**LAPORAN**

**PROJECT WORK**

**APLIKASI SISTEM PENGELOLAAN DATA PERPUSTAKAAN MENGGUNAKAN LARAVEL BERBASIS WEBSITE**

Diajukan untuk memenuhi salah satu syarat kelulusan dari SMK Negeri 1 Cimahi

Oleh:

NAMA : ILHAM SHIDDIQ

NO. INDUK SISWA : 181113836

TINGKAT : III (TIGA)

PAKET KEAHLIAN : REKAYASA PERANGKAT LUNAK

BIDANG KEAHLIAN : TEKNOLOGI INFORMASI DAN KOMUNIKASI



**SEKOLAH MENENGAH KEJURUAN NEGERI 1**

**CIMAHI**

**2021**

# **KATA PENGANTAR**

Puji dan syukur penulis panjatkan kehadirat Allah SWT. yang senantiasa melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan karya tulis ini dengan baik walaupun masih banyak kekurangan yang harus diperbaiki lagi. Karya tulis ini dibuat dengan tujuan sebagai salah satu syarat kelulusan dari SMK Negeri 1 Kota Cimahi, dan sebagai bentuk pertanggung jawaban atas *project work* yang telah penulis laksanakan selama kurang lebih 5 bulan. Karya tulis ini berkaitan dengan *project work* yang berjudul O’Library.

Penulis menyadari masih banyak batasan ilmu pengetahuan yang dimiliki dan kesalahan baik dalam penulisan atau perangkaian kata-kata, sehingga karya tulis ini jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan masukan agar karya tulis ini dapat menjadi lebih baik dari sebelumnya dan layak dikatakan sebagai karya tulis utuh yang sudah sesuai.

Proses pembuatan *project work* dan karya tulis ini dibuat tak lepas dari bantuan dan bimbingan oleh beberapa pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih sebesar-besarnya kepada:

1. Orang tua dan segenap keluarga yang telah memberikan dorongan do’a kepada penulis,
2. Kepala Sekolah SMK Negeri 1 Cimahi, yaitu Bapak Daud Saleh, M.M.,
3. Kepala Perpustakaan SMK Negeri 1 Cimahi,
4. Kepala Kompetensi Keahlian Rekayasa Perangkat Lunak SMK Negeri 1 Cimahi, Bapak Agus Rahmawan, ST.,
5. Kepala Bengkel Rekayasa Perangkat Lunak SMK Negeri 1 Cimahi, Bapak Agus Suratna, S.Pd.,
6. Ibu Siti Maryam, S.Kom., M.Kom., selaku pembimbing penulis yang selalu memberikan saran dan masukan selama pelaksanaan *project work*,
7. Teman-teman semua khususnya untuk XII RPL A yang selalu memberikan semangat kepada penulis.

Serta semua pihak lain yang telah turut memberikan dorongan semangat kepada penulis, yang tak mungkin penulis sebutkan satu-satu pada kesempatan kali ini. Semoga dengan adanya karya tulis ini, dapat menambah semangat pembaca untuk menggali ilmu lebih dalam lagi, terlebih di bidang Teknologi Informasi dan Komunikasi yang sekarang sedang berkembang pesat.

|  |  |  |
| --- | --- | --- |
|  |  | Cimahi, Januari 2021 |
|  |  |  |
|  |  | Penulis |

# **DAFTAR ISI**

[**KATA PENGANTAR** 2](#_Toc63164134)

[**DAFTAR ISI** 4](#_Toc63164135)

[**DAFTAR GAMBAR** 8](#_Toc63164136)

[**DAFTAR TABEL** 9](#_Toc63164137)

[**BAB I PENDAHULUAN** 10](#_Toc63164138)

[1.1 Latar Belakang Masalah 10](#_Toc63164139)

[1.2 Tujuan 11](#_Toc63164140)

[1.3 Pembatasan Masalah 11](#_Toc63164141)

[1.4 Sistematika Pembahasan 12](#_Toc63164142)

[**BAB II LANDASAN TEORI** 14](#_Toc63164143)

[2.1. Perpustakaan 14](#_Toc63164144)

[2.2. *Ebook* 14](#_Toc63164145)

[2.3. Data 15](#_Toc63164146)

[2.4. *Website* 15](#_Toc63164147)

[2.4.1 *Website* Statis 15](#_Toc63164148)

[2.4.2 *Website* Dinamis 16](#_Toc63164149)

[2.5. Web Aplikasi 16](#_Toc63164150)

[2.6. HTML (*Hypertext Markup Language*) 16](#_Toc63164151)

[2.7. *Blade* 17](#_Toc63164152)

[2.8. CSS (*Cascading Style Sheet*) 17](#_Toc63164153)

[2.9. SASS (*Syntactically Awesome Style Sheet*) 18](#_Toc63164154)

[2.10. Javascript 18](#_Toc63164155)

[2.11. Jquery 19](#_Toc63164156)

[2.12. AJAX (*Asyncronous* Javascript and XML) 19](#_Toc63164157)

[2.13. PHP : *Hypertext Preprocessor* 20](#_Toc63164158)

[2.14. *Framework* 20](#_Toc63164159)

[2.15. Konsep MVC 20](#_Toc63164160)

[2.16. Laravel 21](#_Toc63164161)

[2.17. Bootstrap 21](#_Toc63164162)

[2.18. Basis Data (*Database*) 22](#_Toc63164163)

[2.19. MySQL 22](#_Toc63164164)

[2.20. Webpack 23](#_Toc63164165)

[2.21. XAMPP 23](#_Toc63164166)

[2.22. DFD (*Data Flow Diagram*) 23](#_Toc63164167)

[2.23. ERD (*Entity Relationship Diagram*) 24](#_Toc63164168)

[2.24. *Use case Diagram* 25](#_Toc63164169)

[2.25. Flowmap 26](#_Toc63164170)

[**BAB III ANALISA DAN PERANCANGAN** 28](#_Toc63164171)

[3.1. Analisa Aplikasi 28](#_Toc63164172)

[3.1.1. Analisa Sistem Pengguna 28](#_Toc63164173)

[3.1.2. Analisa Kebutuhan Sistem 29](#_Toc63164174)

[3.2. Perancangan Sistem 30](#_Toc63164175)

[3.2.1. *Use Case Diagram* 30](#_Toc63164176)

[3.2.2. *Data Flow Diagram* (DFD) 30](#_Toc63164177)

[3.2.3. *Entity Relationship Diagram* (ERD) 38](#_Toc63164178)

[3.2.4. Flowmap 39](#_Toc63164179)

[3.3. Perancangan *Layout* 48](#_Toc63164180)

[3.3.1. Perancangan *Layout Landing Page* 48](#_Toc63164181)

[3.3.2. Perancangan *Layout* Halaman *Login* 49](#_Toc63164182)

[3.3.3. Perancangan *Layout* Halaman *Dashboard Admin*/Pustakawan 50](#_Toc63164183)

[3.3.4. Perancangan *Layout* Halaman Data Buku 50](#_Toc63164184)

[3.3.5. Perancangan *Layout* Halaman Data Anggota 51](#_Toc63164185)

[3.3.6. Perancangan *Layout* Halaman Data *Ebook* 51](#_Toc63164186)

[3.3.7. Perancangan *Layout* Halaman Data Penerbit 52](#_Toc63164187)

[3.3.8. Perancangan *Layout* Halaman Data Kategori 52](#_Toc63164188)

[3.3.9. Perancangan *Layout* Halaman Data Pengajuan 53](#_Toc63164189)

[3.3.10. Perancangan *Layout* Halaman Data Transaksi 53](#_Toc63164190)

[3.3.11. Perancangan *Layout* Halaman Data Laporan 54](#_Toc63164191)

[3.3.12. Perancangan *Layout* Halaman Data Pustakawan/*Admin* 54](#_Toc63164192)

[3.3.13. Perancangan *Layout* Halaman Cetak Laporan 55](#_Toc63164193)

[3.3.14. Perancangan *Layout* Halaman Cetak Kartu Anggota 55](#_Toc63164194)

[3.3.15. Perancangan *Layout* Halaman Konfigurasi 56](#_Toc63164195)

[3.3.16. Perancangan *Layout* Halaman *Dashboard* Anggota 56](#_Toc63164196)

[3.3.17. Perancangan *Layout* Halaman Data Buku Anggota 57](#_Toc63164197)

[3.3.18. Perancangan *Layout* Halaman Data *Ebook* Anggota 57](#_Toc63164198)

[3.3.19. Perancangan *Layout* Halaman Detail Buku/*Ebook* 58](#_Toc63164199)

[**BAB IV IMPLEMENTASI DAN PENGUJIAN** 59](#_Toc63164200)

[4.1. Implementasi Tampilan Aplikasi 59](#_Toc63164201)

[4.1.1. Implementasi Tampilan Halaman *Landing Page* 59](#_Toc63164202)

[4.1.2. Implementasi Tampilan Halaman *Login* 59](#_Toc63164203)

[4.1.3. Implementasi Tampilan Halaman *Dashboard* *Admin*/Pustakawan 59](#_Toc63164204)

[4.1.4. Implementasi Tampilan Halaman Data Buku 59](#_Toc63164205)

[4.1.5. Implementasi Tampilan Halaman Data Anggota 59](#_Toc63164206)

[4.1.6. Implementasi Tampilan Halaman Data *Ebook* 59](#_Toc63164207)

[4.1.7. Implementasi Tampilan Halaman Data Penerbit 59](#_Toc63164208)

[4.1.8. Implementasi Tampilan Halaman Data Kategori 59](#_Toc63164209)

[4.1.9. Implementasi Tampilan Halaman Data Pengajuan *Ebook* 59](#_Toc63164210)

[4.1.10. Implementasi Tampilan Halaman Data Transaksi 59](#_Toc63164211)

[4.1.11. Implementasi Tampilan Halaman Data Laporan 59](#_Toc63164212)

[4.1.12. Implementasi Tampilan Halaman Data Pustakawan 59](#_Toc63164213)

[4.1.13. Implementasi Tampilan Halaman Cetak Laporan 59](#_Toc63164214)

[4.1.14. Implementasi Tampilan Halaman Cetak Kartu Anggota 59](#_Toc63164215)

[4.1.15. Implementasi Tampilan Halaman Konfigurasi 59](#_Toc63164216)

[4.1.16. Implementasi Tampilan Halaman *Dashboard* Anggota 59](#_Toc63164217)

[4.1.17. Implementasi Tampilan Halaman Data Buku Anggota 59](#_Toc63164218)

[4.1.18. Implementasi Tampilan Halaman Data *Ebook* Anggota 59](#_Toc63164219)

[4.1.19. Implementasi Tampilan Halaman Detail Buku/*Ebook* 59](#_Toc63164220)

[4.2. Struktur *Database* 59](#_Toc63164221)

[4.3. Pengujian 59](#_Toc63164222)

[**BAB V PENUTUP** 60](#_Toc63164223)

[5.1. Kesimpulan 60](#_Toc63164224)

[5.2. Saran 60](#_Toc63164225)

[**DAFTAR PUSTAKA** 61](#_Toc63164226)

# **DAFTAR GAMBAR**

Gambar 2.1 Logo dari HTML

Gambar 2.2 Logo dari CSS

Gambar 2.3 Logo dari SASS

Gambar 2.4 Logo dari Javascript

Gambar 2.5 Logo dari PHP

Gambar 2.6 Logo dari Laravel

Gambar 2.7 Logo dari Bootstrap

Gambar 2.8 Logo dari XAMPP

# **DAFTAR TABEL**

Tabel

# **BAB I PENDAHULUAN**

## **Latar Belakang Masalah**

Perkembangan zaman saat ini, mendorong keberadaan teknologi menjadi hal yang memiliki arti penting bagi kehidupan manusia. Berbagai peralatan diciptakan dengan tujuan tertentu, memiliki kemudahan yang menjadi nilai tambah akan pentingnya teknologi. Dengan begitu, penggunaan teknologi bagi manusia menjadi suatu keharusan yang wajib hadir untuk membantu menyelesaikan berbagai masalah dan pekerjaan.

*Internet* merupakan salah satu contoh dari teknologi masa kini yang keberadaannya sudah sangat melekat dengan manusia. Penggunaan *internet* sudah tersebar di seluruh dunia, setiap orang dapat mengaksesnya dengan bebas. *Internet* memudahkan proses komunikasi dan pembagian informasi secara cepat, baik itu antar daerah, kota, negara bahkan benua. Namun, di balik kemudahan yang begitu banyaknya, dalam *internet* itu sendiri tersimpan banyak sisi negatif yang memiliki pengaruh dan dampak besar pada manusia.

Pengambilan sisi positif dari teknologi termasuk *internet*, merupakan hal penting yang harus dilakukan manusia saat ini. Hal ini dilakukan agar manusia tidak terjebak di dunia yang tak nyata yang sewaktu-waktu dapat memperbudak dirinya. Membuat teknologi yang tepat guna merupakan salah satu cara yang digunakan untuk mengambil sisi positif dari *internet* dan teknologi. Agar nantinya, teknologi tersebut dapat digunakan sesuai dengan kegunaanya dan tidak disalah gunakan.

Keberadaan aplikasi contoh dari penggunaan teknologi tepat guna zaman sekarang. Penggunaan aplikasi saat ini sudah menjadi hal lumrah yang ada di masyarakat, berbagai aplikasi hadir dengan kegunannya masing-masing. Tak terlepas dari dunia sekolah, banyak aplikasi yang dikemas dengan tujuan untuk memudahkan siswa selama proses belajar. Ada pula aplikasi yang dibuat dengan tujuan untuk memudahkan guru dalam mendata nilai, meyusun rapot hingga pendataan perpustakaan.

Aplikasi O’Library contohnya, dibuat dengan menggunakan basis *website* agar semakin mudah dijangkau dan dipakai oleh semua pengguna terutama siswa. Dibuat dengan tujuan utama untuk memudahkan pendataan buku yang tersedia di perpustakaan sekolah, memiliki penyimpanan digital agar semakin mudah diakses.

## **Tujuan**

Tujuan yang dicapai dalam pembuatan web aplikasi O’Libraryadalah :

1. Membuat kemudahan dalam mengelola data bagi para pustakawan dan *admin.*
2. Memudahkan pencarian buku bagi para anggota.
3. Membuat fitur yang mudah dimengerti.
4. Tampilan yang sederhana agar mudah digunakan oleh pustakawan, *admin* mapun anggota.
5. Memudahkan untuk mencari anggota yang belum mengembalikan buku yang sudah jatuh tempo.
6. Dapat menggunakan *ebook* yang bisa digunakan dimanapun melalui persetujuan pustakawan.
7. Laporan yang dikeluarkan berbentuk *file* digital, yaitu excel dan PDF*(Portable Document Format)*.

## **Pembatasan Masalah**

Masalah yang disampaikan pada latar belakang perlu dibatasi agar sesuai dengan apa yang dibahas pada karya tulis ini. Berikut batasan - batasan masalahnya :

1. Aplikasi ini berbentuk *website*.
2. *Web* aplikasi ini digunakan sebagai pembukuan data perpustakaan yang dilakukan secara digital dan lebih terorganisir.
3. Penginputan data masih dilakukan secara manual.
4. Belum menggunakan fitur *scan bar code* untuk pendataan yang lebih mudah dan cepat.

## **Sistematika Pembahasan**

Penyusunan laporan *project work* ini terbagi ke dalam beberapa bab dengan tujuan agar pembaca dapat lebih mudah untuk membaca dan memahami isi dari laporan ini.

Bab I Pendahuluan memuat penjelasan dari latar belakang masalah pembuatan aplikasi, dijabarkan daftar tujuan dari pembuatan aplikasi, pembatasan masalah yang isinya berkenaan dengan ruang lingkup pembahasan di laporan ini, dan penjabaran singkat sistematika pembahasan yang dibagi menjadi beberapa bab.

Bab II Landasan Teori membahas singkat materi-materi yang menunjang serta menjadi acuan penulis dalam menyusun laporan ini. Digunakan pula sebagai sumber informasi bagi pembaca bila menemukan beberapa istilah baru yang belum dipahami.

Bab III Analisa dan Perancangan membahas mengenai hasil analisis terhadap web aplikasi yang dibuat serta penjabaran rancangan pembuatan web aplikasi. Analisa yang ada, dilakukan pada pengguna yang terlibat serta penjelasan haknya dalam mengakses *web* aplikasi. Juga dilakukan analisa kebutuhan sistem mengenai spesifikasi perangkat yang dipakai selama pengembangan *web* aplikasi. Terkait perancangan, dijelaskan mengenai struktur *database* yang dipakai serta perencanaan *layout/mockup* yang telah dibuat sebelum proses pembuatan *web* aplikasi dilakukan.

Bab IV Implementasi dan Pengujian membahas terkait hasil proses yang dilakukan selama pembuatan *web* aplikasi, dijelaskan mengenai implementasi pembuatan struktur dalam *database*, implementasi rancangan *layout* menjadi sebuah *web* yang utuh, implementasi sistem yang berisi pemasangan *software* yang menjadi kebutuhan wajib saat proses pengembangan serta diberikan tabel pengujian terhadap fitur-fitur yang ada pada *web* aplikasi.

Bab V Penutup berisi simpulan penulis terkait pembuatan aplikasi yang dimulai dari perancangan hingga implemetasi. Juga diberikan saran-saran untuk *web* aplikasi yang diharapkan dapat diimplementasikan agar nantinya memiliki kualitas yang lebih baik dari sekarang.

# **BAB II LANDASAN TEORI**

## **Perpustakaan**

Perpustakaan merupakan kata tak asing yang terdengar di telinga kita, karena kehadirannya yang hampir ada dimana-mana. Tak terkecuali di lingkungan sekolah, kini salah satu hal yang wajib ada di sekolah yaitu adanya perpustakaan. Dalam arti tradisioal, perpustakaan adalah suatu tempat/bangunan yang memiliki banyak koleksi buku ataupun majalah. Secara umum, berdirinya suatu perpustakaan yang memiliki koleksi besar, dibiayai dan dioperasikan oleh sebuah kota atau institusi, serta dimanfaatkan oleh masyarakat sebagai tempat membaca dan menggali ilmu pengetahuan.

Namun, zaman modern sekarang, sebuah perpustakaan tidak hanya memiliki koleksi yang terdiri atas buku, majalah ataupun media baca cetak lainnya, tak jarang banyak yang memiliki koleksi buku dalam bentuk digital, atau lebih sering disebut dengan *ebook*. Selain itu, saat ini umumpunya setiap perpustakaan memiliki fasilitas *internet* yang dapat diakses bebas oleh masyarakat yang berkunjung untuk membaca.

## ***Ebook***

Media baca memiliki banyak sekali jenisnya, dari media cetak sampai digital. Mengikuti perkembangan zaman, media baca dapat dijumpai dalam bentuk digital secara mudah tanpa harus membawa buku secara langsung, yang hanya dibutuhkan hanyalah *gadget*/ponsel pintar yang sekarang telah banyak orang pakai.

*Ebook* adalah buku elektronik yang dibuat secara digital yang umumnya dipakai di *gadget*. Banyak orang lebih memilih untuk membaca buku digital dibanding buku kertas karena kemudahan yang diperoleh. Selain kemudahan, sumber ilmu yang ada di *ebook* tidak akan berbeda dengan buku secara fisik, sehingga dapat dipastikan bahwa dengan *ebook*, informasi serta ilmu pengetahuan yang tercetak akan sama saja tanpa adanya pengurangan.

## **Data**

Data merupakan kumpulan fakta yang dijadikan menjadi satu. Dalam keilmuan, fakta dikumpulkan untuk menjadi data. Data kemudian diolah sehingga dapat diutarakan secara jelas dan tepat sehingga dapat dimengerti oleh orang lain. Dalam suatu kumpulan data, tentu beragam dan banyak memiliki jenis, sehingga pemilahan data perlu dilakukan, hal ini biasa disebut dengan klasifikasi data.

Dalam ilmu teknologi, tak terlepas pula dengan adanya suatu data. Data yang digunakan merupakan hal penting yang hadir untuk dapat menampilkan semua informasi yang dibutuhkan. Bentuk pengolahan data dalam teknologi biasa disebut dengan basis data.

## ***Website***

Situs *web* atau *website* merupakan sekumpulan halaman *web* yang saling berhubungan yang umumnya dibuat oleh perorangan, kelompok atau suatu badan resmi. Sebuah situs *web* ditempatkan setidaknya pada satu *web server* yang dapat diakses melalui jaringan *internet*. Dengan adanya *website*, informasi yang terkandung akan mudah diakses oleh orang-orang diseluruh dunia.

*Website* memiliki 2 jenis yang memiliki karakteristik yang berbeda. Berikut penjelasan dari macam-macam *website* yang umum ditemukan.

## ***Website* Statis**

*Website* statis adalah *website* yang dibuat secara manual dengan mengetikkan kode-kode *Hypertext Markup Language* (HTML) yang isinya tidak dimaksudkan untuk adanya perubahan konten secara berkala. *Website* statis dibuat hanya untuk menampilkan informasi yang tetap sehingga tidak ada interaksi yang terjadi antara *web* dengan pengguna. *Website* statis tidak terhubung dengan *database* atau basis data, sehingga untuk pengubahan data, hanya dilakukan dengan mengubah skripnya langsung. Contoh dari *website* statis adalah *website* portfolio, yang isinya merupakan data tetap yang minim perubahan. Selain itu, *website company profile* juga termasuk ke dalam penggunaan *website* statis.

## ***Website* Dinamis**

*Website* dinamis merupakan *website* yang secara spesifik di desain agar konten di dalamnya dapat diubah secara berkala dengan mudah, yang umumnya perubahan konten ini dilakukan dalam satu periode tertentu. Berbeda dengan *website* statis, *website* dinamis merupakan *web* yang penuh interaksi dengan pengguna, interaksi yang terjadi seperti memberikan komentar, *member* *rating*, mengisi *form* data, dan membuat pesanan. Hal-hal tersebut membutuhkan data yang *up-to-date* atau data terbaru, sehingga *website* dinamis dipakai untuk mengatasi interaksi tersebut.

Penggunaan *database* atau basis data pada website dinamis merupakan hal yang penting, karena untuk mengatasi perubahan data, dilakukan oleh sistem yang terhubung dengan *database*. Berbagai contoh dari *website* dinamis adalah *website* berita dan *website e-commerce*.

## ***Web* Aplikasi**

*Web* aplikasi adalah teknologi yang merupakan gabungan dari *web* dan aplikasi. Sesuai anamanya, *web* aplikasi yaitu aplikasi yang akses penggunaannya dilakukan di penjelajah *website* dan dikases dengan jaringan *internet*. *Web* aplikasi bersifat dinamis dan akan terus berkembang.

*Web* aplikasi dibuat dengan menggunakan kode bahasa yang dibuat untuk pengembangan web, seperti ASP, Perl, Javascript, PHP, Python dan Ruby serta didukung dengan bahasa khusus perancangan *web* seperti HTML dan CSS. Sementara itu, untuk kebutuhan yang lebih kompleks dan rumit, *web* aplikasi dapat dibuat dengan menggunakan *framework*, seperti Django, Codeigniter dan Laravel.

## **HTML (*Hypertext Markup Language*)**

HTML adalah sebuah bahasa markah (*markup*) yang digunakan untuk membuat sebuah halaman *web* yang berisikan informasi. HTML bermula dari sebuah bahasa yang digunakan di dunia penerbitan dan percetakan yang disebut SGML (*Standard Generalized Markup Language*). HTML sangat lumrah digunakan para *programmer* untuk membuat sebuah *website* statis maupun dinamis dibantu oleh CSS, JS, *database* sehingga tampilan menjadi lebih bagus.

HTML berasal dari kombinasi antara *text* dan *symbol*/kode yang dimasukkan dalam sebuah *file*, HTML bukanlah sebuah bahasa pemrograman sehingga HTML tidak punya kemampuan fungsionalitas yang dinamis. Dalam pengembangan aplikasi O’LibraryHTML merupakan bahasa yang menjadi kerangka dasar aplikasi.



Gambar 2.1 Logo dari HTML

## ***Blade***

*Blade* merupakan fitur yang disediakan *framework* Laravel yang berfungsi untuk membantu menyusun *view*/tampilan *web*. *Blade* adalah teknik *templating* sederhana, tetapi akan sangat membantu *developer* karena *blade* sedikit berbeda dengan HTML, di dalam *file* blade, *developer* tidak perlu mengetikkan sintaks dasar PHP di dalamnya, dan diganti dengan sintaks yang lebih mudah.

Selain itu, dengan blade pula terdapat fitur *templating*. Di suatu *website*, tentu terdapat beberapa halaman *web* yang memiliki tampilan yang sama, yang membedakan hanya kontennya saja. Sehingga dibutuhkan cara untuk mempersingkat waktu agar tidak semua tampilan memiliki filenya sendiri yang isinya sama persis dengan *file* lain, *blade* menyediakan cara untuk menggunakan fungsi *extend*, untuk dapat mengambil tampilan dari *file* lain.

## **CSS (*Cascading Style Sheet*)**

CSS merupakan kumpulan perintah yang digunakan untuk mengatur *design* dari proses pembuatan *website*, karena hal ini tidak bisa dilakukan oleh HTML. Penggunaan CSS akan membuat tampilan dari sebuah *web* akan lebih terstruktur dan seragam, karena di CSS, pengaturan warna, bentuk, posisi hingga ukuran elemen HTML dilakukan disini. Karena CSS terlihat dalam tampilan depan sehingga biasa disebut *Front* *End*.



Gambar 2.2 Logo dari CSS

## **SASS (*Syntactically Awesome Style Sheet*)**

SASS adalah bahasa pre-prosesor untuk CSS. SASS dibuat karena dalam CSS, konsep pemrograman seperti *variable*, *function*, dan *condition* tidak dapat digunakan, sehingga dengan SASS, konsep-konsep tersebut dapat dipakai dan penulisan CSS akan lebih singkat dan terstruktur. SASS bukanlah bahasa pemrograman, melainkan bahasa yang digunakan untuk membuat CSS.



Gambar 2.3 Logo dari SASS

## **Javascript**

JavaScript adalah bahasa pemrograman tingkat tinggi yang bersifat dinamis, cara kerja Javascript hanya memerlukan *interpreter* sehingga tidak membutuhkan *compiler*. Hampir semua *website* menggunakan Javascript, karena dengan menggunakan Javascript sebuah *website* menjadi lebih menarik dan pengguna dapat berinteraksi dengan *website*.

Javascript terkenal sintaksnya yang mudah dipahami dan terus mengalami pengembangan. Contohnya, pada tahun 2015, Javascript mengeluarkan *versi* yang bernama Javascript Ecmascript 6 (JS ES 6), dengan itu, banyak sintaksnya yang dibuat lebih singkat dan penambahan fungsionalitas, sehinggal *developer* akan lebih mudah dalam proses pengembangan *website*.



Gambar 2.4 Logo dari Javascript

## **Jquery**

Jquery adalah *library* Javascript yang paling populer, yang pertama kali dibuat pada tahun 2006. Jquery bukanlah bahasa yang berdiri sendiri, melainkan bekerja sama dengan Javascript. Jquery dibuat dengan tujuan untuk mempercepat *developer* dalam membuat *website*. Dengan Jquery, *developer* pemula dapat belajar lebih mudah dibanding langsung belajar untuk memahami Javascript. Dengan jquery, banyak hal yang dapat dilakukan dan dipakai untuk *website*, sehingga akan lebih interaktif.

## **AJAX (*Asyncronous Javascript and XML*)**

AJAX merupakan teknologi javascript modern yang kerap digunakan oleh banyak *website* dinamis. AJAX digunakan untuk proses pengiriman *request*, dan respon akan didapat tanpa adanya proses *refresh*/pembaruan pada halaman *web*. Jika AJAX dihubungkan dengan sistem *backend* seperti PHP, maka AJAX berperan sebagai perantara antara *user* dengan *database*, AJAX akan mengirimkan *request* ke *database* sesuai permintaan *user*, lalu *database* akan mengirimkan respon atau hasil data *request* yang kemudian dikirimkan kepada *user* tanpa adanya proses *refresh page* pada *website*.

## **PHP : *Hypertext Preprocessor***

PHP Adalah bahasa pemrograman *server-side* *scripting* yang digunakan untuk mengembangkan *web* dinamis. Kini, PHP menjadi salah satu bahasa paling populer digunakan dalam pembuatan *website* dinamis, karena sintaksnya yang mudah dimengerti serta dapat dikombinasikan dengan bahasa lain seperti Javascript. PHP dapat disisipkan dengan mudah pada *file* HTML, sehingga proses penampilan data akan mudah.



Gambar 2.5 Logo dari PHP

## ***Framework***

*Framework* adalah sebuah kerangka *program* yang isinya adalah berbagai fungsi, *plugin*, dan konsep yang membentuk suatu sistem. *Framework* digunakan untuk membantu (memfasilitasi) jenis pemrograman tertentu untuk proyek pengembangan *software* atau *website*, tujuan *framework* ialah untuk memudahkan/membantu *developer* dalam membuat suatu *website* dengan waktu yang singkat, tersusun dan terstruktur.

Berbagai *framework* sekarang hadir dengan mendukung beberapa bahasa pemrograman populer, seperti Django untuk python, Vue untuk Javascipt dan Laravel untuk PHP.

## **Konsep MVC**

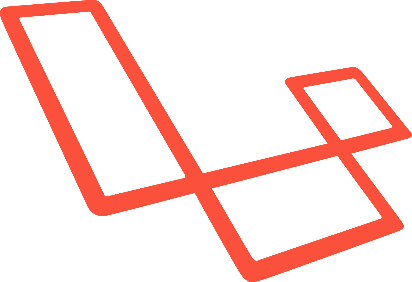
MVC (*Model* *View* *Controller*) merupakan suatu konsep yang digunakan dengan cara memisahkan data, tampilan dan pemrosesan kedua hal tersebut. Semua hal yang berurusan dengan data dari *database*, akan ditangani oleh *Model*, ini merupakan bagian terpenting karena semua data yang tampil bersumber dari sini. Untuk bagian yang mengurus tampilan atau *frontend* dari konsep MVC yaitu *View*. Dan yang terakhir, bagian yang mengatur lalu lintas antara *model* dan *view* yaitu *controller*, bagian ini mengelola data yang di dapat dari *model*, kemudian ditampilkan ke *view*.

Konsep MVC sudah digunakan oleh banyak *framework* terkenal, seperti Laravel dan Django. Walaupun konsep ini cukup sulit untuk dipelajari bagi pemula, namun banyak *developer* yang merasa terbantu dengan konsep MVC. Keunggulan dari konsep MVC yaitu struktur *file* yang lebih terstruktur.

## **Laravel**

Laravel adalah *framework* PHP yang dirilis dibawah lisensi MIT yang paling populer dan banyak digunakan dalam proses pengembangan *website* yang *powerful*, cepat, aman dan *simple*. Laravel diluncurkan pada 2011 dan mengalami pertumbuhan yang cukup signifikan, hingga kini Laravel salah satu *framework* yang sangat populer di dunia.

Hal yang menyebabkan Laravel populer yaitu banyak didukung oleh berbagai *package* tambahan yang dapat dipakai untuk aplikasi, seperti *package* yang dapat mengubah suatu data menjadi *file* *excel* ataupun pdf. Selain itu, Laravel dibangun menggunakan konsep MVC, sehingga struktur *file* yang dipakai lebih terstruktur.



Gambar 2.6 Logo dari Laravel

## **Bootstrap**

Bootstrap adalah kerangka kerja CSS yang telah dibuat khusus untuk mengembangkan *website* dengan desain *responsive*. Kerangka kerja ini berisi *template* desain berbasis HTML dan CSS, hal ini menyebabkan hampir semua *web* *developer* menggunakan bootsrap agar tampilan lebih baik dan *responsive*.

Konsep yang digunakan dalam Bootstrap yaitu konsep *class*, berbagai *class* sudah disediakan oleh Bootstrap dan *developer* dapat langsung menggunakannya sesuai kebutuhan. Kini, Bootstrap merupakan framework CSS paling populer dan banyak digunakan dibanding *framework* CSS lain seperti Materialize.



Gambar 2.7 Logo dari Bootstrap

## **Basis Data (*Database*)**

Basis data dapat didefinisikan atau diartikan sebagai kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (*software*) *program* atau aplikasi untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang kemudian disimpan.

Basis data merupakan aspek yang sangat penting dalam sistem informasi karena berfungsi sebagai gudang penyimpanan data untuk diolah lebih lanjut. Basis data menjadi penting karena dapat mengorganisasi data, menghidari duplikasi data, menghindari hubungan antar data yang tidak jelas dan juga *update* yang rumit.

## **MySQL**

MYSQL adalah sebuah manajemen *database* yang menggunakan instruksi – instruksi khusus dalam menjalankan perintah sesuatu untuk memanipulasi data, MYSQL yang gratis ini telah digunakan sebanyak 6 juta instalasi di seluruh dunia dibawah lisensi GNU *General* *Public* *License* (GPL). SQL(*Structured* *Query* *Language*) sendiri sudah diakui ANSI secara *de facto* bahwa SQL mempunyai standar yang khusus.

Hal-hal yang dapat dilakukan dengan *query* SQL meliputi *query* sederhana seperti *create*, *read*, *update* dan *delete* data, hingga *query* rumit seperti *subquery* dan penggabungan 2 atau lebih tabel yang biasa disebut *inner* *join.*

## **Webpack**

Webpack merupakan sebuah module bundler JavaScript untuk membungkus module/dependency/library/assets menjadi satu paket *file* berekstensi JavaScript (.js). selain sebagai *module* *bundler*, dengan berbagai *loader* dan *plugin* yang disediakan untuk webpack, *developer* dapat melakukan beragam hal, seperti mengubah kode Javascript ES6 menjadi kode Javascript ES5, duplikasi *file* dengan mudah, menggunakan *loader* SASS dan SCSS dan *cache* *generator* untuk penggunaan *service-worker*. Dengan menggunakan webpack, file-file yang dibutuhkan akan dijadikan satu, sehingga akan memiliki ukuran yang relatif ringan.

## **XAMPP**

XAMPP adalah *software*/perangkat lunak yang mendukung berbagai sistem operasi, merupakan kompilasi dari beberapa *program*. Fungsinya yaitu sebagai *server* yang berdiri sendiri yang terdiri atas *program* Apache HTTP *Server*, MySQL *Database* dan Perl. Dalam proses pengembangan *website*, XAMPP berperan sebagai *web* *server* yang menangani MySQL *Database*.



Gambar 2.8 Logo dari XAMPP

## **DFD (*Data Flow Diagram*)**

Diagram Alir Data (DAD)/*Data Flow Diagram* (DFD) Adalah sebuah metode yang menggunakan notasi – notasi berorientasi pada sebuah aliran data. DFD menjelaskan analisis kerja suatu sistem yang berguna untuk mempermudah dalam merancang sebuah aplikasi dari *start* sampai *finish*.

DFD memiliki 4 simbol utama yang memiliki fungsinya tersendiri. Simbol DFD dibagi menjadi 4, terdiri atas entitas, proses, aliran data dan *data* *store*/penyimpanan data. Biasanya dalam pembuatan DFD dibuat dalam beberapa level, dimulai dengan diagram konteks, dilanjut hingga proses yang dijabarkan sudah detail dan rinci.

Berikut merupakan simbol-simbol yang terdapat dalam DFD menurut aturan De Marco yang penulis pakai dalam karya tulis ini.

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
| 1. Simbol Entitas | Simbol entitas digambarkan dengan bentuk persegi panjang, menggambarkan entitas/orang/*unit* yang terlibat dalah proses. |
| 1. Simbol Proses | Simbol proses digambarkan dengan bentuk lingkaran, menggabarkan proses yang terjadi dalam sistem, yang terhubung dengan entitas. |
| 1. Aliran Data | Aliran data disimbolkan dengan bentuk anak panah, menghubungkan antara entitas dan proses, menggambarkan arah suatu proses terjadi. |
| 1. *Data* *Store*  |  | | --- | | *Database* | | *Data* *store*/data penyimpanan digambarkan dengan bentuk 2 garis dengan nama *database* di dalamnya. |

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

ERD (*Entity Relationship Diagram*) adalah suatu bentuk *diagram* yang digunakan merancang/menyusun sebuah *database* untuk mengetahui hubungan relasi antar entitas atau objek beserta atributnya. ERD terdiri dari entitas dan atribut yang saling berhubungan yang digambarkan melalui garis yang ditarik dari entitas ke atribut atau dari entitas ke entitas. Entitas itu sendiri digambarkan dengan bentuk persegi panjang, sedangkan untuk atribut digambarkan dengan bentuk *oval*.

Berikut merupakan simbol-simbol ERD yang dipakai dalam laporan ini.

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
| 1. Entitas | Simbol entitas digambarkan dengan bentuk persegi panjang, merupakan objek yang terlibat dalam aplikasi. |
| 1. Atribut | Atribut menggambarkan sifat dari suatu entitas, terdiri atas komponen-komponen yang ada dalam suatu entitas. |
| 1. Relasi | Relasi disimbolkan dengan bentuk belah ketupat, menghubungkan satu entitas dengan entitas lain. |
| 1. Garis Penghubung | Garis penghubung dalam ERD terdapat 2 jenis, garis tanpa anak panah menggambarkan hubungan kardinalitas *Many* (N) sedangkan yang satu lagi menggambarkan *One* (1). |

## ***Use Case Diagram***

*Use case diagram* adalah pemodelan untuk menggambarkan *behaviour*/kelakuan sistem yang akan dibuat. *Use case* menggambarkan sebuah interaksi antara satu atau lebih aktor (*user*) dengan sistem proses yang dibuat. Sederhananya*, use case* *diagram* adalah *diagram* yang menunjukkan setiap proses yang terjadi dalam aplikasi dan siapa saja yang terlibat dalam proses tersebut.

Berikut merupakan simbol-simbol dari *use* *case* diagram yang dipakai dalam laporan ini.

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
| 1. *Use Case* | Digambarkan dengan simbol *oval*, merupakan fungsionalitas yang diberikan oleh sistem yang berhubungan dengan aktor. |
| 1. Aktor | Aktor/*user* yang terlibat dalam aplikasi, yang berhubungan dengan *use case* yang tersedia. |
| 1. Garis Penghubung | Garis yang menghubungkan antara aktor dan *use case*. |
| 1. *Extend*   <<*extend*>> | Garis yang menghubungkan *use case* dengan *use case*, berarti bahwa *use case* yang dituju tetap dapat berjalan walau tanpa disertai *use case* tersebut. |
| 1. *Include*   <<*include*>> | Garis yang menghubungkan *use case* dengan *use case*, berarti bahwa *use case* yang dituju tidak dapat berjalan tanpa disertai *use case* tersebut. |

## **Flowmap**

Flowmap adalah sebuah diagram yang menggambarkan proses yang terjadi di setiap aktor/user. Flowmap merupakan penjelasan lebih detail dari use case diagram, memetakan secara rinci apa saja proses yang berlangsung sampai akhir. Bentuk flowmap mirip dengan flowchart, namun tidak seperti flowchart yang hanya fokus pada setiap proses, flowmap memperhatikan aktor/user yang terlibat pula. Sehingga, penggunaan flowmap akan memudahkan programmer untuk menganalisis alur aplikasi, dan merevisi dengan mudah jika ada alur yang kurang sesuai, hal ini dapat dilakukan karena flowmap dibuat secara tersusun.

Berikut merupakan simbol-simbol flowmap yang dipakai dalam laporan ini.

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | 1. Proses, menunjukkan proses yang diolah/dilakukan oleh komputer. |
|  | 1. Kondisi, menunjukkan pemilihan sesuai kondisi yang diberikan, menghasilkan hasil berupa Yes atau No. |
|  | 1. Dokumen, menunjukkan dokumen data input/output, dapat berupa formulir atau kertas. |
|  | 1. Data, menunjukkan input/output yang akan diproses oleh komputer langsung. |
|  | 1. Manual Input, menunjukkan proses input yang datanya diinputkan lewat keyboard komputer. |
|  | 1. Data Store, menunjukkan penyimpanan database |
|  | 1. Terminator, menunjukkan awal/akhir dari suatu flowmap. |
|  | 1. Garis Panah, menunjukkan arah arus dari suatu flowmap. |

# **BAB III ANALISA DAN PERANCANGAN**

## **Analisa Aplikasi**

O’Library merupakan web aplikasi perpustakaan yang memiliki 2 fungsi utama, yaitu sebagai alat manajemen data perpustakaan serta media untuk melihat beberapa data yang tersedia. Web aplikasi ini dibuat dengan menggunakan framework PHP laravel untuk sisi backend agar memudahkan developer untuk mengembangkan aplikasi serta framework Bootstrap untuk sisi frontend.

## **Analisa Sistem Pengguna**

Berikut adalah tabel analisa kebutuhan pengguna, meliputi daftar pihak berupa level pengguna yang terlibat dalam aplikasi. Adapun pihak dan hak aksesnya adalah sebagai berikut.

Tabel 3.1 Analisa Sistem Penguna

|  |  |
| --- | --- |
| **Level** | **Hak Akses** |
| Admin | 1. Mengakses halaman dashboard yang memuat beberapa informasi penting perpustakaan 2. Melakukan pengolahan data staff pustakawan 3. Melihat laporan pengembalian buku 4. Mencetak laporan baik dalam bentuk file pdf maupun excel 5. Melakukan konfigurasi |
| Pustakawan | 1. Mengakses halaman dashboard yang memuat beberapa informasi penting perpustakaan 2. Melakukan pengolahan data buku perpustakaan 3. Melakukan pengolahan data anggota perpustakaan 4. Melakukan pengolahan data ebook perpustakaan 5. Melakukan pengolahan data penerbit untuk buku dan ebook 6. Melakukan pengolahan data kategori untuk buku dan ebook 7. Melakukan tindakan mengenai permintaan penggunaan ebook 8. Melakukan pengolahan data peminjaman atau pengembalian buku 9. Mengirim peringatan dalam bentuk email kepada anggota jika batas peminjaman telah melebihi batas 10. Mencetak kartu anggota yang terdaftar 11. Melihat laporan pengembalian buku 12. Mencetak laporan baik dalam bentuk file pdf maupun excel 13. Melakukan konfigurasi |
| Member | 1. Melihat serta mencari daftar buku yang tersedia di perpustakaan 2. Melihat serta mencari daftar ebook yang tersedia di perpustakaan 3. Mengirimkan permintaan penggunaan ebook kepada pustakawan 4. Menggunakan ebook jika permintaan disetujui |

## **Analisa Kebutuhan Sistem**

Dalam pembuatan aplikasi, dibutuhkan sistem yang digunakan developer untuk memudahkan dalam pengembangan. Berikut spesifikasi perangkat keras dan perangkat lunak yang digunakan oleh developer.

1. Perangkat Keras

Berikut spesifikasi perangkat keras yang digunakan dalam pengembangan web aplikasi.

1. Laptop
2. Processor : Intel® Core™ i3-2330M CPU @ 2.20GHz
3. Memori : 4,00 GB
4. Harddisk : 500 GB
5. Perangkat Lunak

Berikut beberapa software atau perangkat lunak yang dipakai selama proses pengembangan web aplikasi.

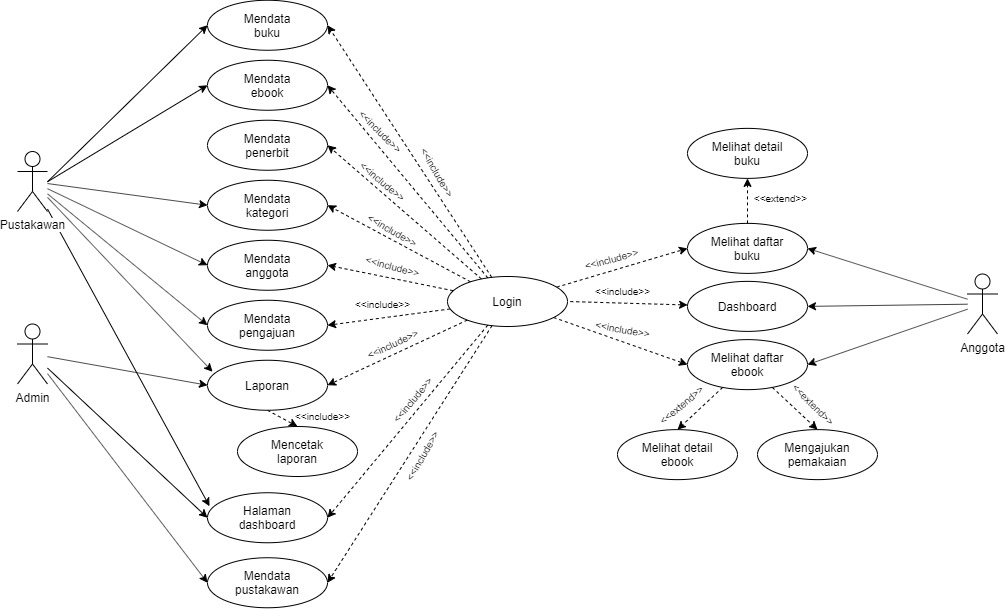
1. Sistem Operasi : Windows 8.1 Pro
2. Browser : Microsoft Edge versi 87.0.664.75, Google Chrome versi 87.0.4280.141 dan Mozilla Firefox versi 30.0
3. Teks Editor : Visual Studio Code (VSC)
4. Desain Grafis : Adobe Photoshop CS6 dan Adobe Illustrator CC
5. Paket Aplikasi Server : XAMPP

## **Perancangan Sistem**

Berikut merupakan analisis dari aplikasi yang dibuat dengan menggunakan diagram-diagram, meliputi use case diagram, data flow diagram (DFD), entity relationship diagram (ERD) dan flowmap.

## **Use Case Diagram**

Use case merupakan diagram yang menggambarkan sekelompok use cases dan aktor disertai dengan hubungan diantaranya. Berikut merupakan use case diagram yang merepresentasikan hubungan tiap user dan proses yang terjadi pada aplikasi O’Library.

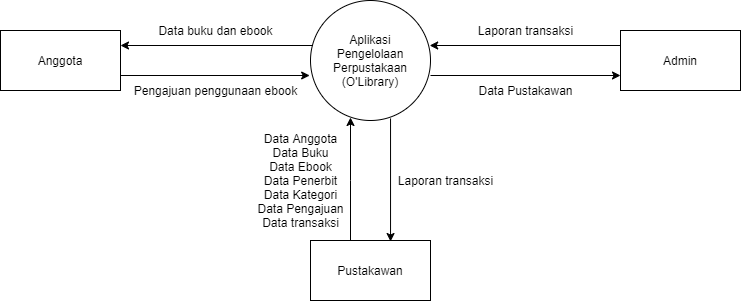


## **Data Flow Diagram (DFD)**

Data flow diagram (DFD) merupakan diagram yang menjelaskan tiap proses yang terjadi di dalam aplikasi serta semua entitas yang terlibat. Berikut DFD dari web aplikasi O’Library yang terbagi menjadi diagram kontes/context diagram, DFD level 1 dan DFD level 2.

* + - 1. Context Diagram

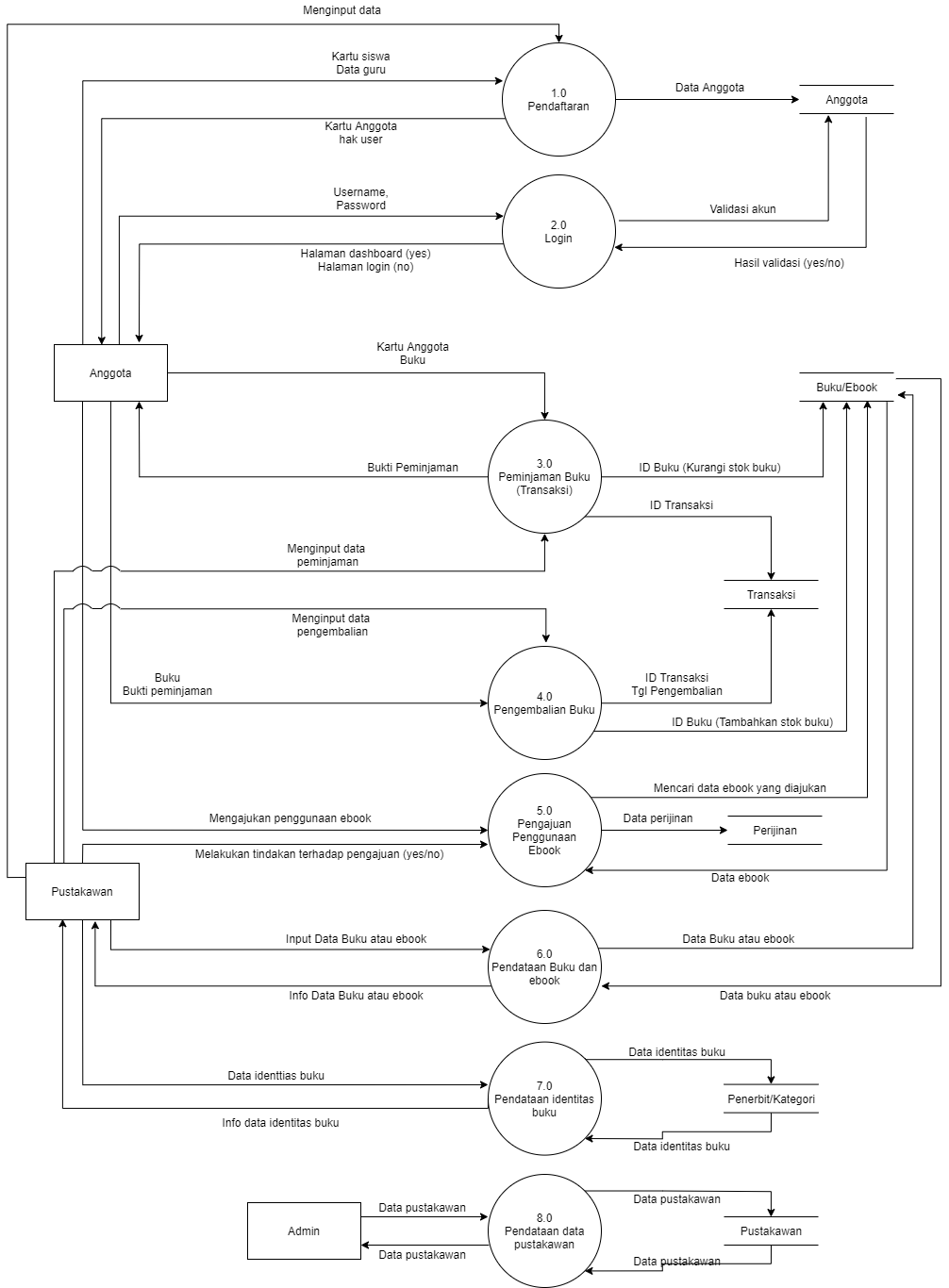
Berikut diagram konteks dari aplikasi yang menjelaskan bahwa terdapat 3 entitas utama yang terlibat dalam sistem aplikasi.



Gambar context diagram/diagram konteks

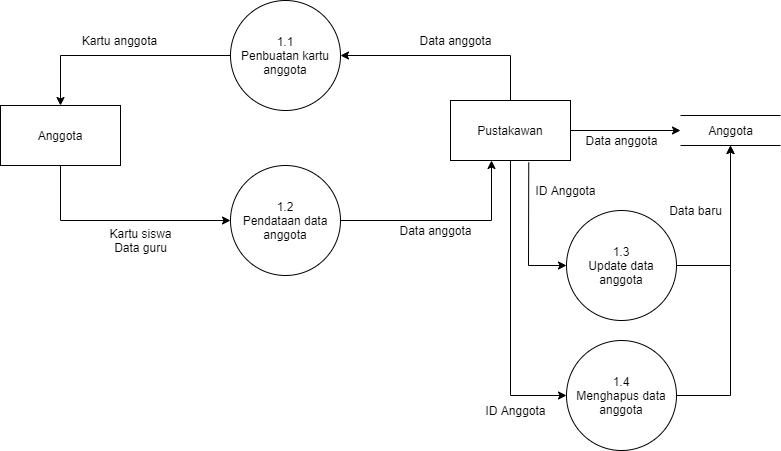
* + - 1. DFD Level 1

Berikut merupakan gambar DFD level 1 yang merupakan pengembangan dari diagram konteks. Dijabarkan bahwa 3 entitas utama terlibat ke dalam 8 proses yang terdapat dalam aplikasi.



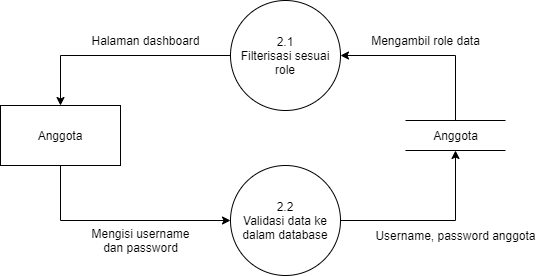
Gambar DFD Level 1

* + - 1. DFD Level 2
         1. DFD Level 2, Proses 1.0 Pendaftaran



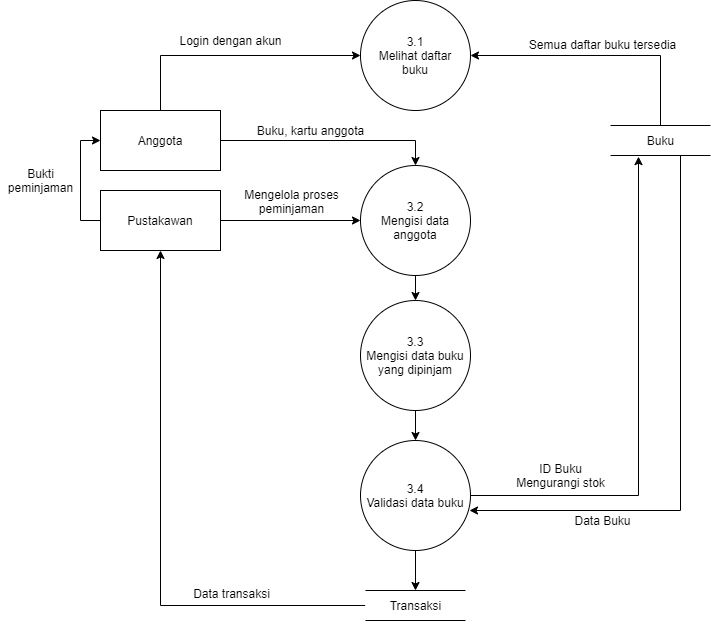
Gambar DFD Level 2 Proses 1.0 Pendaftaran

* + - * 1. DFD Level 2, Proses 2.0 Login



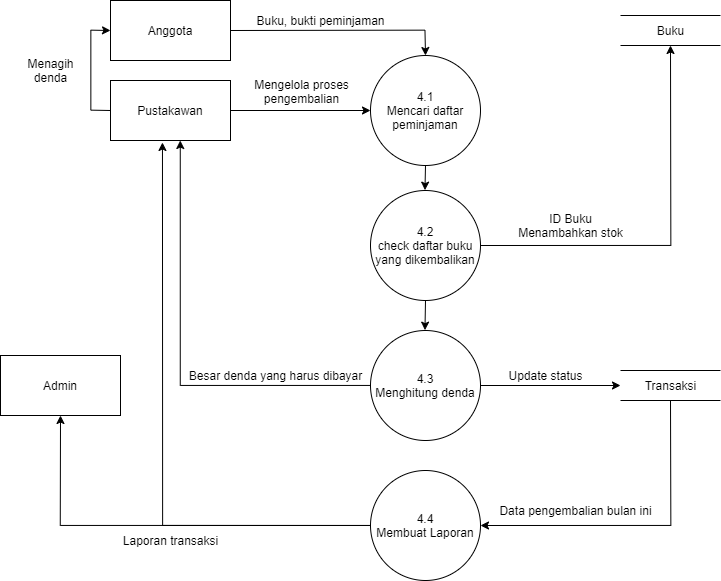
Gambar DFD Level 2 Proses 2.0 Login

* + - * 1. DFD Level 2, Proses 3.0 Peminjaman



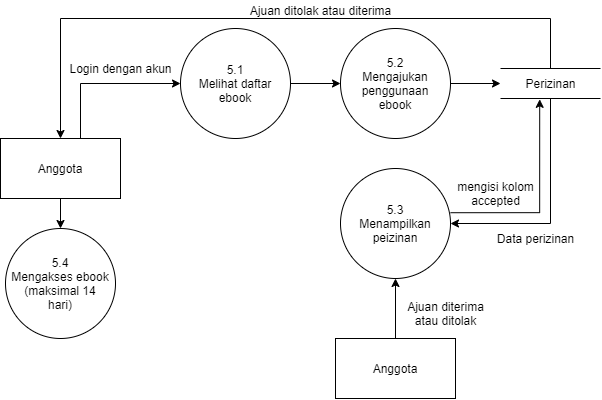
Gambar DFD Level 2 Proses 3.0 Peminjaman

* + - * 1. DFD Level 2, Proses 4.0 Pengembalian



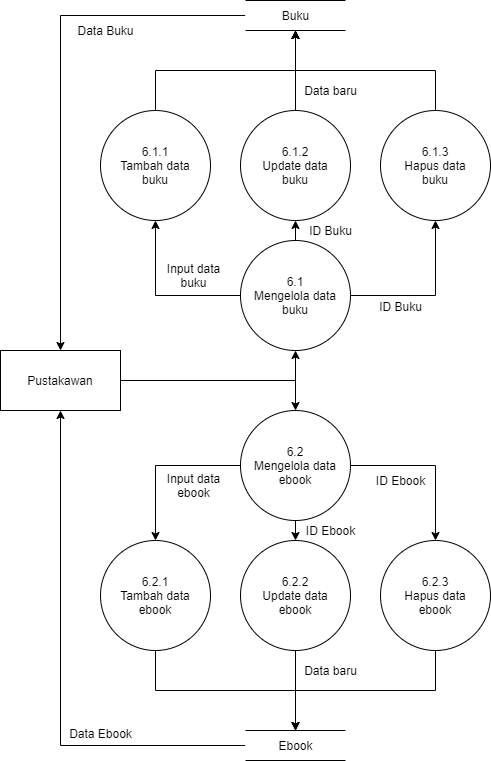
Gambar DFD Level 2 Proses 4.0 Pengembalian

* + - * 1. DFD Level 2, Proses 5.0 Pengajuan Ebook



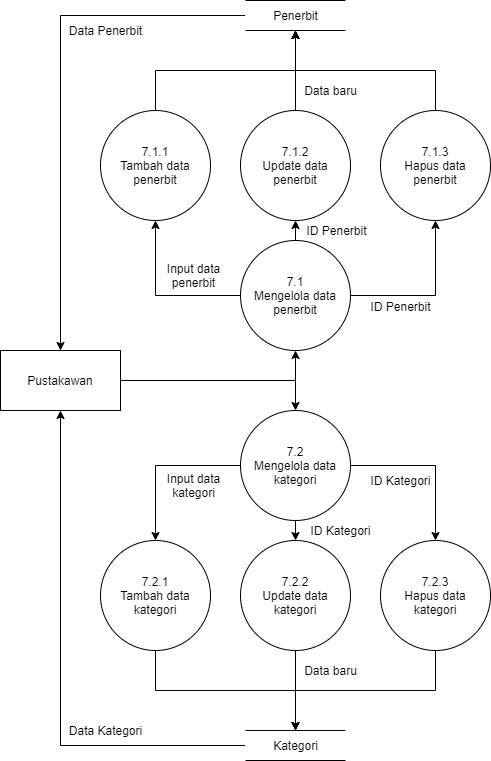
Gambar DFD Level 2 Proses 5.0 Pengajuan Ebook

* + - * 1. DFD Level 2, Proses 6.0 Pendataan Buku dan Ebook



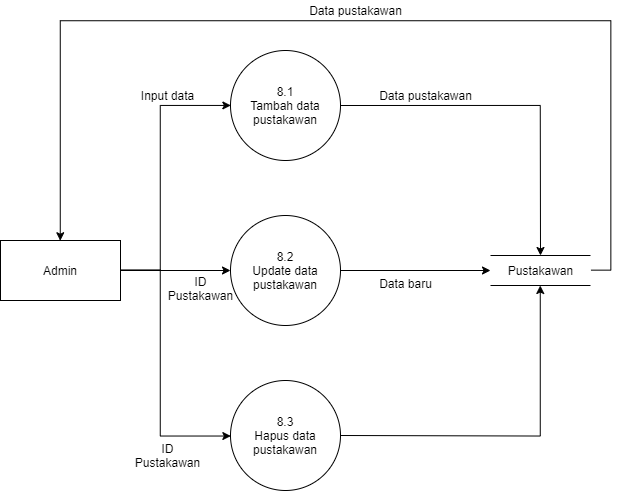
Gambar DFD Level 2 Proses 6.0 Pendataan Buku/Ebook

* + - * 1. DFD Level 2, Proses 7.0 Pendataan Identitas Buku dan Ebook



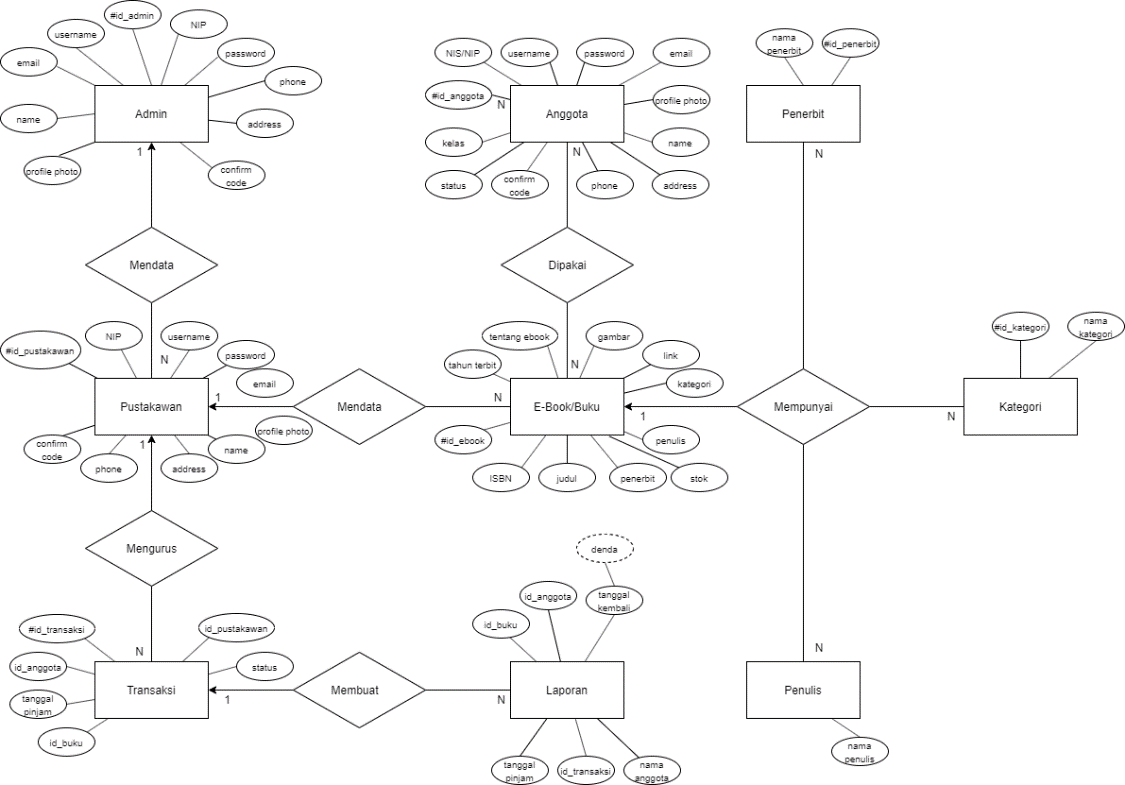
Gambar DFD Level 2 Proses 7.0 Pendataan Identitas Buku/Ebook

* + - * 1. DFD Level 2, Proses 8.0 Pendataan Pustakawan



Gambar DFD Level 2 Proses 8.0 Pendataan Pustakawan

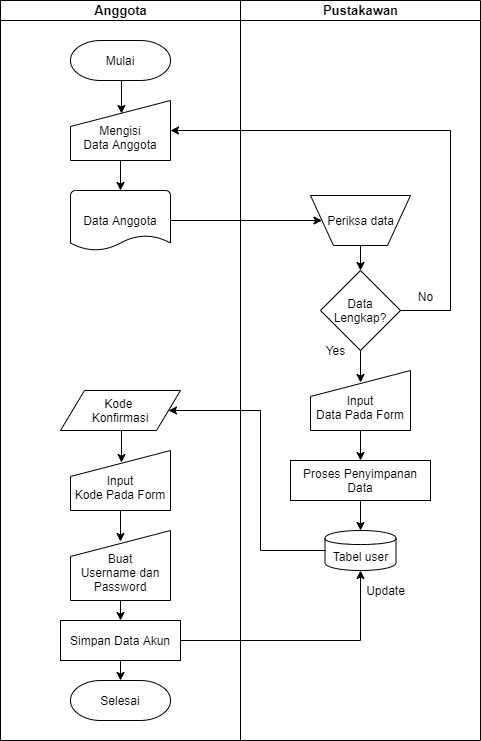
## **Entity Relationship Diagram (ERD)**

****

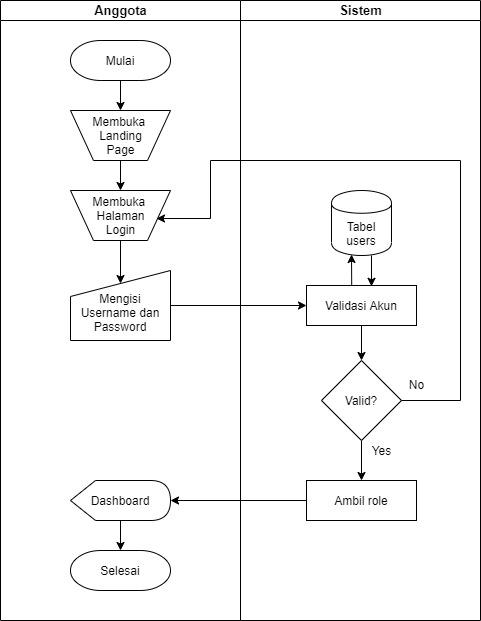
Gambar Entity Relationship Diagram (ERD)

## **Flowmap**

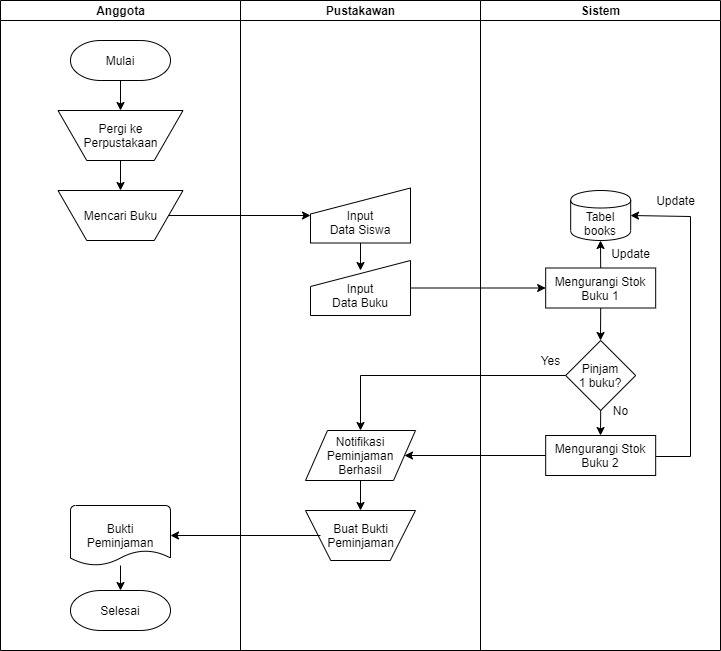
Flowmap Proses 1.0 Pendaftaran



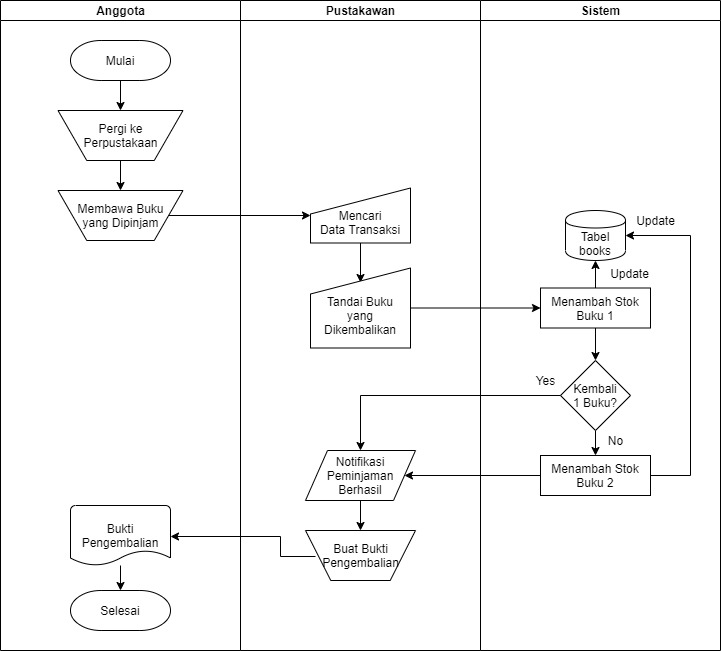
Flowmap Proses 2.0 Login



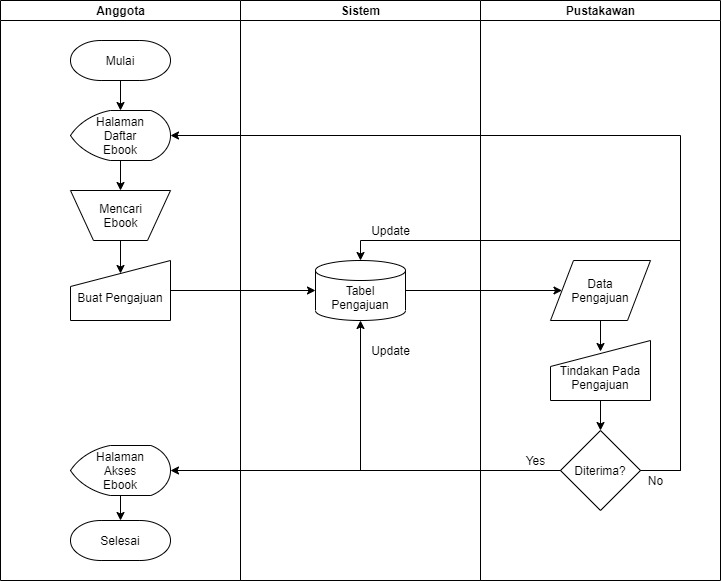
Flowmap Proses 3.0 Peminjaman



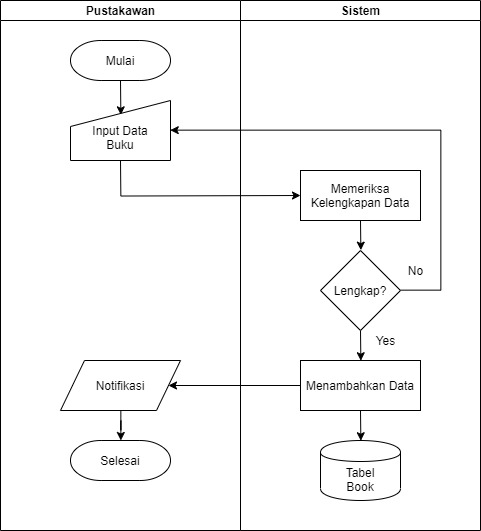
Flowmap Proses 4.0 Pengembalian



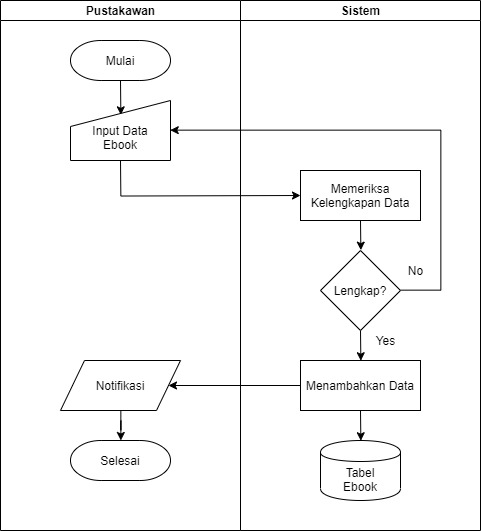
Flowmap Proses 5.0 Pengajuan



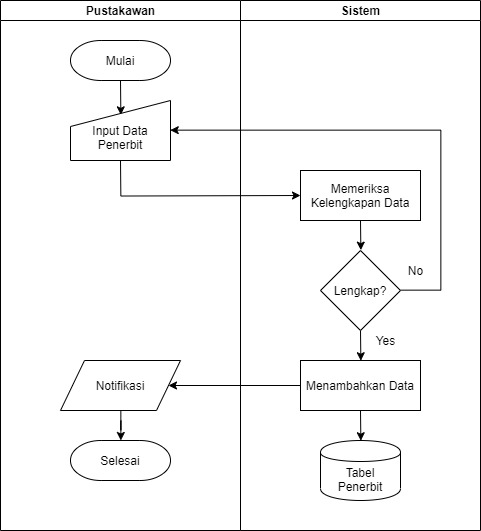
Flowmap Proses 6.0 Pendataan Buku



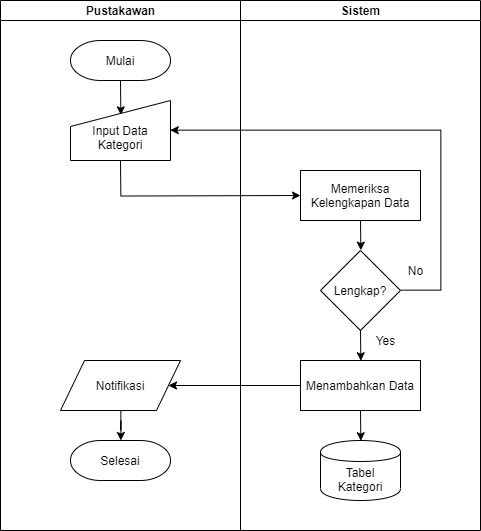
Flowmap Proses 7.0 Pendataan Ebook



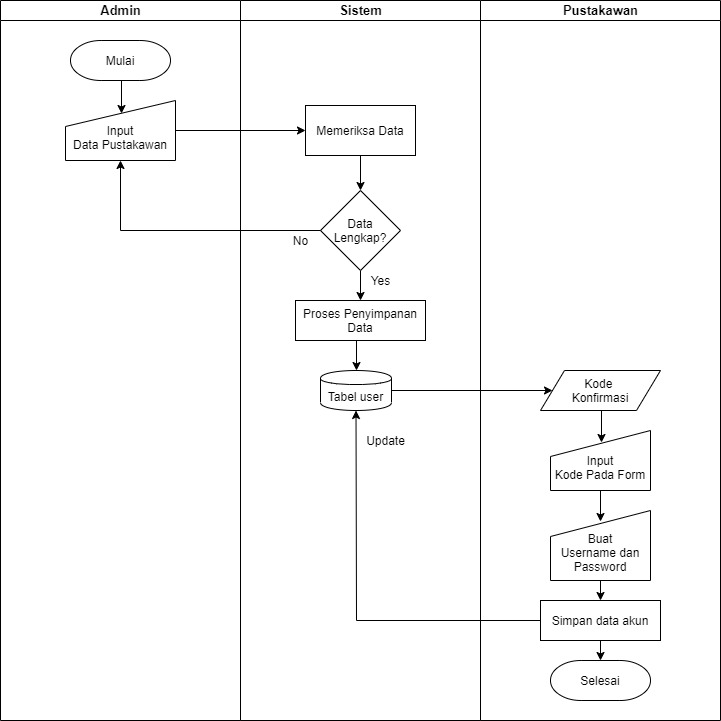
Flowmap Proses 8.0 Pendataan Penerbit



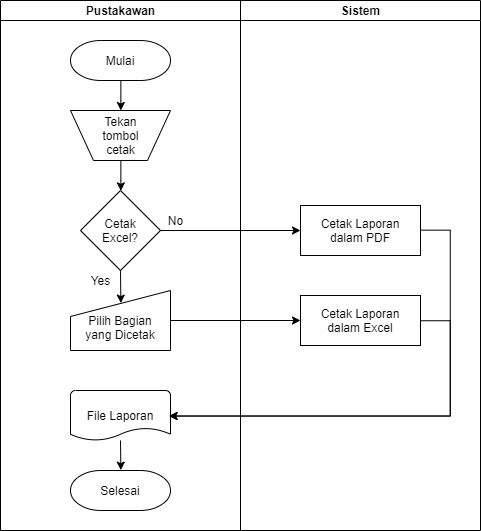
Flowmap Proses 9.0 Pendataan Kategori



Flowmap Proses 10.0 Pendataan Pustakawan



Flowmap Proses 11.0 Laporan



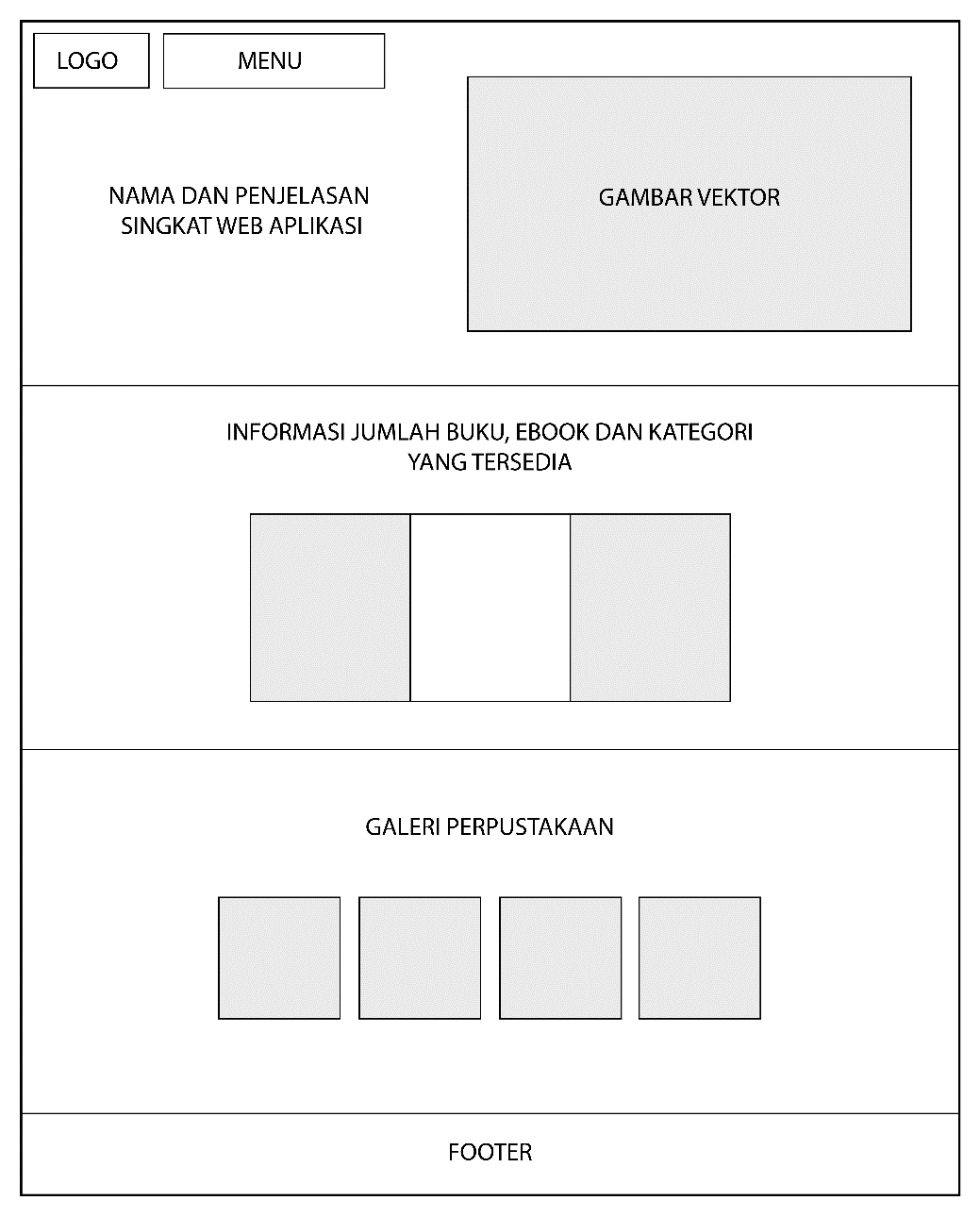
## **Perancangan Layout**

Layout merupakan sketsa desain yang berfungsi sebagai kerangka tampilan suatu aplikasi. Berikut ini adalah layout pada web aplikasi O’Library.

## **Perancangan Layout Landing Page**

Sebelum masuk ke web aplikasi, pengguna mulanya akan diarahkan ke landing page yang mengandung beberapa informasi umum yang dapat dilihat oleh siapa saja. Informasi umum tersebut meliputi nama aplikasi dan deskripsi singkat, serta beberapa data singkat yang bersumber dari database perpustakaan, dan di paling bawah halaman, akan ditampilkan beberapa gambar seputar perpustakaan.

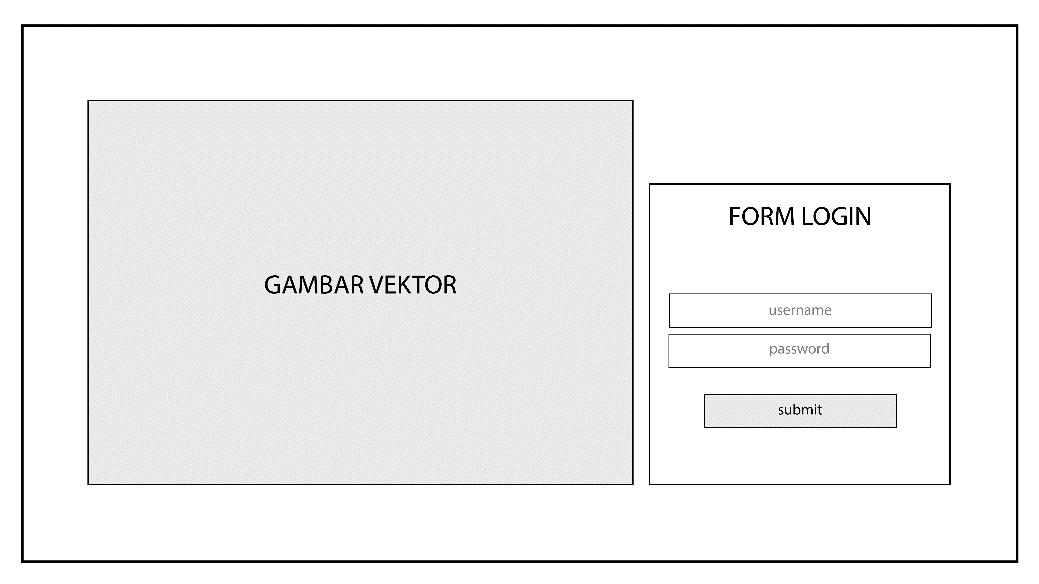
Berikut rancangan tampilan landing page.



Gambar Rancangan Layout Landing Page

## **Perancangan Layout Halaman Login**

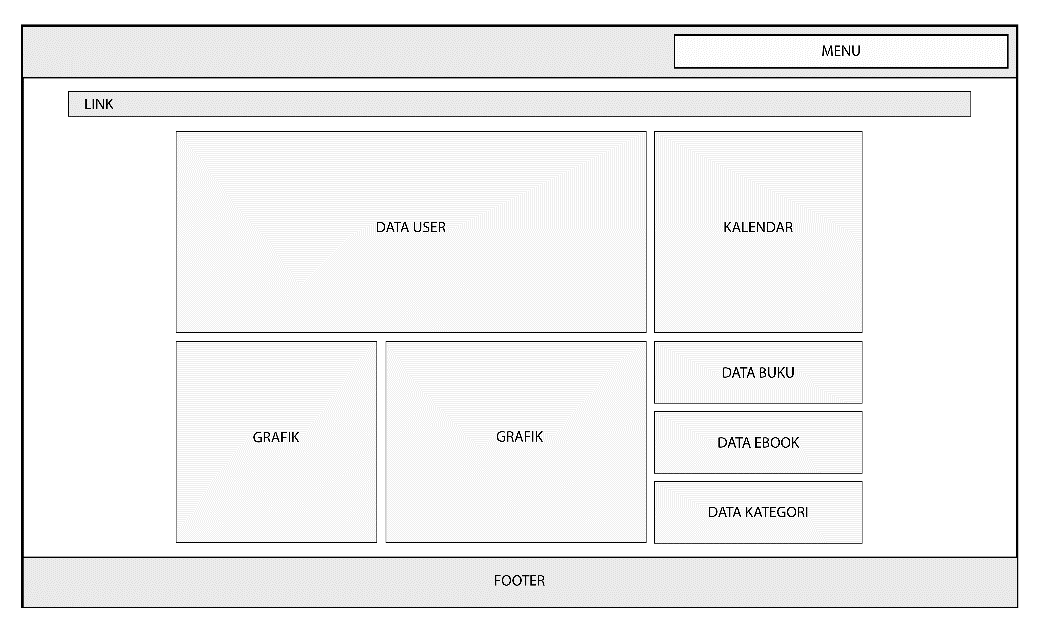
Berikut rancangan tampilan halaman login.



Gambar Rancangan Layout Halaman Login

## **Perancangan Layout Halaman Dashboard Admin/Pustakawan**

Berikut rancangan tampilan halaman dashboard untuk admin dan pustakawan.

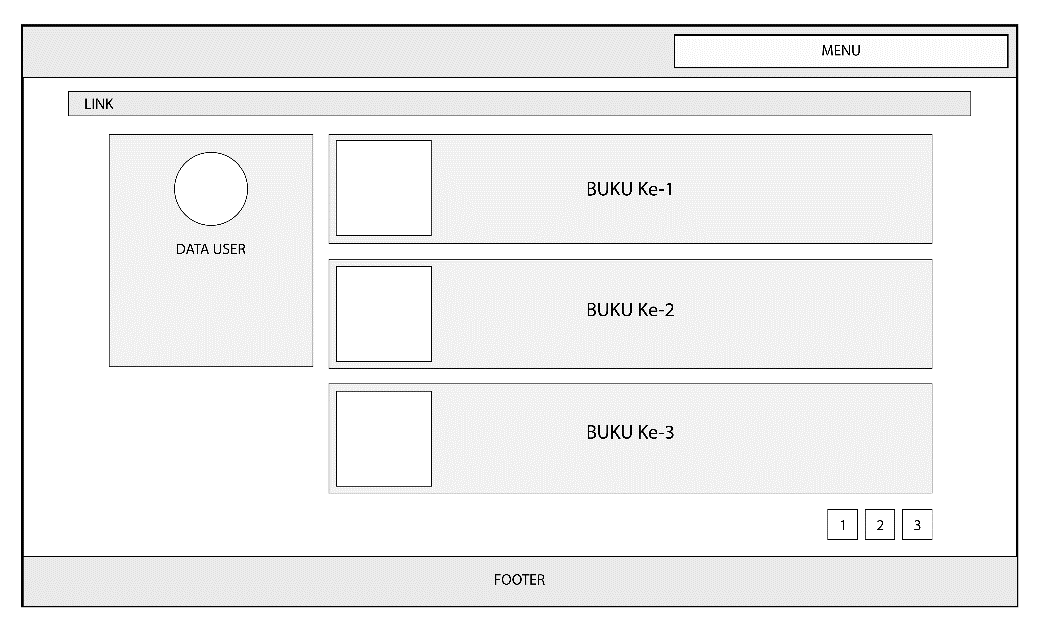


Gambar Rancangan Layout Halaman Dashboard

Admin/Pustakawan

## **Perancangan Layout Halaman Data Buku**

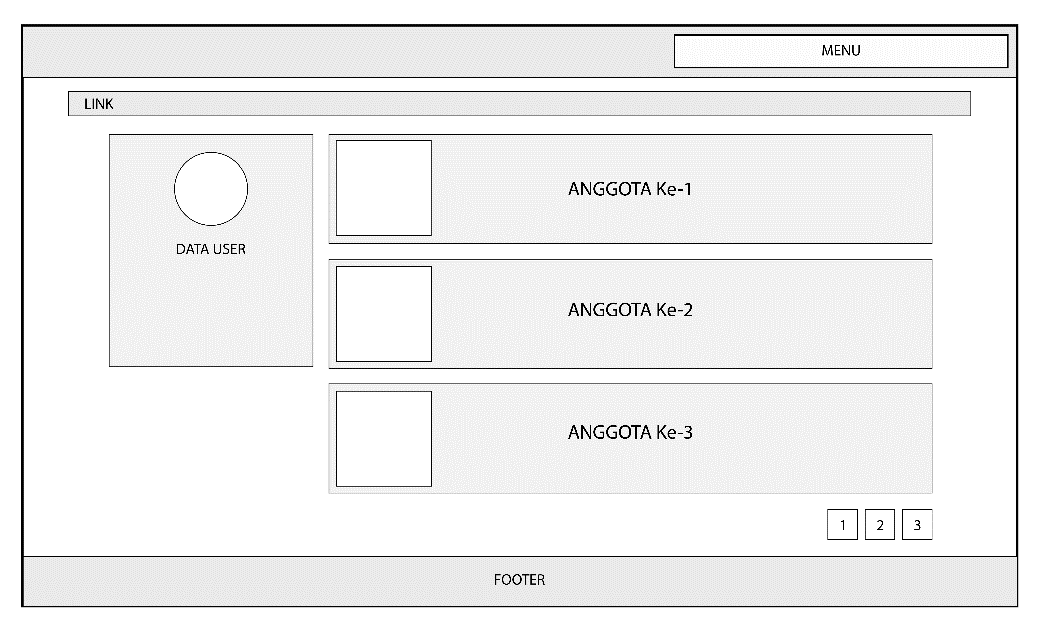
Berikut rancangan tampilan halaman data buku.



Gambar Rancangan Layout Halaman Data Buku

## **Perancangan Layout Halaman Data Anggota**

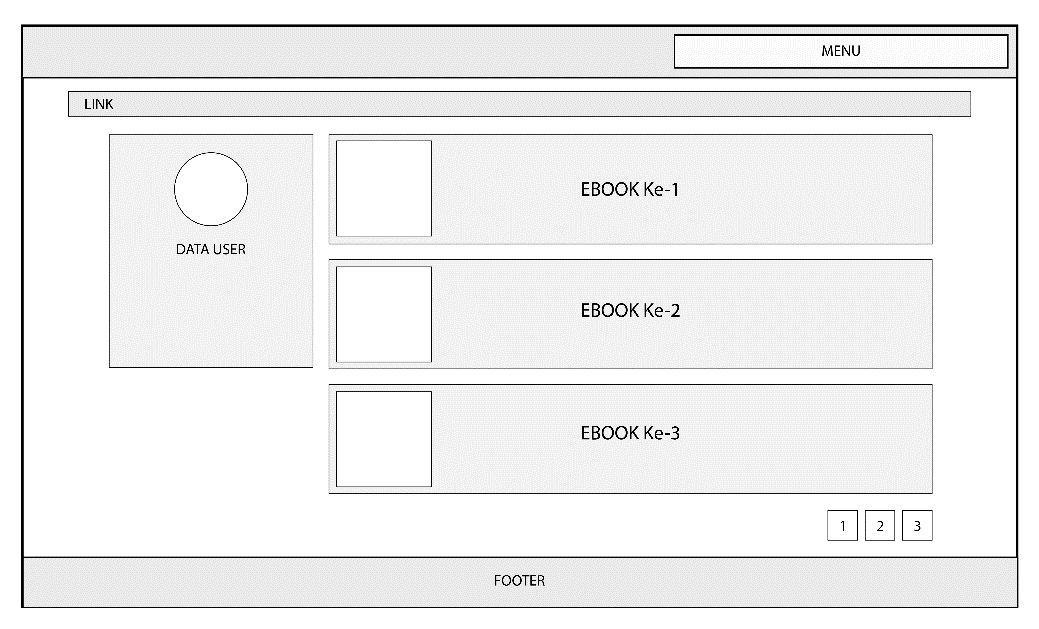
Berikut rancangan tampilan halaman data anggota.



Gambar Rancangan Layout Halaman Data Anggota

## **Perancangan Layout Halaman Data Ebook**

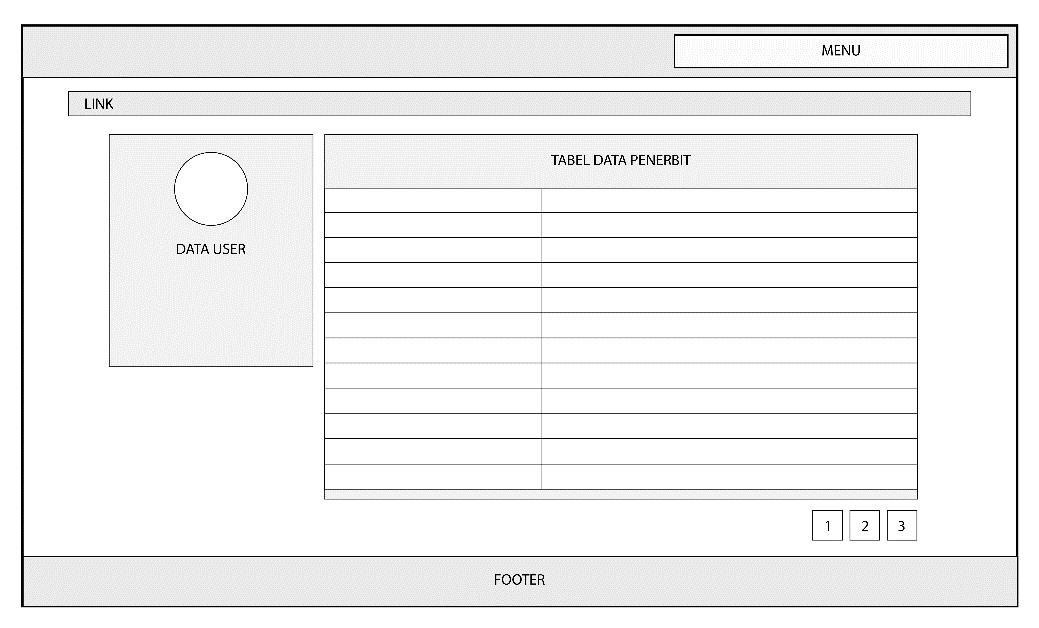
Berikut rancangan tampilan halaman data ebook.



Gambar Rancangan Layout Halaman Data Ebook

## **Perancangan Layout Halaman Data Penerbit**

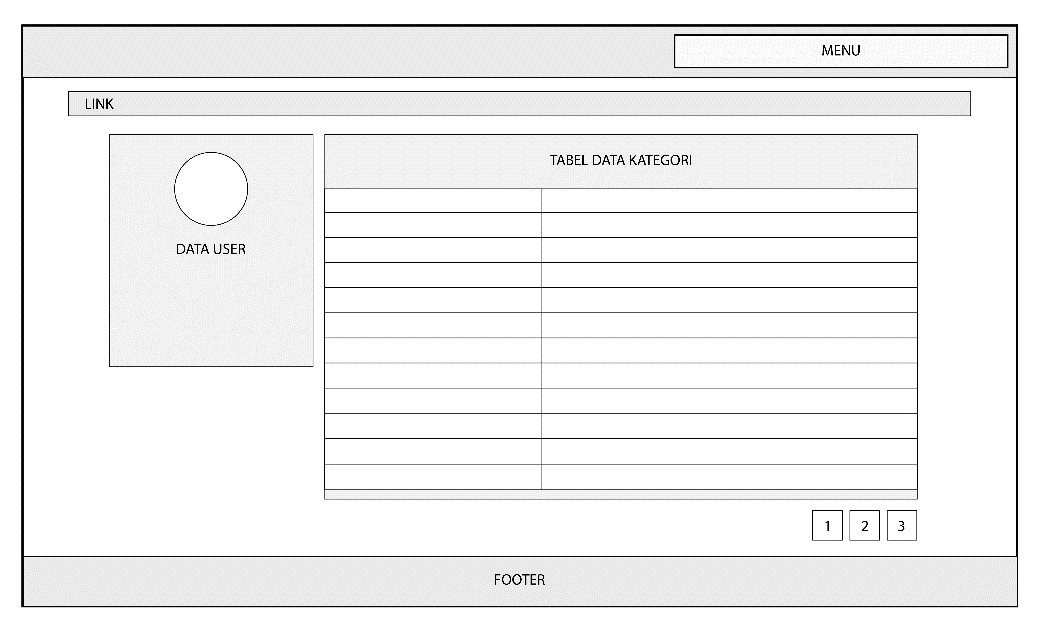
Berikut rancangan tampilan halaman data penerbit.



Gambar Rancangan Layout Halaman Data Penerbit

## **Perancangan Layout Halaman Data Kategori**

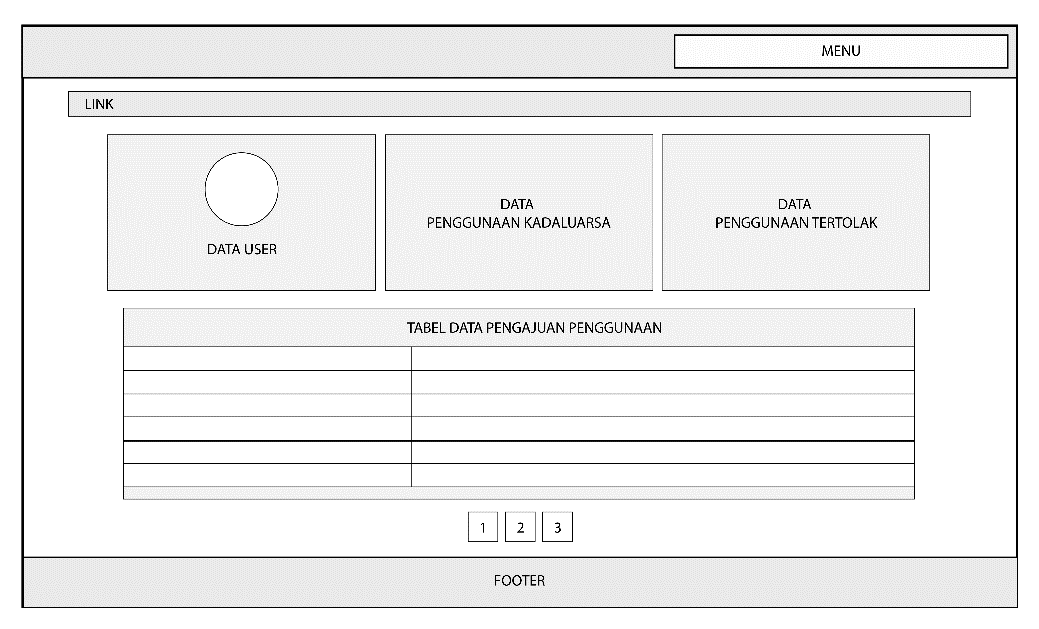
Berikut rancangan tampilan halaman data kategori.



Gambar Rancangan Layout Halaman Data Kategori

## **Perancangan Layout Halaman Data Pengajuan**

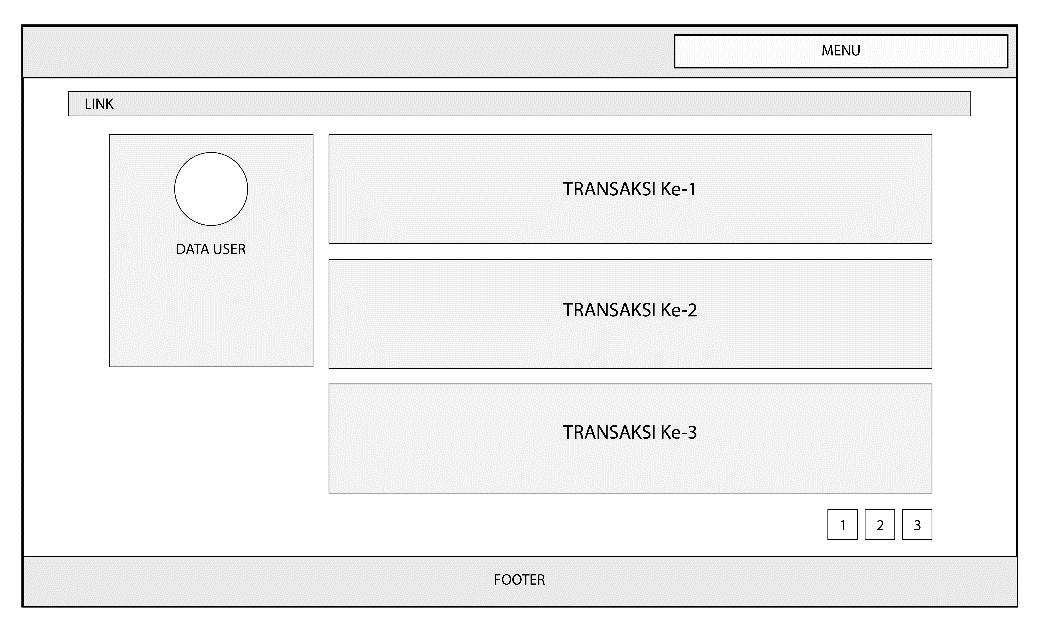
Berikut rancangan tampilan halaman data pengajuan.



Gambar Rancangan Layout Halaman Data Pengajuan

## **Perancangan Layout Halaman Data Transaksi**

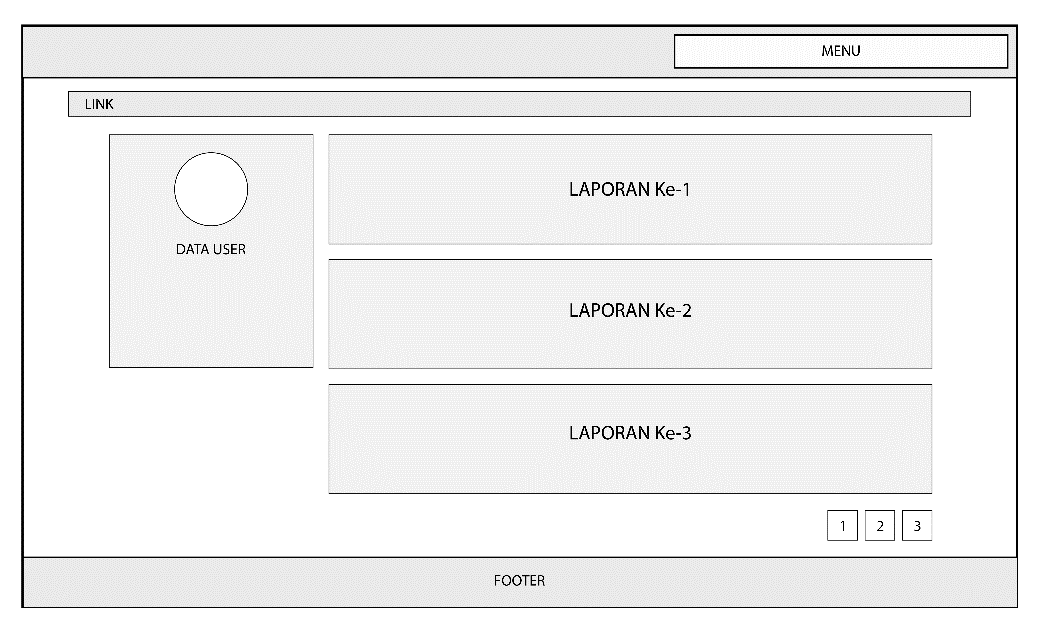
Berikut rancangan tampilan halaman data transaksi.



Gambar Rancangan Layout Halaman Data Transaksi

## **Perancangan Layout Halaman Data Laporan**

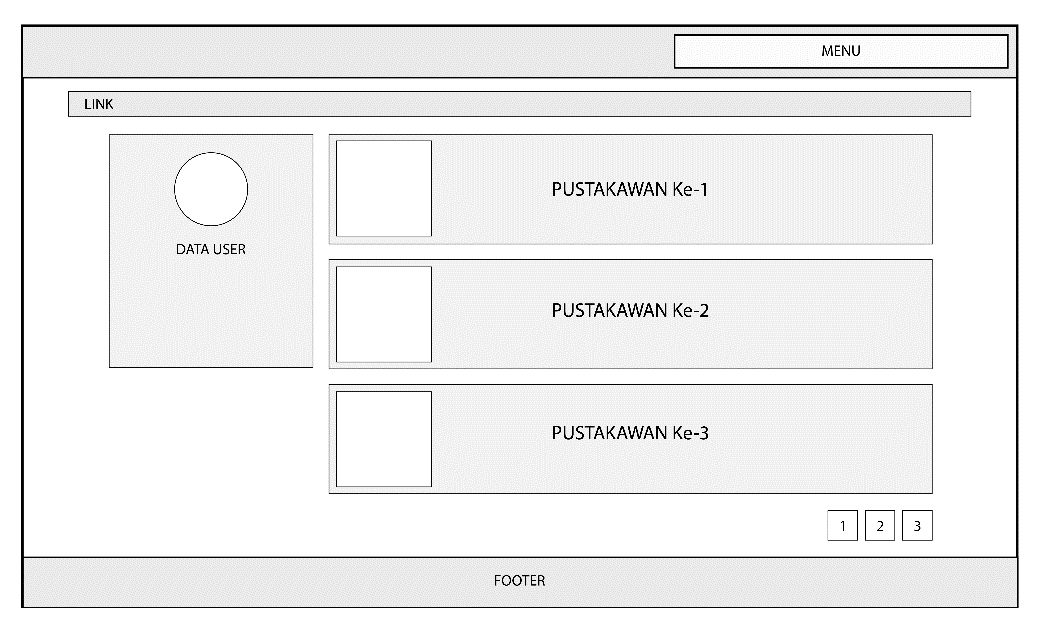
Berikut rancangan tampilan halaman data laporan.



Gambar Rancangan Layout Halaman Data Laporan

## **Perancangan Layout Halaman Data Pustakawan/Admin**

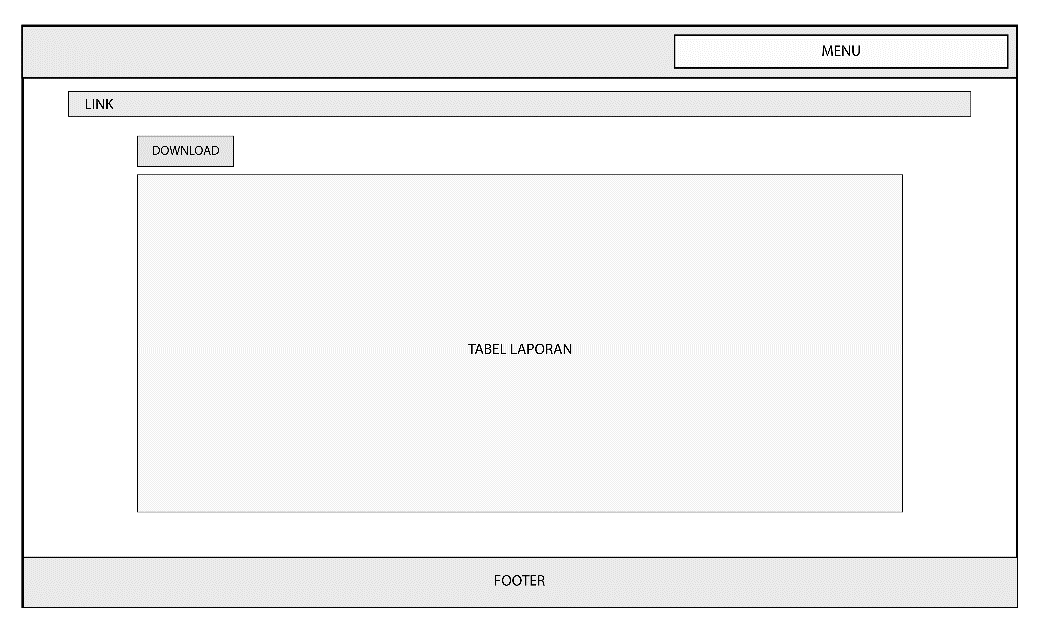
Berikut rancangan tampilan halaman data pustakawan dan admin.



Gambar Rancangan Layout Halaman Data Pustakawan/Admin

## **Perancangan Layout Halaman Cetak Laporan**

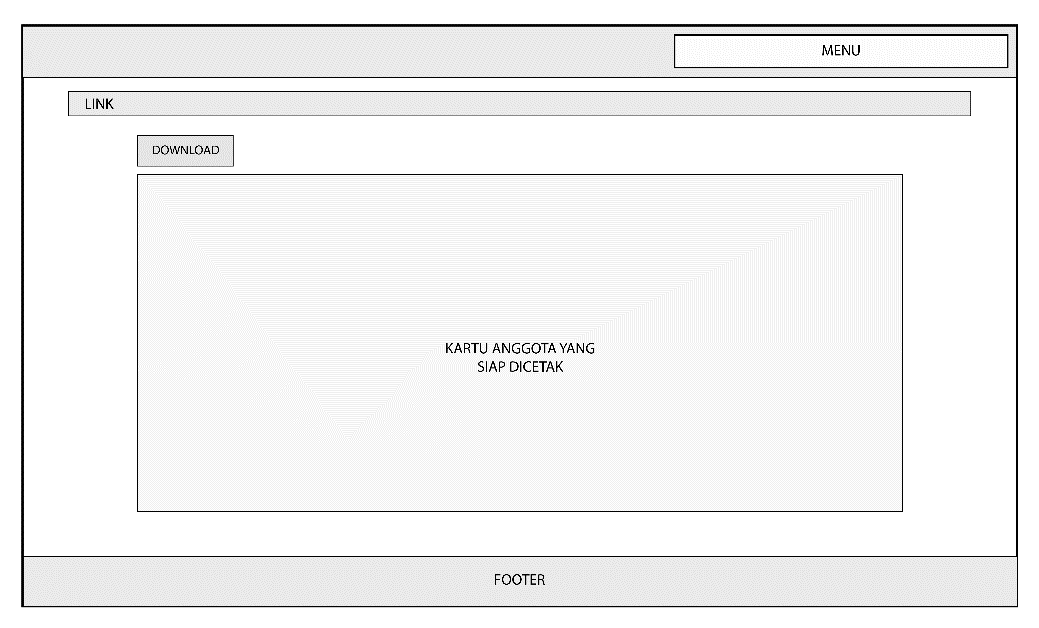
Berikut rancangan tampilan halaman untuk mencetak laporan menjadi file PDF.



Gambar Rancangan Layout Halaman Cetak Laporan

## **Perancangan Layout Halaman Cetak Kartu Anggota**

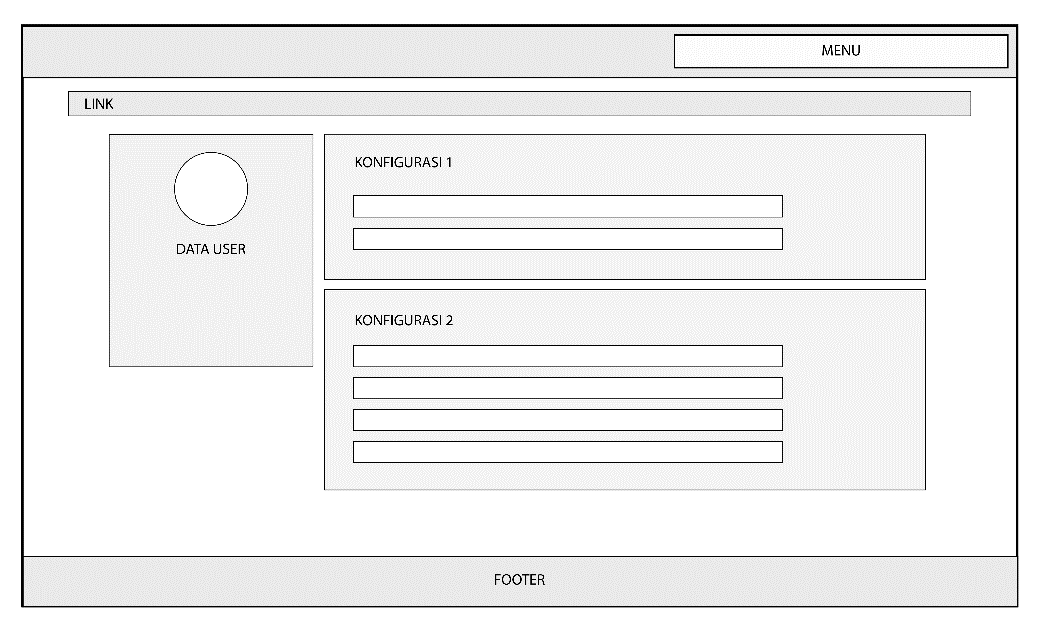
Berikut rancangan tampilan halaman untuk mencetak kartu anggota.



Gambar Rancangan Layout Halaman Cetak Kartu Anggota

## **Perancangan Layout Halaman Konfigurasi**

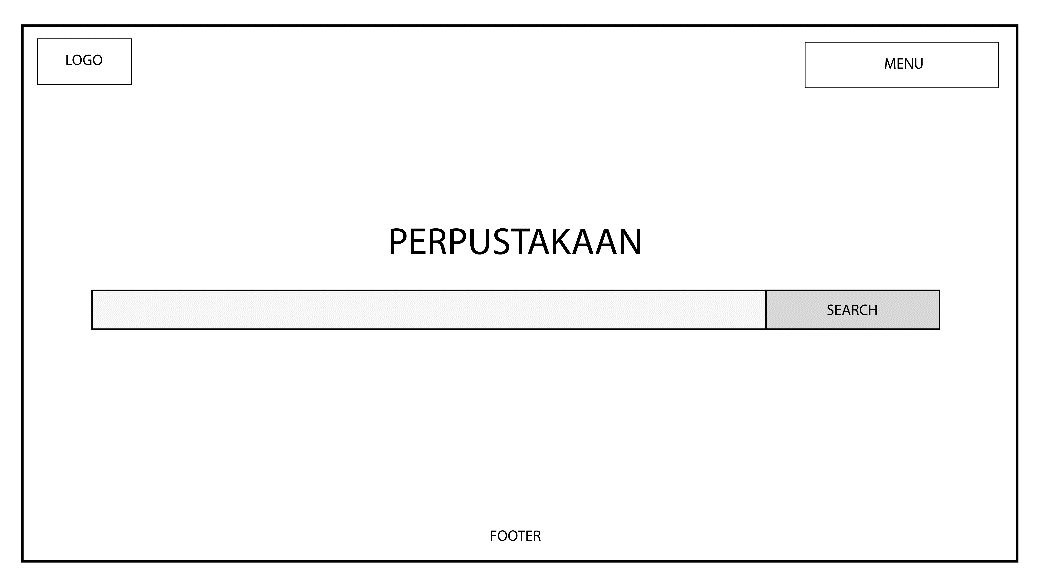
Berikut rancangan tampilan halaman konfigurasi.



Gambar Rancangan Layout Halaman Konfigurasi

## **Perancangan Layout Halaman Dashboard Anggota**

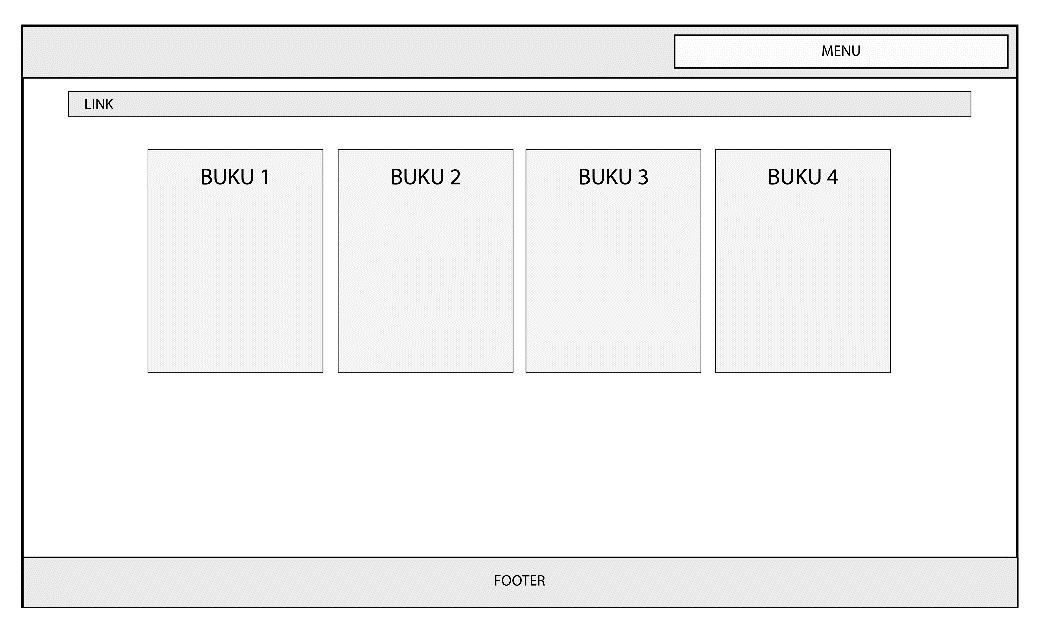
Berikut rancangan tampilan halaman dashboard untuk anggota.



Gambar Rancangan Layout Halaman Dashboard Anggota

## **Perancangan Layout Halaman Data Buku Anggota**

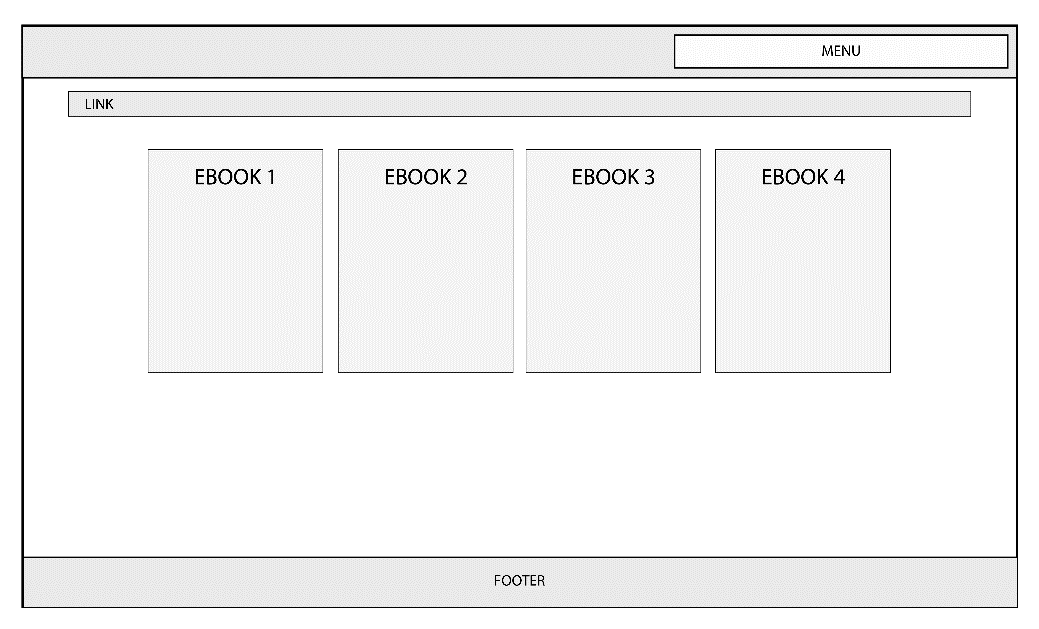
Berikut rancangan tampilan halaman data buku untuk anggota.



Gambar Rancangan Layout Halaman Data Buku Anggota

## **Perancangan Layout Halaman Data Ebook Anggota**

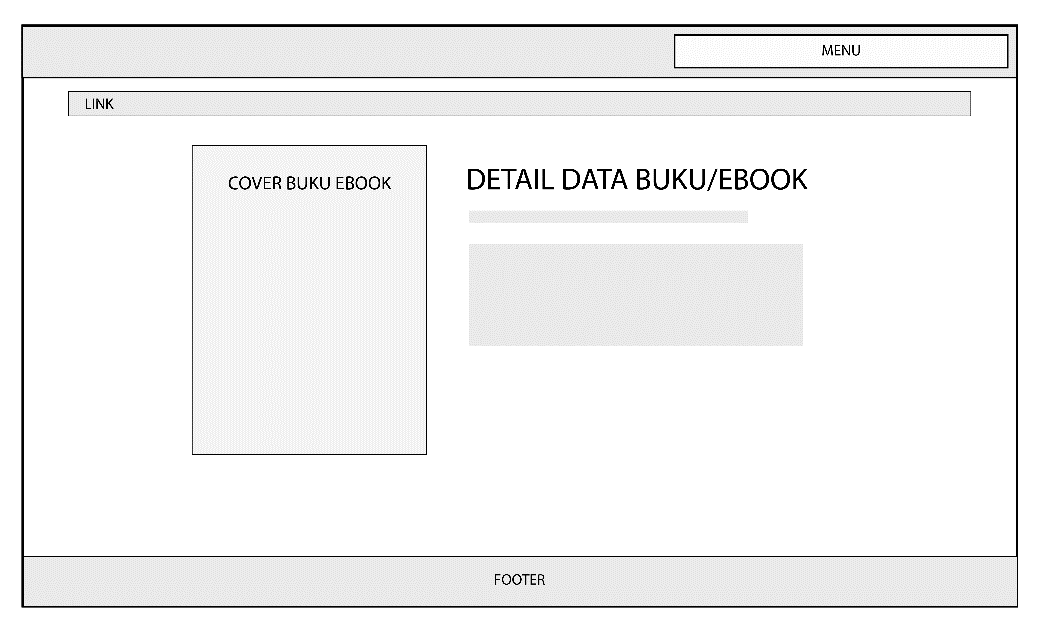
Berikut rancangan tampilan halaman data ebook untuk anggota.



Gambar Rancangan Layout Halaman Data Ebook Anggota

## **Perancangan Layout Halaman Detail Buku/Ebook**

Berikut rancangan tampilan halaman detail buku dan ebook.



Gambar Rancangan Layout Halaman Detail Buku/Ebook

# **BAB IV IMPLEMENTASI DAN PENGUJIAN**

## **Implementasi Tampilan Aplikasi**

## **Implementasi Tampilan Halaman Landing Page**

## **Implementasi Tampilan Halaman Login**

## **Implementasi Tampilan Halaman Dashboard Admin/Pustakawan**

## **Implementasi Tampilan Halaman Data Buku**

## **Implementasi Tampilan Halaman Data Anggota**

## **Implementasi Tampilan Halaman Data Ebook**

## **Implementasi Tampilan Halaman Data Penerbit**

## **Implementasi Tampilan Halaman Data Kategori**

## **Implementasi Tampilan Halaman Data Pengajuan Ebook**

## **Implementasi Tampilan Halaman Data Transaksi**

## **Implementasi Tampilan Halaman Data Laporan**

## **Implementasi Tampilan Halaman Data Pustakawan**

## **Implementasi Tampilan Halaman Cetak Laporan**

## **Implementasi Tampilan Halaman Cetak Kartu Anggota**

## **Implementasi Tampilan Halaman Konfigurasi**

## **Implementasi Tampilan Halaman Dashboard Anggota**

## **Implementasi Tampilan Halaman Data Buku Anggota**

## **Implementasi Tampilan Halaman Data Ebook Anggota**

## **Implementasi Tampilan Halaman Detail Buku/Ebook**

## **Struktur Database**

## **Pengujian**

# **BAB V PENUTUP**

## **Kesimpulan**

## **Saran**

# **DAFTAR PUSTAKA**