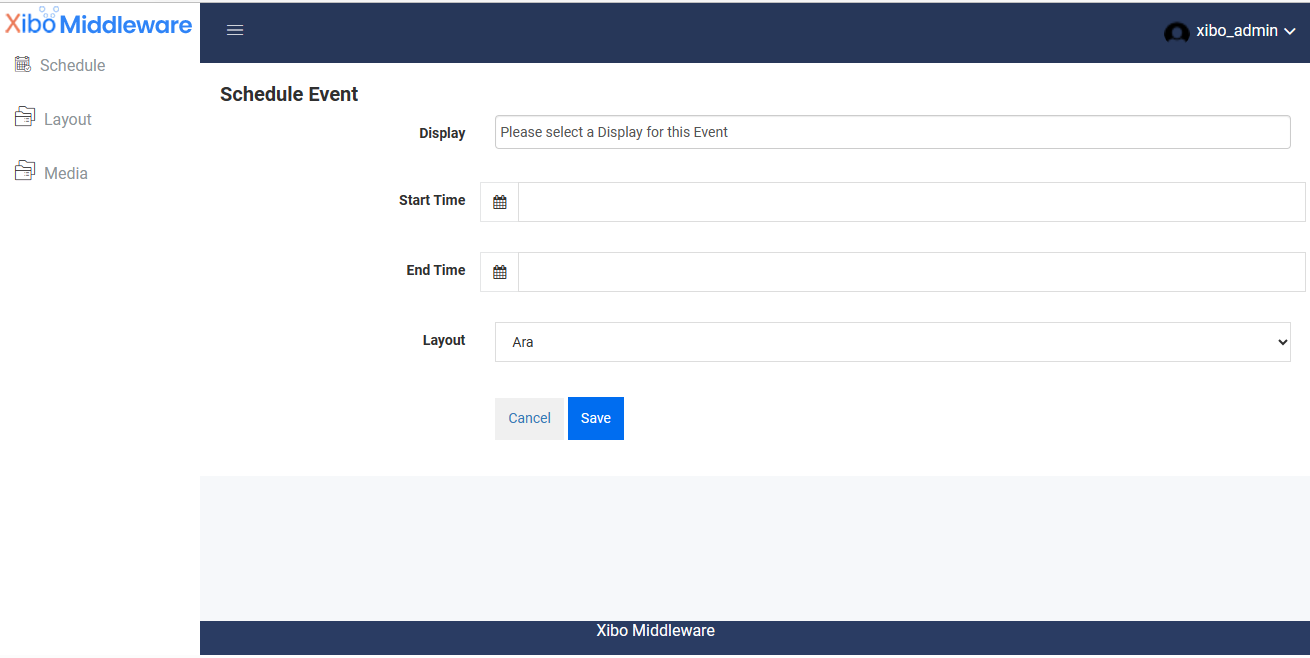
Gambar 3.13 Halaman Login

Gambar diatas merupkan tampilan halaman **Login**. Pada tampilan ini terdapat *username* dan juga *password* yang harus diisi oleh user ketikan akan melakukan proses login untuk masuk kedalam website.



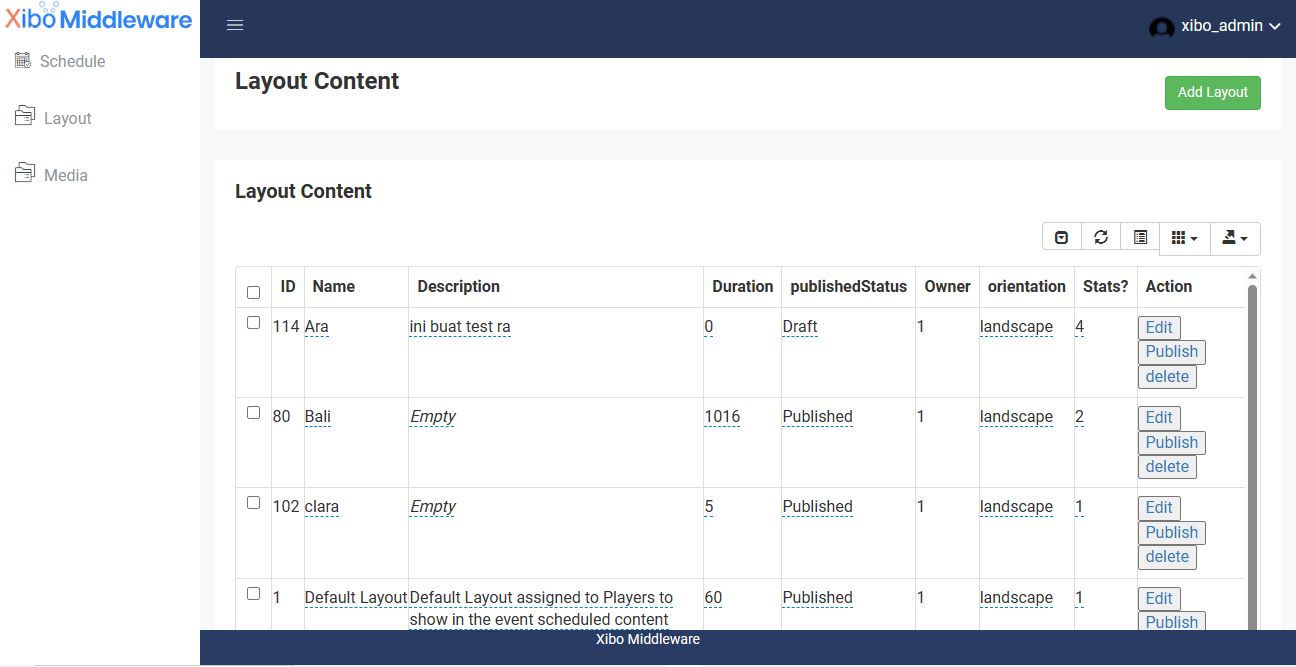
Gambar 3.14 Halaman Schedule

Gambar diatas merupakan tampilan dari halaman **Schedule** atau jadwal. Pada tampilan ini terdapat kalender bulanan yang akan menunjukan event atau kegiatan apa saja yang sudah dijadwalkan. Pada halaman ini juga terdapat button *Add Event* untuk menambahkan event atau kegiatan baru.



Gambar 3.15 Halaman Add Event

Gambar diatas merupakan tampilan dari halaman **Add Even**t. Pada halaman ini terdapat form untuk memilih *display, start time, end time, layout* serta terdapat check box untuk menanyakan user apakah ingin menayangkan layout sesuai dengan waktu CMS atau sesuai waktu dari masing-masing display, jika display berada di tempat dengan waktu yang tidak sama dengan CMS.

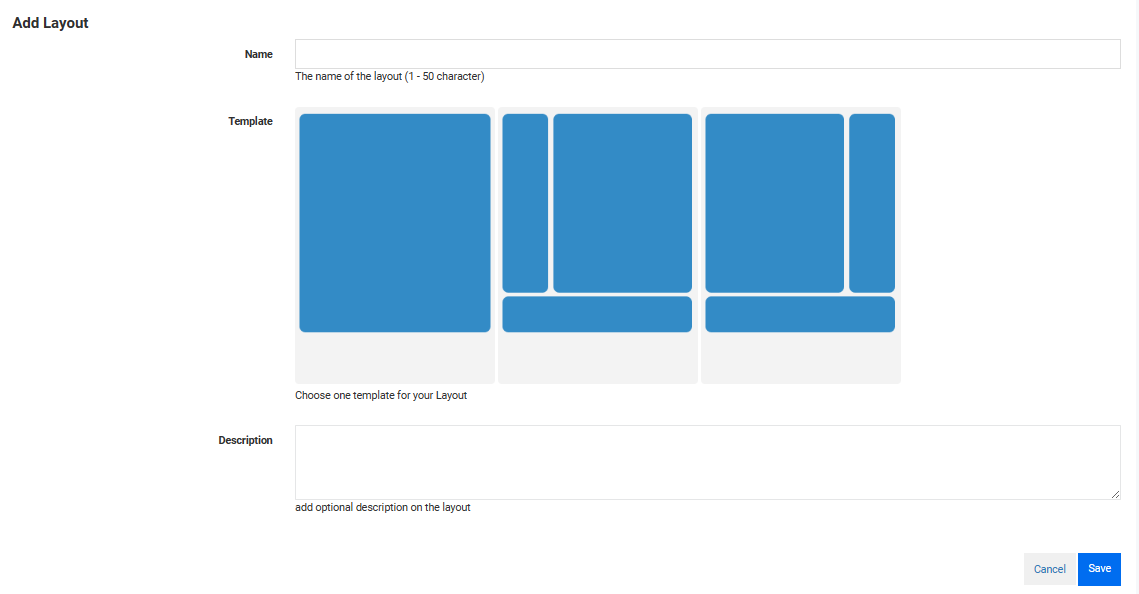
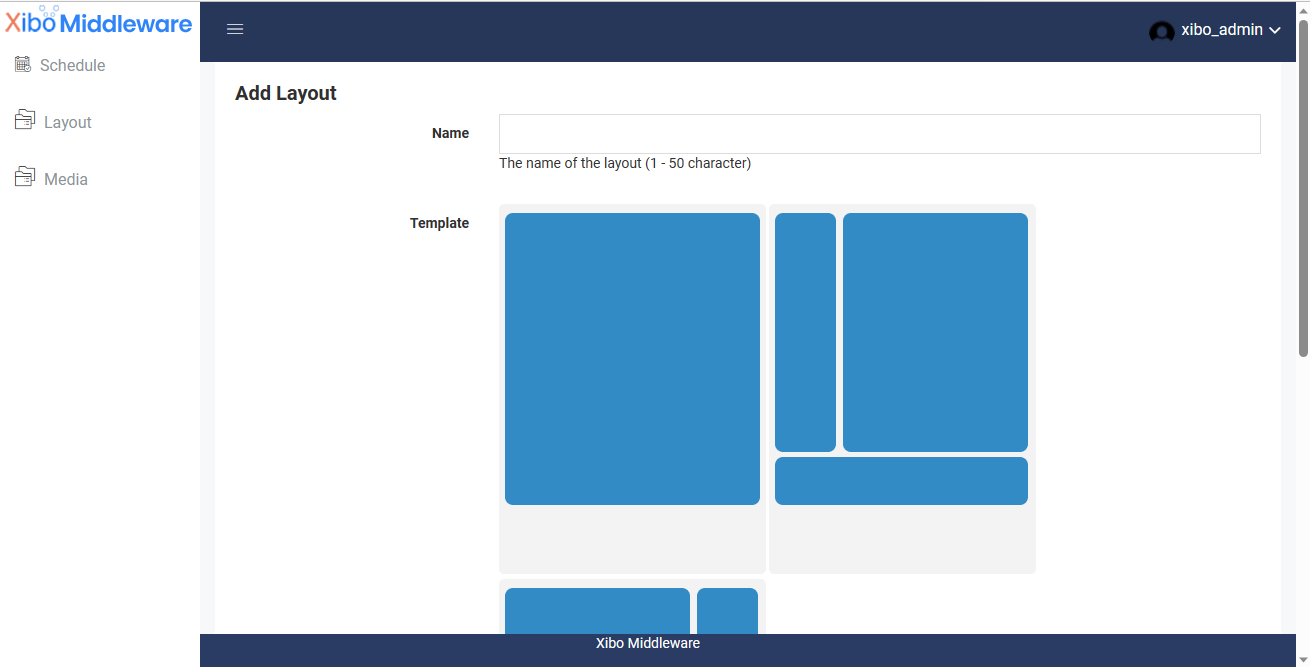


Gambar 3.16 Halaman Layout

Gambar 3.4 Halaman Layout

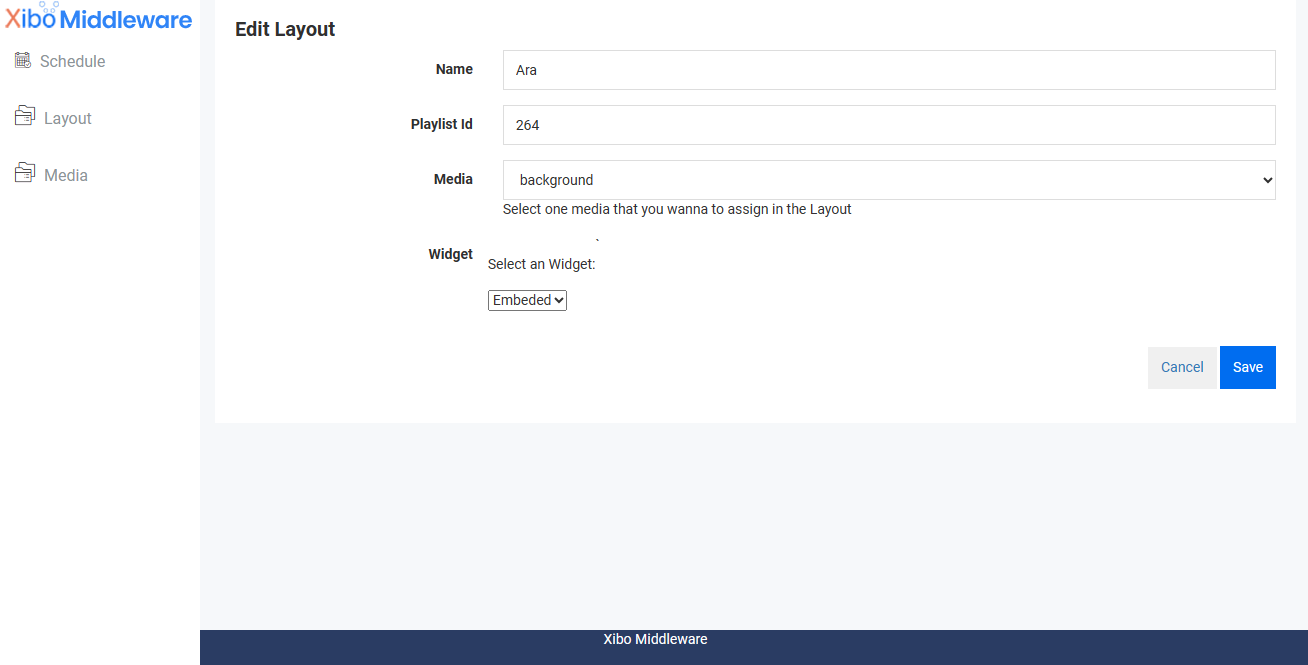
Gambar diatas merupakan tampilan dari halaman **Layout**. Pada halaman ini terdapat tabel yang berisi layout apa saja yang sudah di buat sebelumnya. Pada halaman ini juga terdapat button *Add Layout* untuk menambahkan layout baru, juga tombol untuk edir, publikasi dan hapus.

Gambar 3.17 Halaman Add Layout



Gambar 3. 5 Halaman Add Layout

Gambar diatas merupakan tampilan dari halaman **Add Layout**yang pertama, tedapat form untuk mengisi nama layout, memilih template, serta menuliskan deskripsi yang ingin dituliskan.



Gambar 3.18 Halaman Edit Layout

Gambar diatas merupakan tampilan dari halaman **Edit Layout**. Pada halaman ini pengguna dapat mengisi media yang diinginkan dan juga bisa menambahkan widget

## Rancangan Pengujian

Penelitian ini menggunakna metode pengujian *blackbox testing*. *Blackbox testing* atau bisa disebut tes fungsional ini adalah pengujian yang dilakukan hanya dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari aplikasi yang sedang dikembangkan.

Pengujian program aplikasi ini dilakaukan oleh pengembang dan pengguna yang terlibat untuk memberi data yang akan dimasukan. Selain itu pengguna mencoba berbagai fitur pada aplikasi ini. Hal-hal yang menjadi keutamaan dalam pengujian adalah sebagai berikut :

1. Dapat melakukan unggah media.
2. Dapat membuat *layout*.
3. Dapat mengedit *layout*.
4. Dapat mempulish *layout*.
5. Dapat membuat penjadwalan *layout* ke *Display*.
6. Pengajuan terintegrasi dengan xibo.