**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**

PENGESAHAN DARI PIHAK SEKOLAH

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

Laporan ini telah disetujui oleh:

|  |  |
| --- | --- |
| Ketua Paket Keahlian, | Pembimbing, |
|  |  |
| AGUS RAHMAWAN, S.T. | SITI MARYAM, S.Kom., M.Kom. |
| NIP. 19751202 200604 1 010 | NIP. 19721010 200604 2 020 |

|  |
| --- |
| MENGETAHUI: |
| Kepala SMK Negeri 1 Cimahi |
|  |
| Drs. DAUD SALEH, M.M. |
| NIP. 19630718 198902 1 001 |

# **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 |
|  |  |  |
|  |  | Ilham Shiddiq |

# **DAFTAR ISI**

[**KATA PENGANTAR** i](#_Toc64042352)

[**DAFTAR ISI** iii](#_Toc64042353)

[**DAFTAR GAMBAR** vii](#_Toc64042354)

[**DAFTAR TABEL** x](#_Toc64042355)

[**BAB I PENDAHULUAN** 1](#_Toc64042356)

[1.1 Latar Belakang Masalah 1](#_Toc64042357)

[1.2 Tujuan 2](#_Toc64042358)

[1.3 Pembatasan Masalah 2](#_Toc64042359)

[1.4 Sistematika Pembahasan 3](#_Toc64042360)

[**BAB II LANDASAN TEORI** 5](#_Toc64042361)

[2.1. Perpustakaan 5](#_Toc64042362)

[2.2. *Ebook* 5](#_Toc64042363)

[2.3. Data 6](#_Toc64042364)

[2.4. *Website* 6](#_Toc64042365)

[2.4.1 *Website* Statis 6](#_Toc64042366)

[2.4.2 *Website* Dinamis 7](#_Toc64042367)

[2.5. *Web* Aplikasi 7](#_Toc64042368)

[2.6. HTML (*Hypertext Markup Language*) 7](#_Toc64042369)

[2.7. *Blade* 8](#_Toc64042370)

[2.8. CSS (*Cascading Style Sheet*) 8](#_Toc64042371)

[2.9. SASS (*Syntactically Awesome Style Sheet*) 9](#_Toc64042372)

[2.10. Javascript 9](#_Toc64042373)

[2.11. Jquery 10](#_Toc64042374)

[2.12. AJAX (*Asyncronous Javascript and XML*) 10](#_Toc64042375)

[2.13. PHP : *Hypertext Preprocessor* 11](#_Toc64042376)

[2.14. *Framework* 11](#_Toc64042377)

[2.15. Konsep MVC 11](#_Toc64042378)

[2.16. Laravel 12](#_Toc64042379)

[2.17. Bootstrap 12](#_Toc64042380)

[2.18. Basis Data (*Database*) 13](#_Toc64042381)

[2.19. MySQL 13](#_Toc64042382)

[2.20. Webpack 14](#_Toc64042383)

[2.21. XAMPP 14](#_Toc64042384)

[2.22. DFD (*Data Flow Diagram*) 14](#_Toc64042385)

[2.23. ERD (*Entity Relationship Diagram*) 15](#_Toc64042386)

[2.24. *Use Case Diagram* 16](#_Toc64042387)

[2.25. *Flowmap* 17](#_Toc64042388)

[**BAB III ANALISA DAN PERANCANGAN** 19](#_Toc64042389)

[3.1. Analisa Aplikasi 19](#_Toc64042390)

[3.1.1. Analisa Sistem Pengguna 19](#_Toc64042391)

[3.1.2. Analisa Kebutuhan Sistem 20](#_Toc64042392)

[3.2. Perancangan Sistem 21](#_Toc64042393)

[3.2.1. *Use* *Case* *Diagram* 21](#_Toc64042394)

[3.2.2. *Data* *Flow* *Diagram* (DFD) 22](#_Toc64042395)

[3.2.3. *Entity* *Relationship* *Diagram* (ERD) 30](#_Toc64042396)

[*3.2.4.* *Flowmap* 31](#_Toc64042397)

[3.3. Perancangan *Layout* 42](#_Toc64042398)

[3.3.1. Perancangan *Layout* *Landing* *Page* 42](#_Toc64042399)

[3.3.2. Perancangan *Layout* Halaman *Login* 43](#_Toc64042400)

[3.3.3. Perancangan *Layout* Halaman *Dashboard* *Admin*/Pustakawan 44](#_Toc64042401)

[3.3.4. Perancangan *Layout* Halaman Data Buku 44](#_Toc64042402)

[3.3.5. Perancangan *Layout* Halaman Data Anggota 45](#_Toc64042403)

[3.3.6. Perancangan *Layout* Halaman Data *Ebook* 45](#_Toc64042404)

[3.3.7. Perancangan *Layout* Halaman Data Penerbit 46](#_Toc64042405)

[3.3.8. Perancangan *Layout* Halaman Data Kategori 46](#_Toc64042406)

[3.3.9. Perancangan *Layout* Halaman Data Pengajuan 47](#_Toc64042407)

[3.3.10. Perancangan *Layout* Halaman Data Transaksi 47](#_Toc64042408)

[3.3.11. Perancangan *Layout* Halaman Data Laporan 48](#_Toc64042409)

[3.3.12. Perancangan *Layout* Halaman Data Pustakawan/*Admin* 48](#_Toc64042410)

[3.3.13. Perancangan *Layout* Halaman Cetak Laporan 49](#_Toc64042411)

[3.3.14. Perancangan *Layout* Halaman Cetak Kartu Anggota 49](#_Toc64042412)

[3.3.15. Perancangan *Layout* Halaman Konfigurasi 50](#_Toc64042413)

[3.3.16. Perancangan *Layout* Halaman *Dashboard* Anggota 50](#_Toc64042414)

[3.3.17. Perancangan *Layout* Halaman Data Buku Anggota 51](#_Toc64042415)

[3.3.18. Perancangan *Layout* Halaman Data *Ebook* Anggota 51](#_Toc64042416)

[3.3.19. Perancangan *Layout* Halaman Detail Buku/*Ebook* 52](#_Toc64042417)

[**BAB IV IMPLEMENTASI DAN PENGUJIAN** 53](#_Toc64042418)

[4.1. Implementasi Sistem 53](#_Toc64042419)

[4.2. Implementasi Tampilan Aplikasi 58](#_Toc64042420)

[4.2.1. Implementasi Tampilan Halaman *Landing* *Page* 58](#_Toc64042421)

[4.2.2. Implementasi Tampilan Halaman *Login* 59](#_Toc64042422)

[4.2.3. Implementasi Tampilan Halaman *Dashboard* *Admin*/Pustakawan 59](#_Toc64042423)

[4.2.4. Implementasi Tampilan Halaman Data Buku 60](#_Toc64042424)

[4.2.5. Implementasi Tampilan Halaman Data Anggota 60](#_Toc64042425)

[4.2.6. Implementasi Tampilan Halaman Data *Ebook* 61](#_Toc64042426)

[4.2.7. Implementasi Tampilan Halaman Data Penerbit 62](#_Toc64042427)

[4.2.8. Implementasi Tampilan Halaman Data Kategori 63](#_Toc64042428)

[4.2.9. Implementasi Tampilan Halaman Data Pengajuan *Ebook* 63](#_Toc64042429)

[4.2.10. Implementasi Tampilan Halaman Data Transaksi 64](#_Toc64042430)

[4.2.11. Implementasi Tampilan Halaman Data Laporan 64](#_Toc64042431)

[4.2.12. Implementasi Tampilan Halaman Data Pustakawan 65](#_Toc64042432)

[4.2.13. Implementasi Tampilan Halaman Cetak Laporan 65](#_Toc64042433)

[4.2.14. Implementasi Tampilan Halaman Cetak Kartu Anggota 66](#_Toc64042434)

[4.2.15. Implementasi Tampilan Halaman Konfigurasi 67](#_Toc64042435)

[4.2.16. Implementasi Tampilan Halaman Dashboard Anggota 68](#_Toc64042436)

[4.2.17. Implementasi Tampilan Halaman Data Buku Anggota 68](#_Toc64042437)

[4.2.18. Implementasi Tampilan Halaman Data *Ebook* Anggota 69](#_Toc64042438)

[4.2.19. Implementasi Tampilan Halaman Detail Buku/*Ebook* 69](#_Toc64042439)

[4.3. Struktur *Database* 70](#_Toc64042440)

[4.4. Diagram Relasi 75](#_Toc64042441)

[4.5. Pengujian 76](#_Toc64042442)

[*4.5.1.* *Login* 76](#_Toc64042443)

[4.5.2. Konfirmasi Akun 77](#_Toc64042444)

[4.5.3. Pendataan Buku 78](#_Toc64042445)

[4.5.4. Pendataan Anggota 79](#_Toc64042446)

[4.5.5. Pendataan *Ebook* 80](#_Toc64042447)

[4.5.6. Pendataan Pustakawan/*Admin* 82](#_Toc64042448)

[4.5.7. Pendataan Penerbit 83](#_Toc64042449)

[4.5.8. Pendataan Kategori 84](#_Toc64042450)

[4.5.9. Pendataan Pengajuan Penggunaan *Ebook* 85](#_Toc64042451)

[4.5.10. Pendataan Transaksi 86](#_Toc64042452)

[4.5.11. Laporan 87](#_Toc64042453)

[4.5.12. Konfigurasi 88](#_Toc64042454)

[4.5.13. Pengajuan *Ebook* 89](#_Toc64042455)

[4.5.14. Keamanan 90](#_Toc64042456)

[4.5.15. Mengganti *Password* 91](#_Toc64042457)

[**BAB V PENUTUP** 93](#_Toc64042458)

[5.1. Kesimpulan 93](#_Toc64042459)

[5.2. Saran 93](#_Toc64042460)

[**DAFTAR PUSTAKA** 94](#_Toc64042461)

# **DAFTAR GAMBAR**

[Gambar 2. 1 Logo dari HTML 8](#_Toc64033320)

[Gambar 2. 2 Logo dari CSS 9](#_Toc64033321)

[Gambar 2. 3 Logo dari SASS 9](#_Toc64033322)

[Gambar 2. 4 Logo dari Javascript 10](#_Toc64033323)

[Gambar 2. 5 Logo dari PHP 11](#_Toc64033324)

[Gambar 2. 6 Logo dari Laravel 12](#_Toc64033325)

[Gambar 2. 7 Logo dari Bootstrap 13](#_Toc64033326)

[Gambar 2. 8 Logo dari XAMPP 14](#_Toc64033327)

[Gambar 3. 1 *Use Case Diagram* 22](#_Toc64038066)

[Gambar 3. 2 DFD Konteks Diagram 22](#_Toc64038067)

[Gambar 3. 3 DFD *Level* 1 23](#_Toc64038068)

[Gambar 3. 4 DFD *Level* 2, Proses 1.0 Pendaftaran 24](#_Toc64038069)

[Gambar 3. 5 DFD *Level* 2, Proses 2.0 *Login* 24](#_Toc64038070)

[Gambar 3. 6 DFD *Level* 2, Proses 3.0 Peminjaman 25](#_Toc64038071)

[Gambar 3. 7 DFD *Level* 2, Proses 4.0 Pengembalian 26](#_Toc64038072)

[Gambar 3. 8 DFD *Level* 2, Proses 5.0 Pengajuan *Ebook* 27](#_Toc64038073)

[Gambar 3. 9 DFD *Level* 2, Proses 6.0 Pendataan Buku/*Ebook* 28](#_Toc64038074)

[Gambar 3. 10 DFD *Level* 2, Proses 7.0 Pendataan Identitas Buku/*Ebook* 29](#_Toc64038075)

[Gambar 3. 11 DFD *Level* 2, Proses 8.0 Pendataan Pustakawan 30](#_Toc64038076)

[Gambar 3. 12 *Entity Relationship Diagram* (ERD) 31](#_Toc64038077)

[Gambar 3. 13 *Flowmap* Proses 1.0 Pendaftaran 32](#_Toc64038078)

[Gambar 3. 14 *Flowmap* Proses 2.0 *Login* 33](#_Toc64038079)

[Gambar 3. 15 *Flowmap* Proses 3.0 Peminjaman 34](#_Toc64038080)

[Gambar 3. 16 *Flowmap* Proses 4.0 Pengembalian 35](#_Toc64038081)

[Gambar 3. 17 *Flowmap* Proses 5.0 Pengajuan 36](#_Toc64038082)

[Gambar 3. 18 *Flowmap* Proses 6.0 Pendataan Buku 37](#_Toc64038083)

[Gambar 3. 19 *Flowmap* Proses 7.0 Pendataan *Ebook* 38](#_Toc64038084)

[Gambar 3. 20 *Flowmap* Proses 8.0 Pendataan Penerbit 39](#_Toc64038085)

[Gambar 3. 21 *Flowmap* Proses 9.0 Pendataan Kategori 40](#_Toc64038086)

[Gambar 3. 22 *Flowmap* Proses 10.0 Pendataan Pustakawan 41](#_Toc64038087)

[Gambar 3. 23 *Flowmap* Proses 11.0 Laporan 42](#_Toc64038088)

[Gambar 3. 24 Rancangan *Layout Landing Page* 43](#_Toc64038089)

[Gambar 3. 25 Rancangan *Layout* Halaman *Login* 44](#_Toc64038090)

[Gambar 3. 26 Rancangan *Layout* Halaman *Dashboard* *Admin*/Pustakawan 44](#_Toc64038091)

[Gambar 3. 27 Rancangan *Layout* Halaman Data Buku 45](#_Toc64038092)

[Gambar 3. 28 Rancangan *Layout* Halaman Data Anggota 45](#_Toc64038093)

[Gambar 3. 29 Rancangan *Layout* Halaman Data *Ebook* 46](#_Toc64038094)

[Gambar 3. 30 Rancangan *Layout* Halaman Data Penerbit 46](#_Toc64038095)

[Gambar 3. 31 Rancangan *Layout* Halaman Data Kategori 47](#_Toc64038096)

[Gambar 3. 32 Rancangan *Layout* Halaman Data Pengajuan 47](#_Toc64038097)

[Gambar 3. 33 Rancangan *Layout* Halaman Data Transaksi 48](#_Toc64038098)

[Gambar 3. 34 Rancangan *Layout* Halaman Data Laporan 48](#_Toc64038099)

[Gambar 3. 35 Rancangan *Layout* Halaman Data Pustakawan/*Admin* 49](#_Toc64038100)

[Gambar 3. 36 Rancangan *Layout* Halaman Cetak Laporan 49](#_Toc64038101)

[Gambar 3. 37 Rancangan *Layout* Halaman Cetak Kartu Anggota 50](#_Toc64038102)

[Gambar 3. 38 Rancangan *Layout* Halaman Konfigurasi 50](#_Toc64038103)

[Gambar 3. 39 Rancangan *Layout* Halaman *Dashboard* Anggota 51](#_Toc64038104)

[Gambar 3. 40 Rancangan *Layout* Halaman Data Buku Anggota 51](#_Toc64038105)

[Gambar 3. 41 Rancangan *Layout* Halaman Data *Ebook* Anggota 52](#_Toc64038106)

[Gambar 3. 42 Rancangan *Layout* Halaman Detail Buku/*Ebook* 52](#_Toc64038107)

[Gambar 4. 1 Tampilan *Composer* 53](#_Toc64041092)

[Gambar 4. 2 Tampilan XAMPP 54](#_Toc64041093)

[Gambar 4. 3 Perintah Untuk Membuat Projek Laravel 54](#_Toc64041094)

[Gambar 4. 4 Direktori Projek Laravel 55](#_Toc64041095)

[Gambar 4. 5 *File* .env 57](#_Toc64041096)

[Gambar 4. 6 Implementasi Tampilan Halaman *Landing Page* 58](#_Toc64041097)

[Gambar 4. 7 Implementasi Tampilan Halaman *Login* 59](#_Toc64041098)

[Gambar 4. 8 Implementasi Tampilan Halaman *Dashboard* *Admin*/Pustakawan 59](#_Toc64041099)

[Gambar 4. 9 Implementasi Tampilan Halaman Data Buku 60](#_Toc64041100)

[Gambar 4. 10 Implementasi Tampilan Halaman Data Anggota 61](#_Toc64041101)

[Gambar 4. 11 Implementasi Tampilan Halaman Data *Ebook* 62](#_Toc64041102)

[Gambar 4. 12 Implementasi Tampilan Halaman Data Penerbit 62](#_Toc64041103)

[Gambar 4. 13 Implementasi Tampilan Halaman Data Kategori 63](#_Toc64041104)

[Gambar 4. 14 Implementasi Tampilan Halaman Data Pengajuan *Ebook* 64](#_Toc64041105)

[Gambar 4. 15 Implementasi Tampilan Halaman Data Transaksi 64](#_Toc64041106)

[Gambar 4. 16 Implementasi Tampilan Halaman Data Laporan 65](#_Toc64041107)

[Gambar 4. 17 Implementasi Tampilan Halaman Data Pustakawan 65](#_Toc64041108)

[Gambar 4. 18 Implementasi Tampilan Halaman Cetak Laporan 66](#_Toc64041109)

[Gambar 4. 19 Implementasi Tampilan Halaman Cetak Kartu Anggota 67](#_Toc64041110)

[Gambar 4. 20 Implementasi Tampilan Halaman Konfigurasi 67](#_Toc64041111)

[Gambar 4. 21 Implementasi Tampilan Halaman *Dashboard* Anggota 68](#_Toc64041112)

[Gambar 4. 22 Implementasi Tampilan Halaman Data Buku Anggota 68](#_Toc64041113)

[Gambar 4. 23 Implementasi Tampilan Halaman Data *Ebook* Anggota 69](#_Toc64041114)

[Gambar 4. 24 Implementasi Tampilan Halaman Detail Buku/*Ebook* 69](#_Toc64041115)

[Gambar 4. 25 Diagram Relasi 75](#_Toc64041116)

# **DAFTAR TABEL**

[Tabel 2. 1 Simbol DFD De Marco 15](#_Toc64043856)

[Tabel 2. 2 Simbol-Simbol ERD 16](#_Toc64043857)

[Tabel 2. 3 Simbol-Simbol *Use Case Diagram* 17](#_Toc64043858)

[Tabel 2. 4 Simbol-Simbol *Flowmap* 18](#_Toc64043859)

[Tabel 3. 1 Analisa Sistem Pengguna 20](#_Toc64045479)

[Tabel 4. 1 Penjelasan Direktori Projek Laravel 56](#_Toc64048936)

[Tabel 4. 2 Struktur Tabel *books* 71](#_Toc64048937)

[Tabel 4. 3 Struktur Tabel *categories* 71](#_Toc64048938)

[Tabel 4. 4 Struktur Tabel *publishers* 72](#_Toc64048939)

[Tabel 4. 5 Struktur Tabel *members* 72](#_Toc64048940)

[Tabel 4. 6 Struktur Tabel *ebooks* 72](#_Toc64048941)

[Tabel 4. 7 Struktur Tabel *transactions* 73](#_Toc64048942)

[Tabel 4. 8 Struktur Tabel *detail* *transactions* 73](#_Toc64048943)

[Tabel 4. 9 Struktur Tabel *permissions* 74](#_Toc64048944)

[Tabel 4. 10 Struktur Tabel *users* 74](#_Toc64048945)

[Tabel 4. 11 Struktur Tabel *librarians* 75](#_Toc64048946)

[Tabel 4. 12 Struktur Tabel *configs* 75](#_Toc64048947)

[Tabel 4. 13 Struktur Tabel *gallery* *configs* 76](#_Toc64048948)

[Tabel 4. 14 Tabel Pengujian Proses *Login* 77](#_Toc64048949)

[Tabel 4. 15 Tabel Pengujian Proses Konfirmasi Akun 78](#_Toc64048950)

[Tabel 4. 16 Tabel Pengujian Proses Pendataan Buku 80](#_Toc64048951)

[Tabel 4. 17 Tabel Pengujian Proses Pendataan Anggota 81](#_Toc64048952)

[Tabel 4. 18 Tabel Pengujian Proses Pendataan *Ebook* 82](#_Toc64048953)

[Tabel 4. 19 Tabel Pengujian Proses Pendataan Pustakawan/*Admin* 84](#_Toc64048954)

[Tabel 4. 20 Tabel Pengujian Proses Pendataan Penerbit 85](#_Toc64048955)

[Tabel 4. 21 Tabel Pengujian Proses Pendataan Kategori 86](#_Toc64048956)

[Tabel 4. 22 Tabel Pengujian Pendataan Pengajuan Penggunaan *Ebook* 87](#_Toc64048957)

[Tabel 4. 23 Tabel Pengujian Proses Pendataan Transaksi 87](#_Toc64048958)

[Tabel 4. 24 Tabel Pengujian Proses Pembuatan Laporan 89](#_Toc64048959)

[Tabel 4. 25 Tabel Pengujian Proses Konfigurasi 90](#_Toc64048960)

[Tabel 4. 26 Tabel Pengujian Proses Pengajuan Ebook 91](#_Toc64048961)

[Tabel 4. 27 Tabel Pengujian Keamanan 91](#_Toc64048962)

[Tabel 4. 28 Tabel Pengujian Proses Mengganti Password 92](#_Toc64048963)

# **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. *Web aplikasi* ini dibuat dengan menggunakan framework Laravel dan Bootstrap.
4. Versi Laravel yang digunakan yaitu Laravel 8.0, sedangkan untuk Bootstrap yaitu Bootstrap 5.
5. Penginputan data masih dilakukan secara manual.
6. 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**

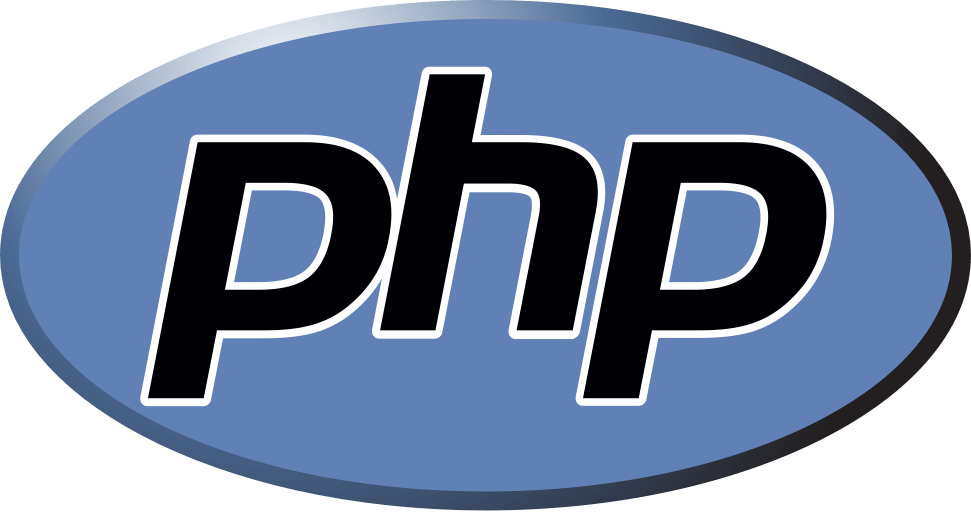
Jquery adalah *library* Javascript yang paling populer, yang pertama kali dibuat pada tahun 2006. Jquery bukanlah bahasa yang berdiri sendiri, melainkan bekerja sama dengan Javascript. Jquery dibuat dengan tujuan untuk mempercepat *developer* dalam membuat *website*. Dengan Jquery, *developer* pemula 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**

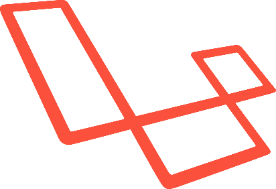
MVC (*Model* *View* *Controller*) merupakan suatu konsep yang digunakan dengan cara memisahkan data, tampilan dan pemrosesan kedua hal tersebut. Semua hal yang berurusan dengan data dari *database*, akan ditangani oleh *Model*, ini merupakan bagian terpenting karena semua data yang tampil bersumber dari 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.

Tabel 2. 1 Simbol DFD De Marco

|  |  |
| --- | --- |
| **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.

Tabel 2. 2 Simbol-Simbol ERD

|  |  |
| --- | --- |
| **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. Berbeda dengan diagram yang lainnya, simbol pada *diagram use case* lebih sedikit dan lebih sederhana.

Tabel 2. 3 Simbol-Simbol Use Case Diagram

|  |  |
| --- | --- |
| **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.

Tabel 2. 4 Simbol-Simbol Flowmap

|  |  |
| --- | --- |
| **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*. |
|  | 1. Simbol ini merupakan simbol yang dipakai untuk menyambungkan proses yang masih ada di halaman yang sama. |
|  | 1. Simbol ini merupakan simbol yang dipakai untuk menyambungkan proses yang berada di halaman yang berbeda. |
|  | 1. Simbol ini merupakan simbol yang dipakai ketika ada pemakaian *internal* memori, seperti penggunaan *array*. |

# **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 Pengguna

|  |  |
| --- | --- |
| **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 :* 500GB
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. *Text* *Editor* : Visual Studio Code (VSC)
4. Desain Grafis : Adobe Photoshop CS6 dan Adobe Illustrator CC
5. Paket Aplikasi *Server* : XAMPP
6. *Software* pendukung: *Composer*

## **Perancangan Sistem**

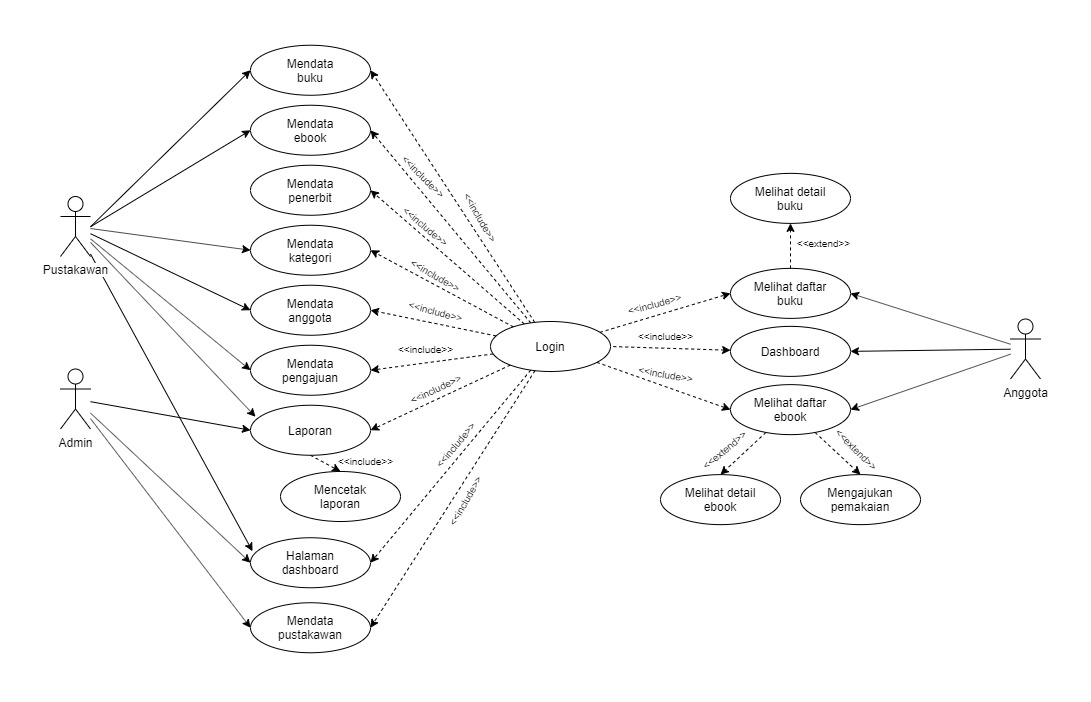
Berikut merupakan analisis dari aplikasi yang dibuat dengan menggunakan diagram-diagram, meliputi *use case diagram*, *data flow diagram* (DFD), *entity relationship diagram* (ERD) dan *flowmap*.

## ***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.

Terdapat 3 aktor/*user* yang terlibat langsung dengan aplikasi O’Library, yaitu *admin*, pustakawan dan *member*. Ketiga *user* tersebut memiliki peran dan haknya masing-masing. Pustakawan merupakan *user* dengan hak akses paling luas, mengakses halaman pendataan buku, *ebook*, penerbit, kategori, pengajuan *ebook*, laporan serta peminjaman/pengembalian. Sedangkan untuk admin, hak akses nya lebih dibatasi dibanding dengan pustakawan, hanya dapat mengakses halaman pendataan pustakawan dan laporan saja. Untuk *member*, diberikan hak akses untuk dapat melihat daftar buku dan *ebook* yang tersedia di perpustakaan, serta dapat mengajukan permintaan penggunaan *ebook* kepada pustakawan.

Proses *login* merupakan proses terpenting yang harus dilakukan oleh ketiga *user* agar dapat mengelola semua proses yang dibutuhkan.



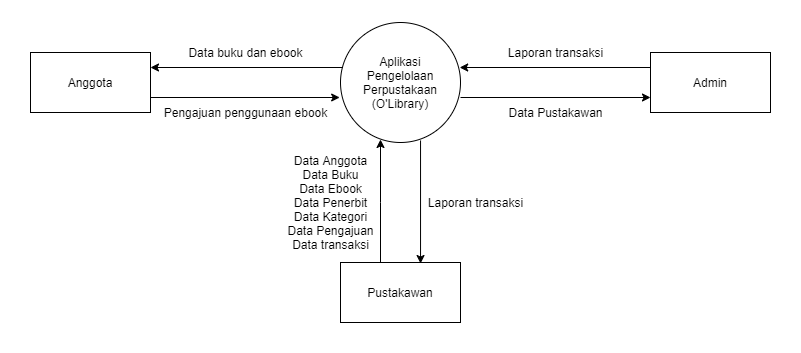
Gambar 3. 1 Use Case Diagram

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

Berikut DFD dari *web* aplikasi O’Library yang terbagi menjadi diagram kontes/*context* *diagram*, DFD *level* 1 dan DFD *level* 2.

* + - 1. *Context* *Diagram*

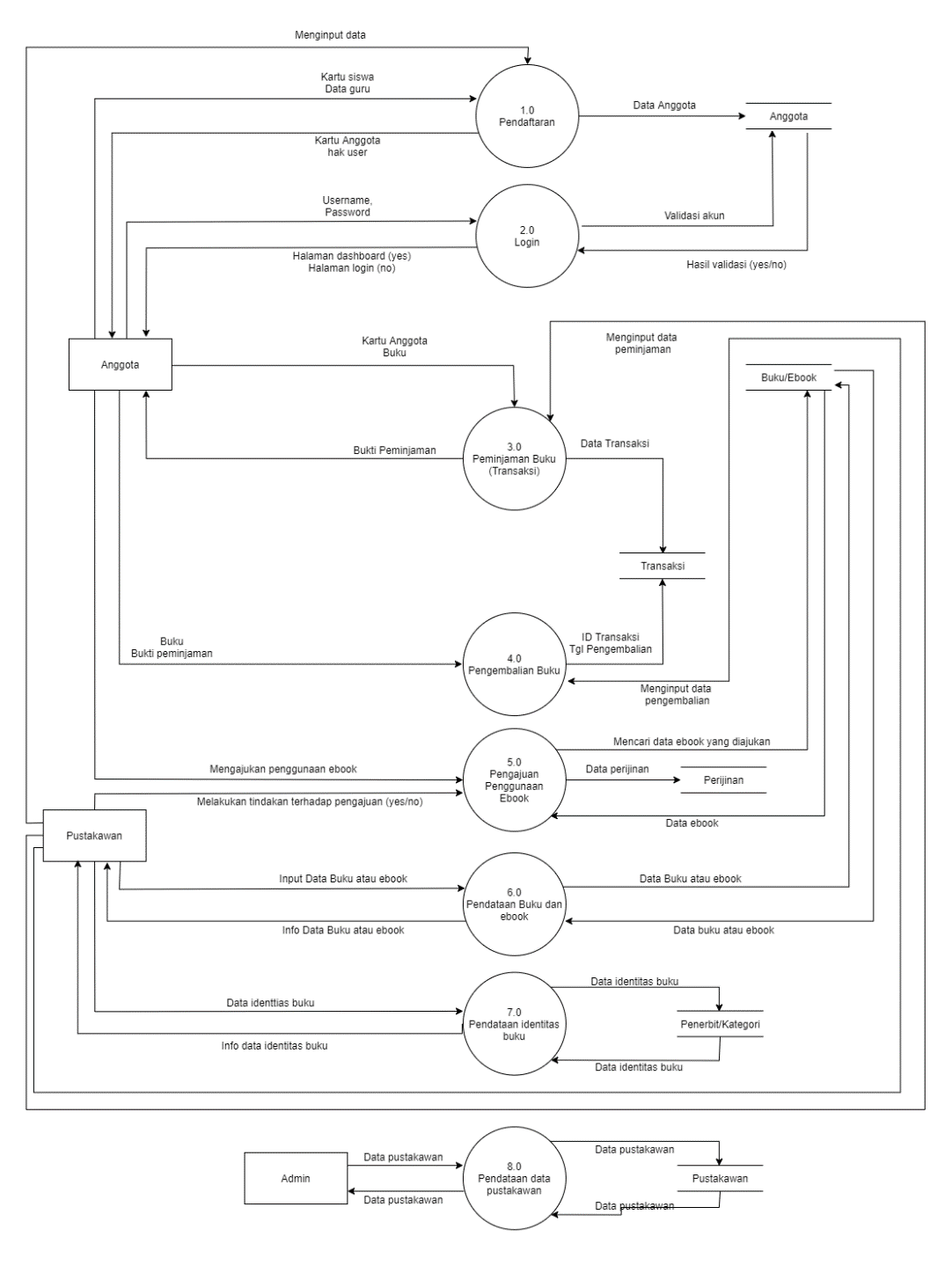
Berikut diagram konteks yang menjelaskan terdapat 3 entitas utama.



Gambar 3. 2 DFD Konteks Diagram

* + - 1. DFD *Level* 1

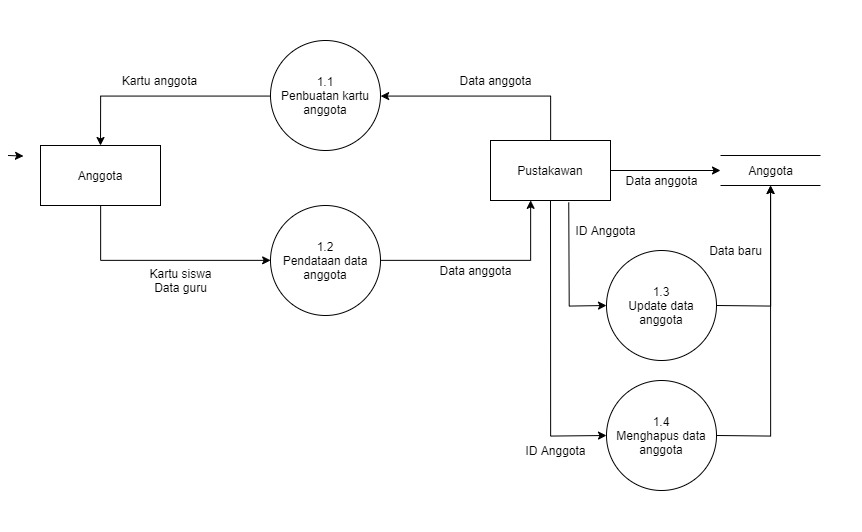
Berikut merupakan gambar DFD *level* 1 yang merupakan pengembangan dari diagram konteks. Dijabarkan bahwa 3 entitas utama terlibat ke dalam 8 proses utama yang terdapat dalam aplikasi. Keterlibatan *database* sangatlah erat dalam proses aplikasi, karena semua sumber informasi serta data didapat dari *database*.



Gambar 3. 3 DFD Level 1

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

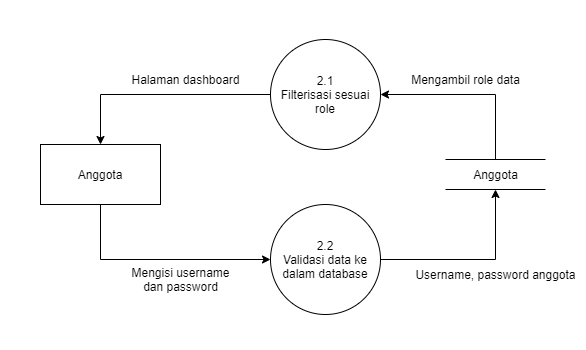
Di proses ini, *user* yang terlibat hanya pustakawan dan *member* saja, *member* datang kepada pustakawan untuk daftar menjadi anggota perpustakaan, kemudian pustakawan akan mendata informasi mengenai calon anggota yang akan daftar.



Gambar 3. 4 DFD Level 2, Proses 1.0 Pendaftaran

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

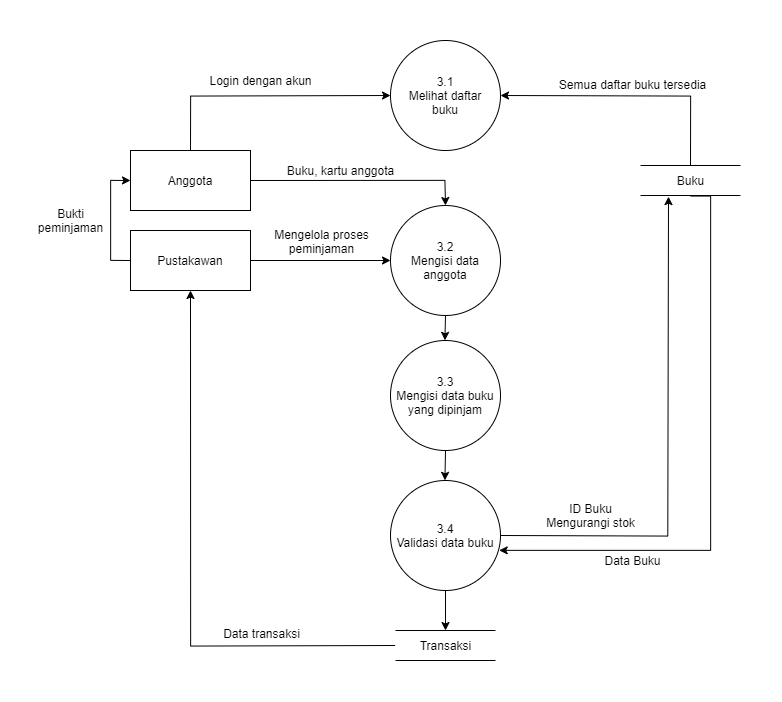
Berikut merupakan proses *login* yang terjadi di sistem.



Gambar 3. 5 DFD Level 2, Proses 2.0 Login

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

Untuk proses peminjaman, *member* datang ke perpustakaan dan mencari buku yang akan dipinjam. Kemudian pustakawan akan mendata, informasi mengenai buku apa yang akan dipinjam serta informasi anggota yang meminjam buku tersebut. Setelah selesai, pustakawan akan membuat bukti peminjaman berupa catatan kecil yang diselipkan di buku.

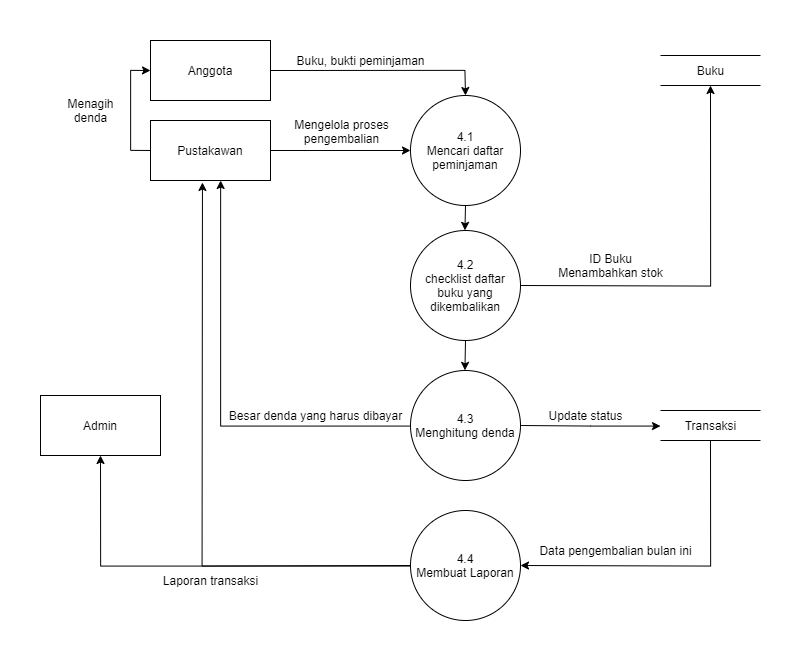


Gambar 3. 6 DFD Level 2, Proses 3.0 Peminjaman

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

Pada proses ini, mirip dengan proses peminjaman, *member* akan datang ke perpustakaan untuk mengembalikan buku, lalu pustakawan akan mendata informasi mengenai anggota dan buku apa yang dikembalikan. Lalu sistem akan menghitung denda jika anggota tersebut mengembalikan lebih dari batas wajar peminjaman.

Kemudian, data pengembalian tersebut akan otomatis masuk ke dalam halaman laporan, yang berisi informasi seperti nama anggota, buku yang dipinjam, tanggal peminjaman, tanggal pengembalian, keterangan tepat waktu atau tidak, serta denda yang ditanggung jika terlambat mengembalikan.

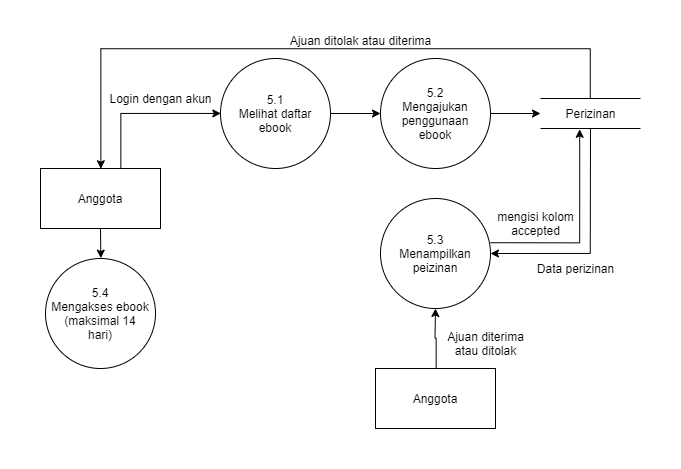


Gambar 3. 7 DFD Level 2, Proses 4.0 Pengembalian

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

Proses ini merupakan proses yang dilakukan oleh 2 *user*. Proses pengajuan *ebook* merupakan proses saat seorang anggota mengajukan permintaan untuk penggunaan *ebook* kepada pustakawan. Proses pengajuan ini dilakukan di sisi anggota. Kemudian, data pengajuan tersebut akan masuk dalam *database*, yang akan tampil di halaman pengajuan yang dapat diakses oleh pustakawan. Data pengajuan tersebut berisi informasi anggota yang mengajukan, *ebook* apa yang diajukan serta alasan mengapa anggota tersebut ingin menggunakan *ebook*.

Saat data pengajuan sudah masuk di *database* dan ditampilkan ke halaman pengajuan, seorang pustakawan berkewajiban untuk memutuskan apakah pengajuan tersebut layak diterima atau ditolak. Satu-satunya hal yang menyebabkan pengajuan tersebut ditolak yaitu alasan yang diberikan anggota dalam pengajuan tidak jelas atau tidak logis. Sehingga, dalam proses ini, seorang anggota harus memasukkan alasan yang meyakinkan dan logis, jika tidak, pengajuan tersebut akan ditolak oleh pustakawan.

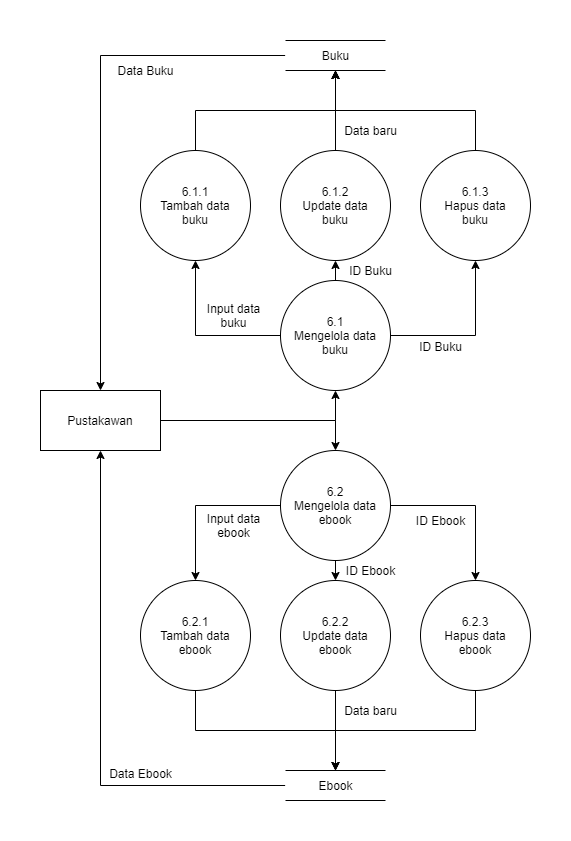


Gambar 3. 8 DFD Level 2, Proses 5.0 Pengajuan Ebook

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

Proses selanjutnya yaitu proses yang dilakukan oleh pustakawan yaitu proses pendataan buku atau *ebook*. Seorang pustakawan akan menginputkan data terkait buku/*ebook* baru yang menjadi koleksi baru perpustakaan, data tersebut seperti judul, penerbit, penulis, kategori, stok tersedia, *link* untuk *ebook* serta tahun terbit. Selain menambahkan, pustakawan juga dapat mengubah atau bahkan menghapus data buku/*ebook*.

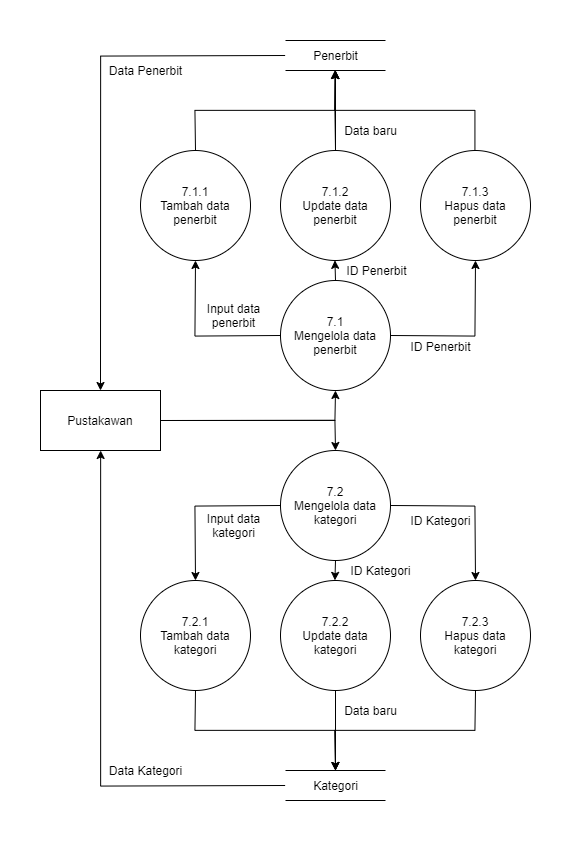
Proses ini merupakan murni proses yang hanya dapat dilakukan oleh pustakawan, karena ini merupakan hal yang berhubungan dengan data pembukuan koleksi perpustakaan.



Gambar 3. 9 DFD Level 2, Proses 6.0 Pendataan Buku/Ebook

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

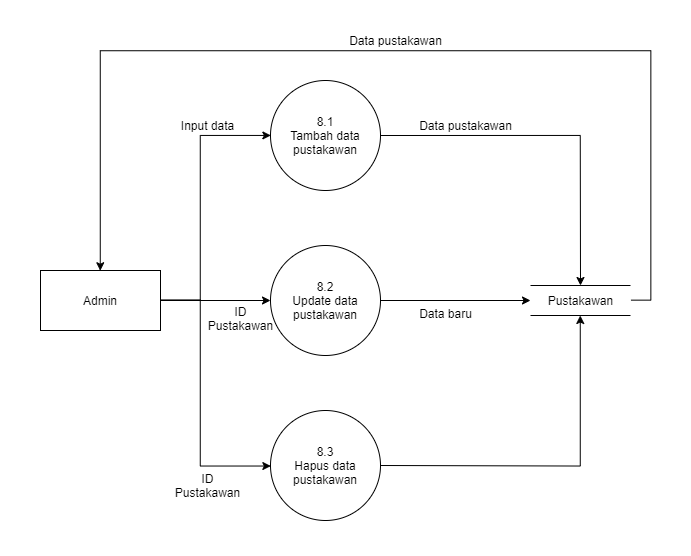
Mirip dengan proses pendataan buku/*ebook*, proses pendataan identitas merupakan proses yang berkaitan dengan penerbit dan kategori yang merupakan salah satu data penting saat pustakawan memasukkan data buku/*ebook* yang baru. Pustakawan dapat menghapus atau mengubah data ini, jika dirasa terdapat data yang tidak tepat. Layaknya dengan pendataan buku/*ebook*, proses ini merupakan proses yang hanya dapat diakses oleh pustakawan, karena hal ini termasuk pula dalam data pembukuan koleksi perpustakaan yang tak dapat diakses oleh *admin* atau *member*.



Gambar 3. 10 DFD Level 2, Proses 7.0 Pendataan Identitas Buku/Ebook

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

Proses ini merupakan proses yag hanya dapat dilakukan oleh *admin*. Seorang *admin* akan menambahkan data staff perpustakaan baru yang prosesnya mirip saat pustakawan menambahkan data anggota baru. *Admin* dapat mengubah datanya jika terdapat kesalahan dalam memasukkan data, atau bahkan menghapus datanya, jika terdapat orang yang sudah tidak lagi menjabat sebagai staff perpustakaan.

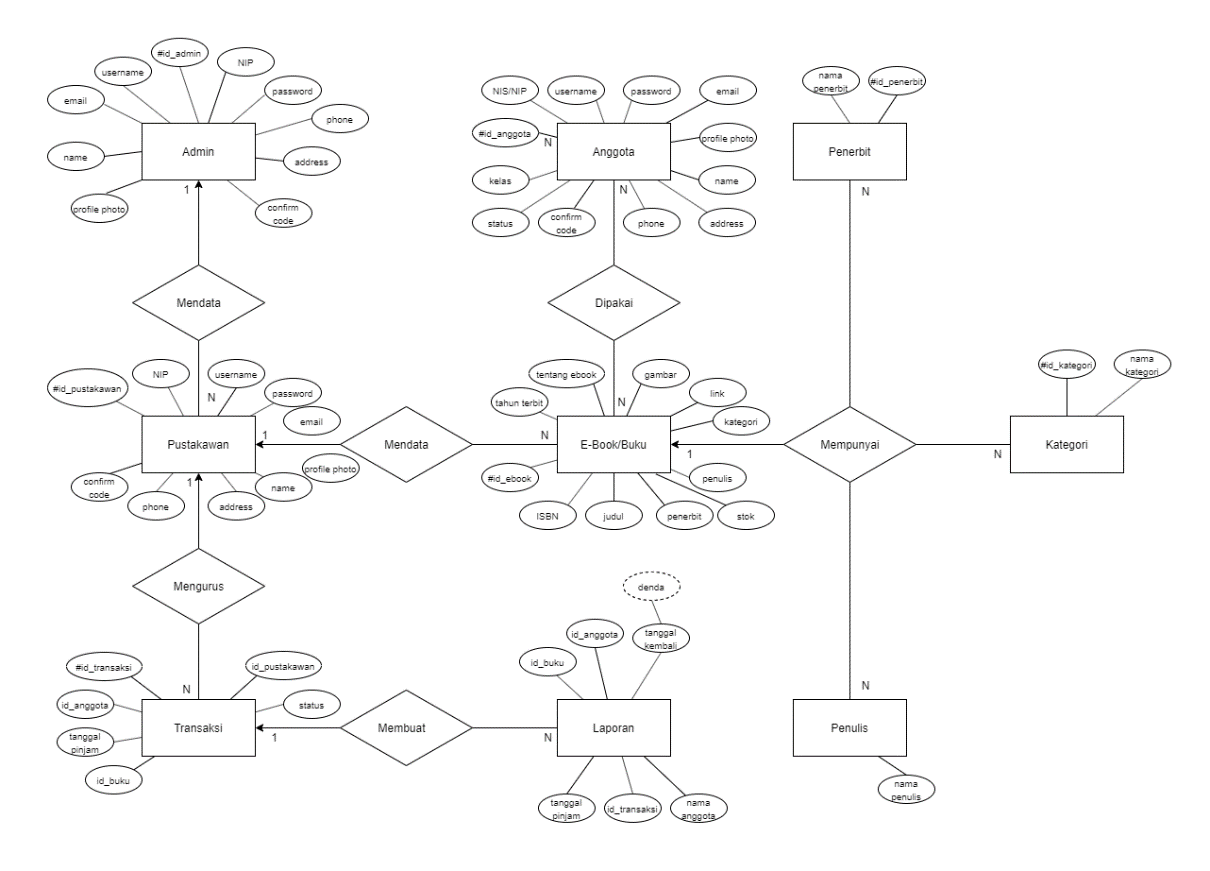


Gambar 3. 11 DFD Level 2, Proses 8.0 Pendataan Pustakawan

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

Berikut merupakan diagram relasi dari setiap entitas yang terlibat dalam aplikasi. Terdapat 9 entitas utama yang terlibat, dan memiliki atribut/kebutuhannya masing-masing. Juga memiliki hubungan dengan entitas lain. Menampilkan pula hubungan kardinalitas di antara entitas-entitas yang saling berhubungan, menunjukkan perbandingan yang terjadinya diantaranya, dapat berupa 1:1 (*One* *to* *one*), 1:N (*One to many*), N:1 (*Many to one*) atau N:N (*Many to many*).

ERD merupakan hal yang penting dalam sebuah perancangan aplikasi sebelum aplikasi tersebut dibuat, digunakan untuk menganalisis keterlibatan entitas dan hubungan di antara entitas-entitas tersebut. Juga membantu sebelum merancang struktur basis data yang dibutuhkan.

****

Gambar 3. 12 Entity Relationship Diagram (ERD)

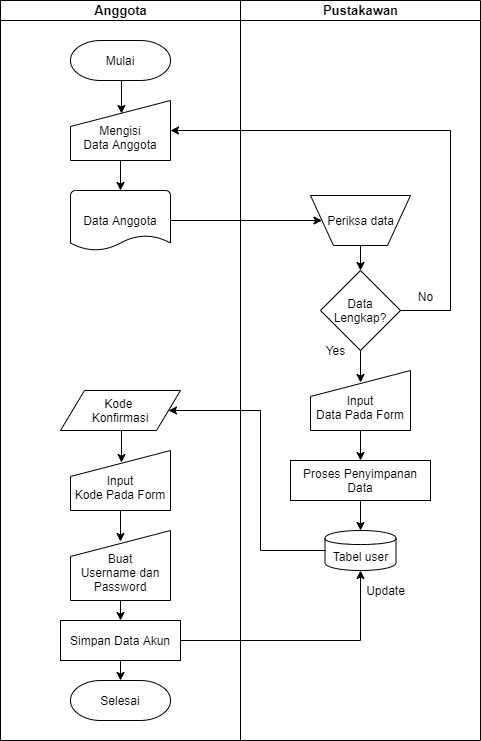
## ***Flowmap***

*Flowmap* Proses 1.0 Pendaftaran

Pada *flowmap* di bawah, dijelaskan bahwa proses pendaftaran dimulai saat calon anggota mengisi data-data yang dibutuhkan untuk mendaftar sebagai anggota perpustakaan. Kemudian, pustakawan akan mengecek secara manual data-data yang telah diisi calon anggota, apakah sudah lengkap atau belum. Jika dirasa sudah, pustakawan akan mengetikkan data-data tersebut dalam aplikasi yang nantinya akan disimpan ke dalam *database*.

Setelah data berhasil dimasukkan ke dalam *database*, sistem akan otomatis membuat kode konfirmasi yang akan langsung dikirim ke anggota via *email*. Kode konfirmasi merupakan kode angka 5 digit yang dipakai anggota untuk membuat akunnya sendiri, dengan mengisi *username* dan *password*.

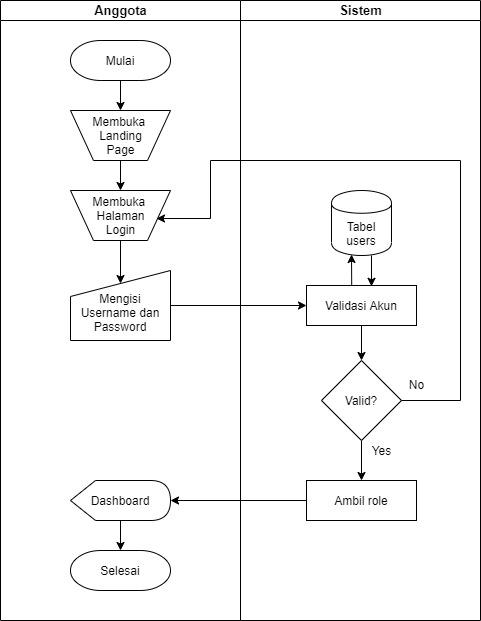
Setelah proses itu selesai, maka anggota sudah boleh masuk *web* perpustakaan dengan akunnya sendiri.



Gambar 3. 13 Flowmap Proses 1.0 Pendaftaran

*Flowmap* Proses 2.0 Login

Pada proses *login*, dimulai saat *user* mencoba untuk *login* dengan memasukkan *username* dan *password* yang sudah dibuat, lalu proses validasi akan dilakukan sendiri oleh sistem. Jika data yang dimasukkan tidak terdaftar di *database*, makan pengguna akan diarahkan kembali ke halaman *login*, sedangkan jika yang dimasukkan benar/*valid*, maka sistem akan otomatis melakukan *filtering* sesuai *role* yang dimiliki oleh anggota yang *login*, lalu akan diarahkan ke halaman *dashboard*.



Gambar 3. 14 Flowmap Proses 2.0 Login

*Flowmap* Proses 3.0 Peminjaman

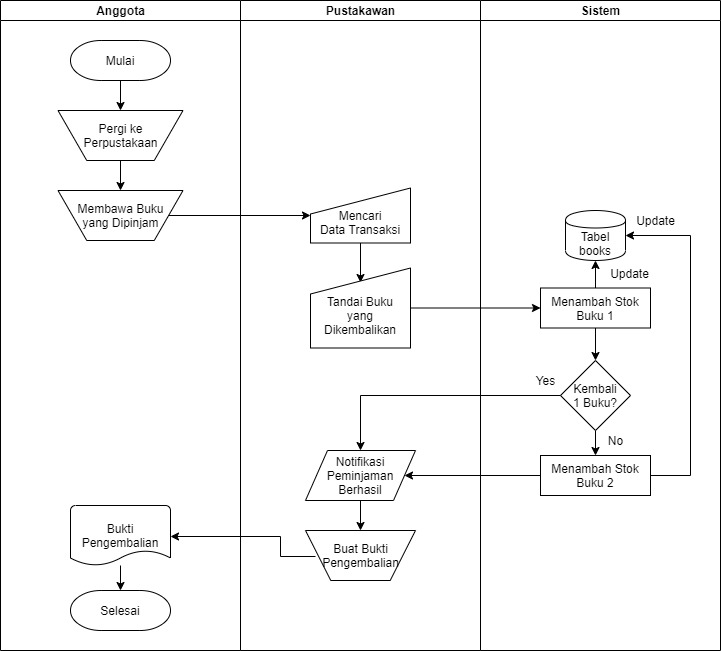
Proses peminjaman akan berlangsung saat anggota pertama kali mencari buku yang sedang dibutuhkan, lalu mendatangi meja pustakawan untuk didata. Pustakawan akan mendata mengenai anggota yang meminjam serta buku apa yang ingin dipinjam. Setelah itu sistem akan melakukan sendiri proses penyimpanan data ke dalam *database*. Setelah itu, pustakawan membuat bukti peminjaman yang umumnya berupa catatan kecil yang diselipkan di halaman belakang buku.



Gambar 3. 15 Flowmap Proses 3.0 Peminjaman

*Flowmap* Proses 4.0 Pengembalian

Layaknya proses peminjaman, proses pengembalian pun mirip seperti proses anggota akan meminjam buku. Anggota datang untuk mengembalikan buku, lalu pustakawan akan mencari data peminjaman terkait anggota yang meminjam serta buku apa saja yang akan dikembalikan. Data pengembalian akan otomatis terupdate yang dilakukan oleh sistem pada *database*, sekaligus menghitung besar denda yang harus dibayar jika anggota tersebut mengembalikan buku lebih dari batas wajar peminjaman.

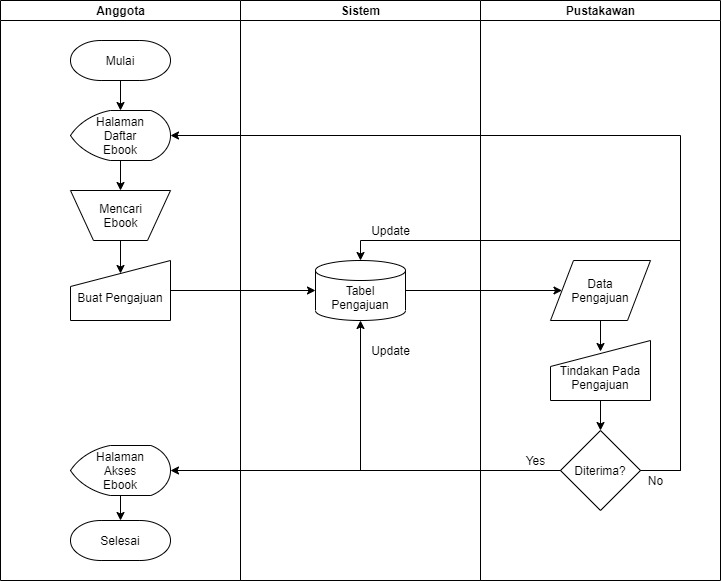


Gambar 3. 16 Flowmap Proses 4.0 Pengembalian

*Flowmap* Proses 5.0 Pengajuan

Proses ini merupakan proses yag dilakukan anggota untuk mendapatkan perijinan pustakawan dalam hal penggunaan *ebook*. Anggota akan mengisi *form* yang berisikan alasan untuk penggunaan *ebook*, data *form* tersebut akan masuk ke dalam *database* dan ditampilkan di halaman pustakawan untuk ditindak lanjuti. Pustakawan akan membaca satu persatu alasan yang diajukan dalam setiap permintaan penggunaan *ebook*, jika alasan yang diberikan dirasa kurang meyakinkan, tidak logis dan tidak jelas, makan pustakawan berhak untuk menolak pengajuan tersebut dengan memberikan alasan penolakan.

Namun, jika alasan yang diberikan masuk akal dan jelas, maka pustakawan akan menerima pengajuan tersebut, sehingga anggota dapat menggunakan *ebook* yang diajukan sesuai dengan batas waktu yang telah ditentukan. Saat batas penggunaan telah melebihi batas waktu, maka anggota sudah tidak bisa lagi menggunakan *ebook* tersebut, dan jika masih membutuhkan *ebook* tersebut, anggota harus mengisi kembali *form* yang berisikan alasan pengajuan.



Gambar 3. 17 Flowmap Proses 5.0 Pengajuan

*Flowmap* Proses 6.0 Pendataan Buku

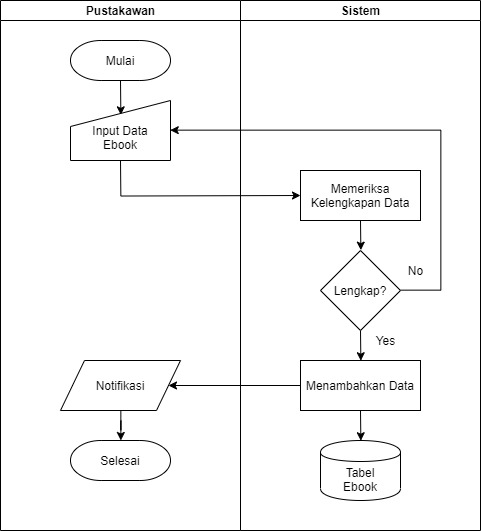
Setiap adanya koleksi buku baru di perpustakaan, sebelum disimpan di rak, buku-buku tersebut akan didata terlebih dahulu. Pustakawan melakukan proses ini dengan memasukkan data-data penting yang dibutuhkan seperti judul buku penerbit, kategori, penulis, stok tersedia, tahun terbit dan gambar dari *cover*/sampul buku agar mudah untuk dikenali. Data-data tersebut dimasukkan ke dalam suatu *form* khusus yang nantinya akan disimpan ke dalam database. Data-data buku pula akan ditampilkan di sisi anggota, menampilkan buku-buku yang tersedia di perpustakaan. Jadi, sebelum meminjam, anggota dapat memastikan terlebih dahulu, apakah buku yang akan dipinjam masih tersedia atau tidak.



Gambar 3. 18 Flowmap Proses 6.0 Pendataan Buku

*Flowmap* Proses 7.0 Pendataan *Ebook*

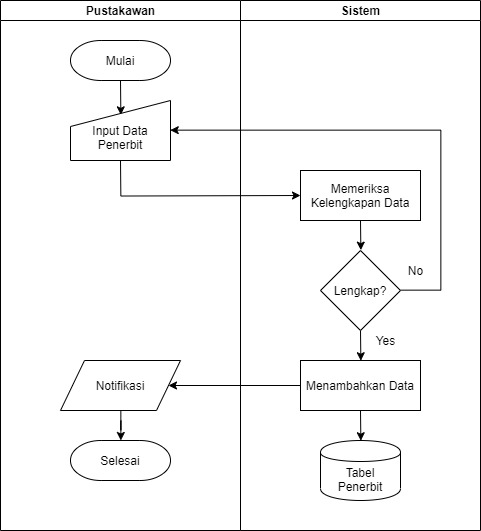
Sama halnya dengan buku, setiap adanya koleksi *ebook* baru di perpustakaan, *ebook*-*ebook* tersebut akan didata terlebih dahulu. Pustakawan melakukan proses ini dengan memasukkan data-data penting yang dibutuhkan seperti judul buku penerbit, kategori, penulis, *link* *ebook*, tahun terbit dan gambar dari *cover*/sampul *ebook* agar mudah untuk dikenali. Data-data tersebut dimasukkan ke dalam suatu *form* khusus yang nantinya akan disimpan ke dalam *database*. Data-data *ebook* ini akan ditampilkan di sisi anggota, menampilkan daftar *ebook* yang tersedia di perpustakaan. Namun, untuk menggunakannya, anggota wajib untuk mengajukan penggunaan *ebook* kepada pustakawan.



Gambar 3. 19 Flowmap Proses 7.0 Pendataan Ebook

*Flowmap* Proses 8.0 Pendataan Penerbit

Penerbit merupakan salah satu data penting yang dimasukkan jika terdapat koleksi buku/*ebook* baru di perpustakaan. Pustakawan akan mengelola data penerbit, seperti menambah, mengubah atau menghapus data. Data-data yang dibutuhkan disini hanyalah nama penerbit, dikelola dalam suatu *form* khusus yang terdpaat di halaman pustakawan.



Gambar 3. 20 Flowmap Proses 8.0 Pendataan Penerbit

*Flowmap* Proses 9.0 Pendataan Kategori

Penerbit merupakan salah satu data penting yang dimasukkan jika terdapat koleksi buku/*ebook* baru di perpustakaan. Pustakawan akan mengelola data penerbit, seperti menambah, mengubah atau menghapus data. Data-data yang dibutuhkan disini hanyalah nama penerbit, dikelola dalam suatu *form* khusus yang terdpaat di halaman pustakawan.

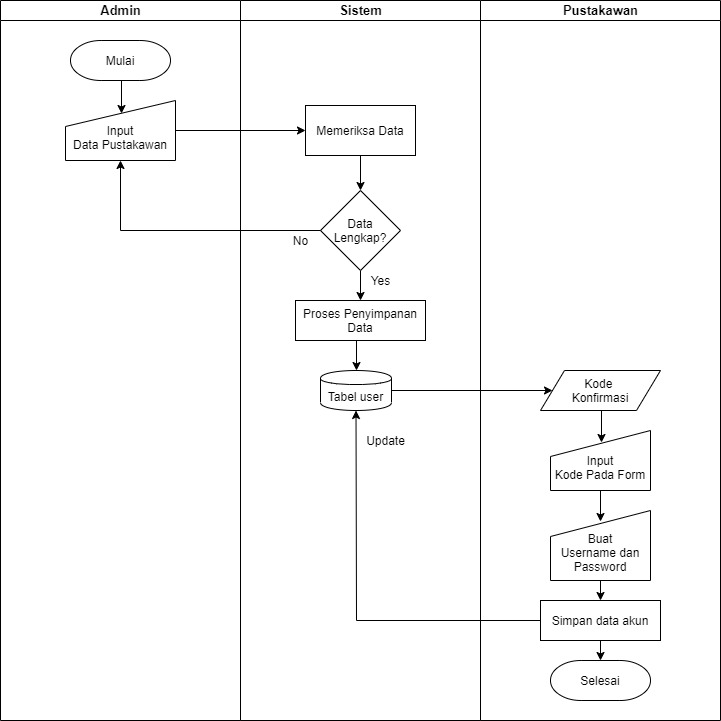


Gambar 3. 21 Flowmap Proses 9.0 Pendataan Kategori

*Flowmap* Proses 10.0 Pendataan Pustakawan

Mengelola data pustakawan mirip dengan pendataan pada anggota, tetapi data pustakawan hanya dapat dikelola oleh *admin*. *Admin* akan memasukkan data-data yang dibutuhkan untuk calon pustakawan yang baru, lalu calon pustakawan akan mendapat *email* yang berisi kode konfirmasi. Kode konfirmasi tersebut digunakan untuk membuat *username* dan *password* sendiri. Setelah itu, pustakawan sudah bisa *login* dan menjalankan tugasnya.

*Admin* juga memiliki hak untuk menghapus data pustakawan jika pustakawan terkait sudah tidak lagi menjabat sebagai staff perpustakaan. Dengan itu, data-data tak terpakai tidak akan menumpuk di halaman daftar pustakawan agar *admin* sewaktu-waktu dapat mudah mencari pustakawan jika dibutuhkan.



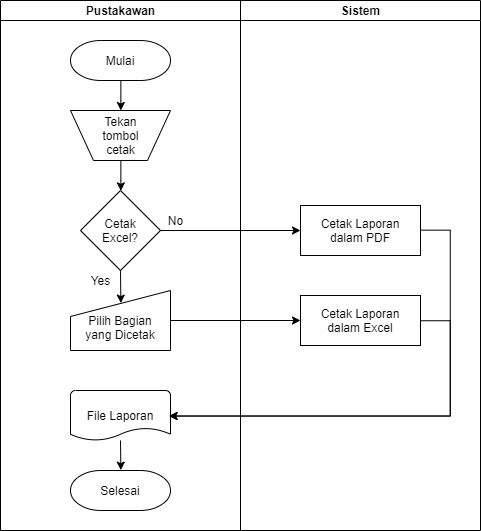
Gambar 3. 22 Flowmap Proses 10.0 Pendataan Pustakawan

*Flowmap* Proses 11.0 Laporan

Dalam susunan *database*, laporan tidak berdiri sendiri alias tidak memiliki tabel khusus. Data laporan berasal dari data transaksi, saat anggota sudah mengembalikan buku yang dipinjam. Terdapat 2 fitur yang sangat membantu di halaman laporan, yaitu eksport data laporan menjadi *file* excel atau *file* pdf.

*File* excel berisi data-data lengkap yang berasal dari *database* langsung, data-data tersebut dapat berupa data anggota, *admin*/pustakawan, buku, *ebook*, penerbit, kategori atau transaksi. Ini akan sangat memudahkan untuk mengawasi dan mengontrol data-data perpustakaan, terutama untuk jabatan tinggi yang berwenang untuk melakukan itu seperti kepala sekolah.

Sementara untuk *file* pdf, berbeda dengan excel, disini hanya berisi rangkuman beberapa proses yang terjadi dalam suatu periode. Misal, banyaknya peminjaman yang terjadi di bulan tertentu atau mencari data keterlambatan anggota dalam mengembalikan buku.



Gambar 3. 23 Flowmap Proses 11.0 Laporan

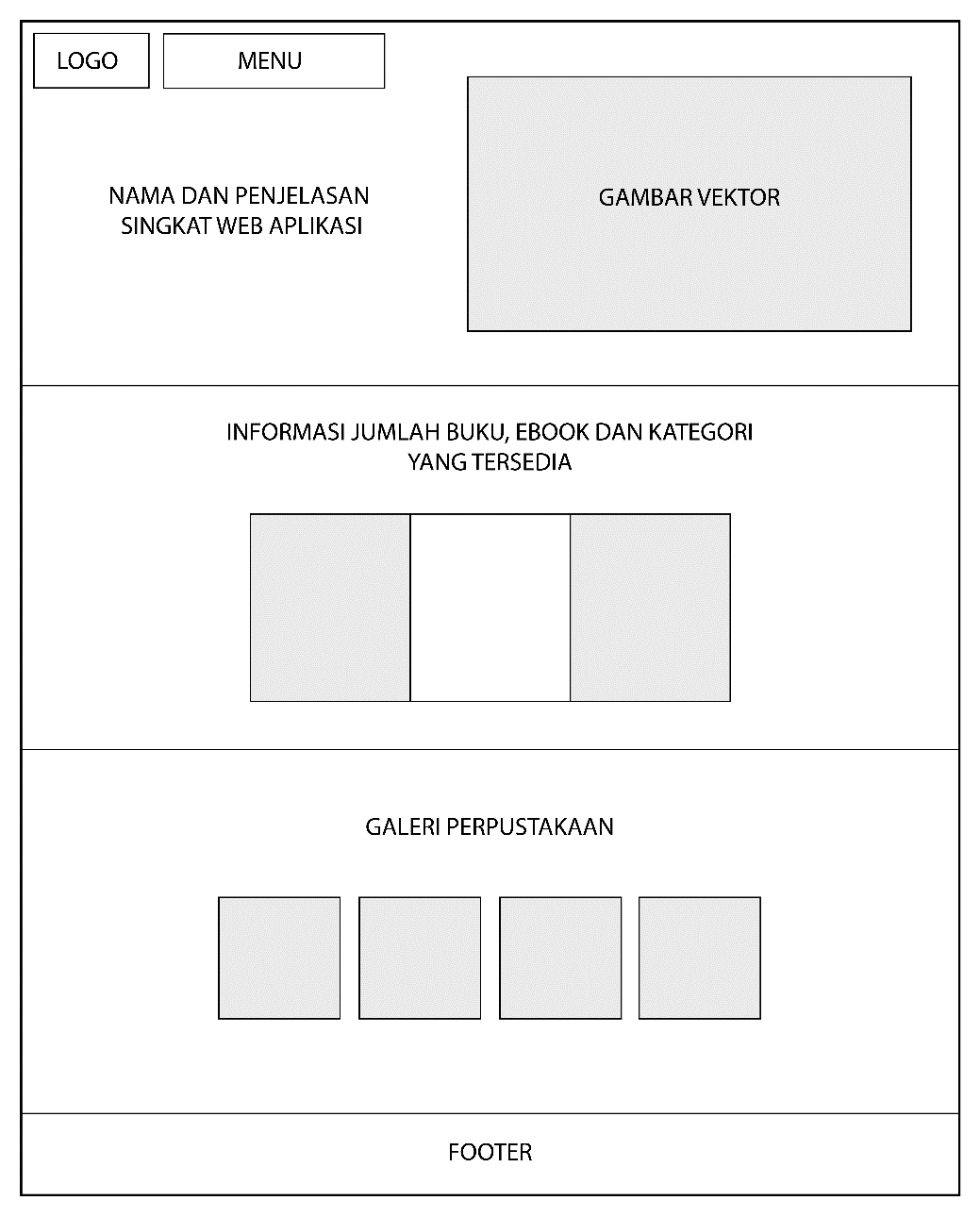
## **Perancangan *Layout***

*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.

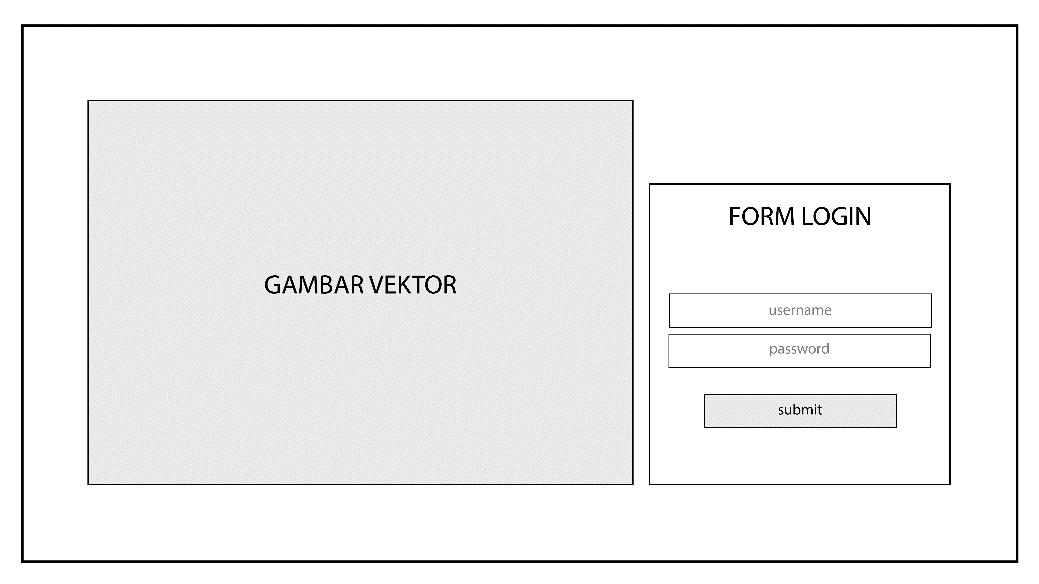
Berikut rancangan tampilan *landing* *page*. Terdiri atas, 3 bagian utama, yaitu penjelasan singkat, informasi perpustakaan dan galeri.



Gambar 3. 24 Rancangan Layout Landing Page

## **Perancangan *Layout* Halaman *Login***

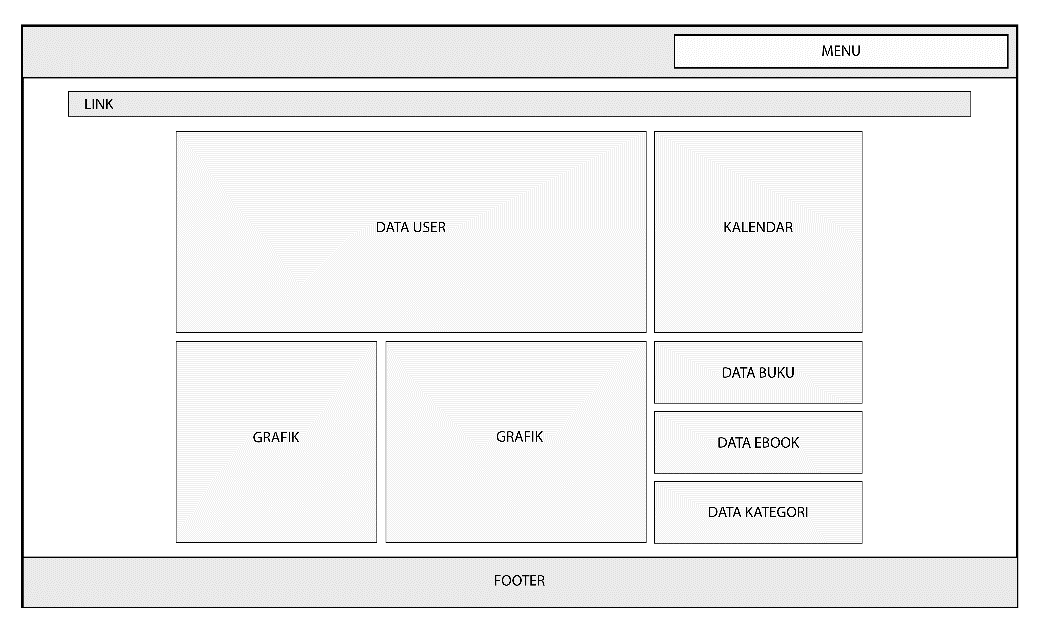
Berikut rancangan tampilan halaman *login*. Hanya menampilkan *form* *input* untuk diisi *username* dan *password* agar pengguna dapat masuk ke aplikasi.



Gambar 3. 25 Rancangan Layout Halaman Login

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

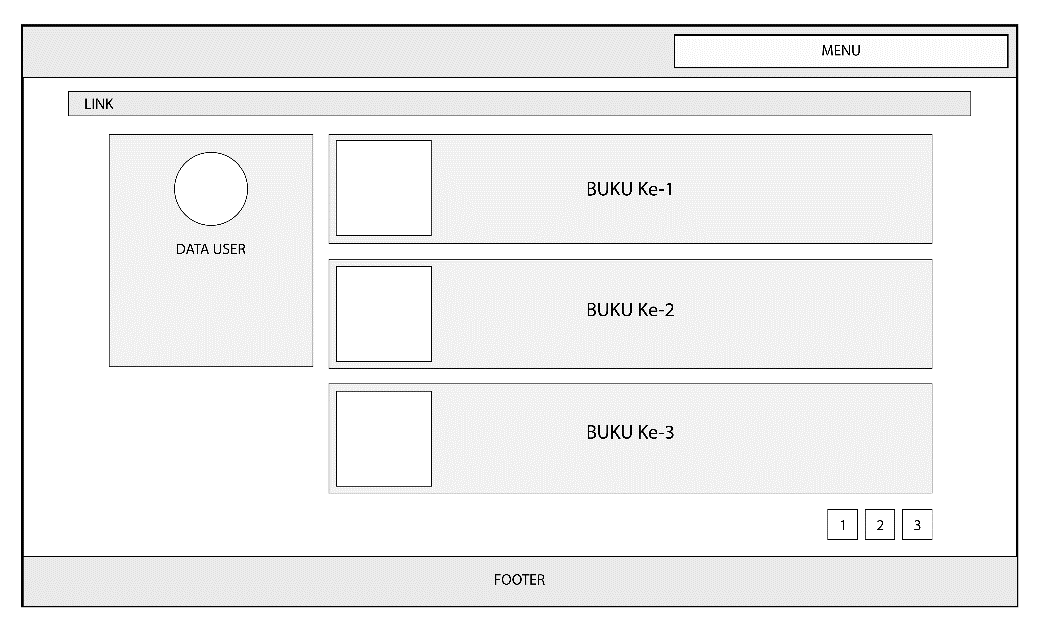
Berikut rancangan tampilan halaman *dashboard* untuk admin dan pustakawan. Mengandung informasi penting terkait perpustakaan yang ditampilkan dalam bentuk grafik.



Gambar 3. 26 Rancangan Layout Halaman Dashboard Admin/Pustakawan

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

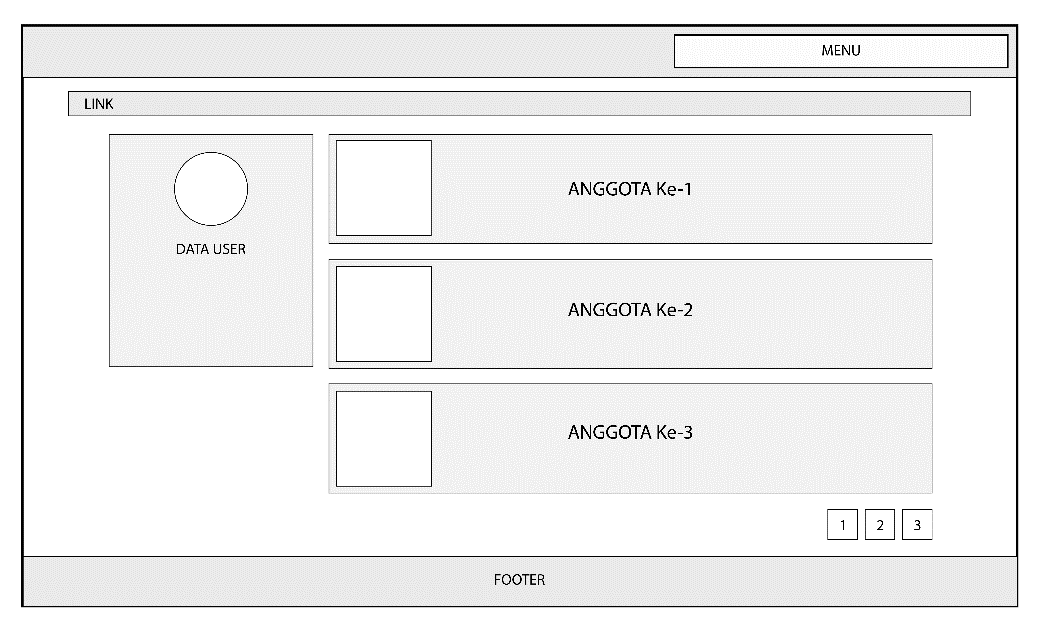
Berikut rancangan tampilan halaman data buku. Berisi daftar-daftar buku yang ditampilkan menggunakan *card* agar mudah untuk dibaca.



Gambar 3. 27 Rancangan Layout Halaman Data Buku

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

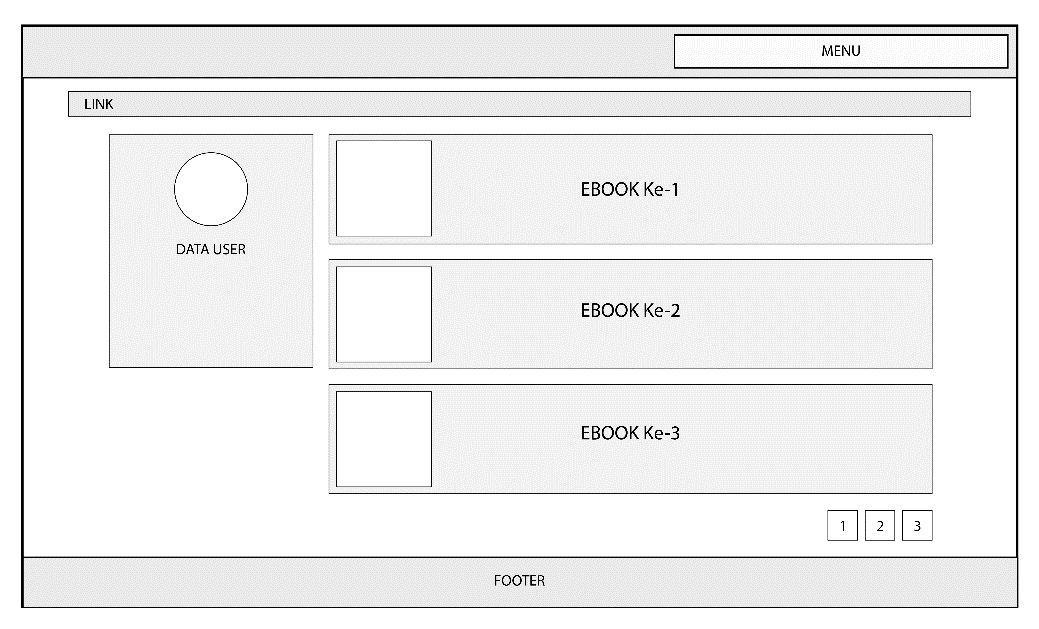
Berikut rancangan tampilan halaman data anggota. Mirip dengan buku, daftar anggota ditampilkan dengan menggunakan *card* agar mudah untuk dibaca.



Gambar 3. 28 Rancangan Layout Halaman Data Anggota

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

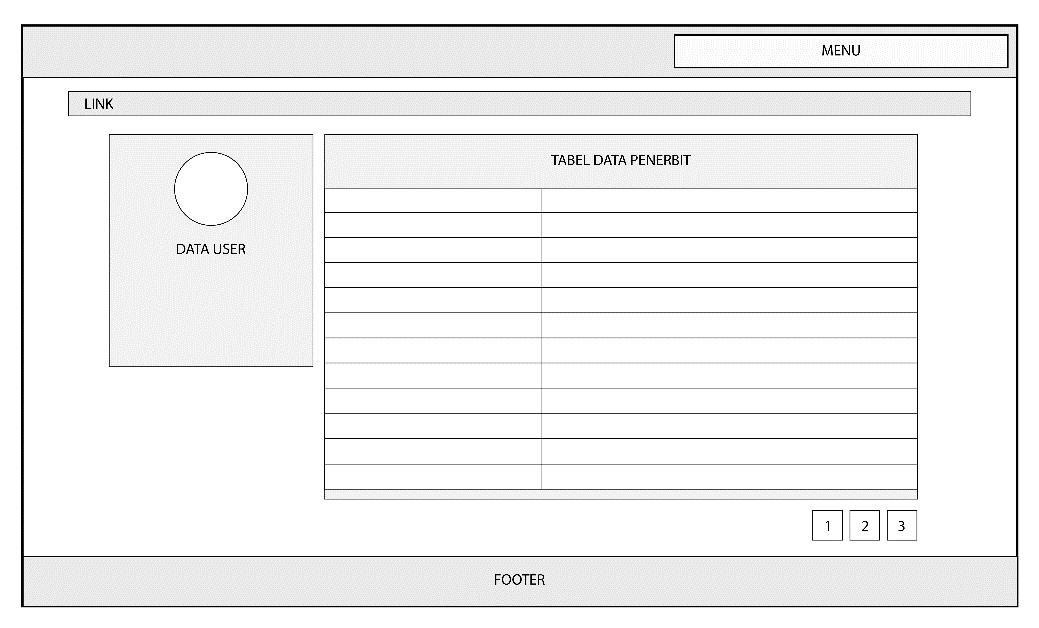
Berikut rancangan tampilan halaman data *ebook*. Mirip dengan buku, daftar *ebook* ditampilkan dengan menggunakan *card* agar mudah untuk dibaca.



Gambar 3. 29 Rancangan Layout Halaman Data Ebook

## **Perancangan *Layout* Halaman Data Penerbit**

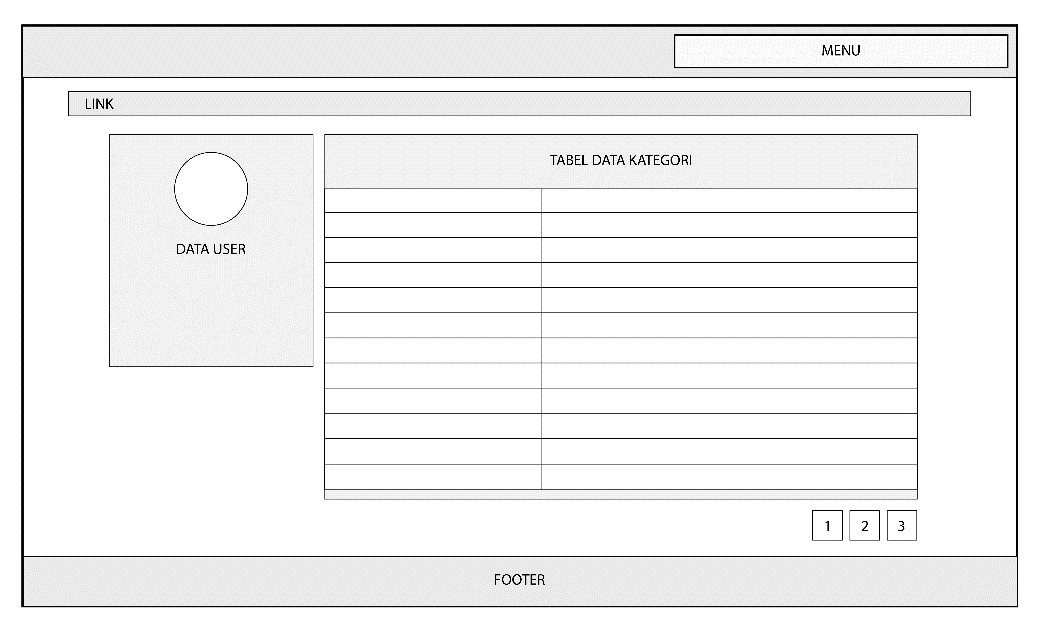
Berikut rancangan tampilan halaman data penerbit. Data penerbit ditampilkan dengan menggunakan tabel sederhana yang terdiri atas 2 kolom.



Gambar 3. 30 Rancangan Layout Halaman Data Penerbit

## **Perancangan *Layout* Halaman Data Kategori**

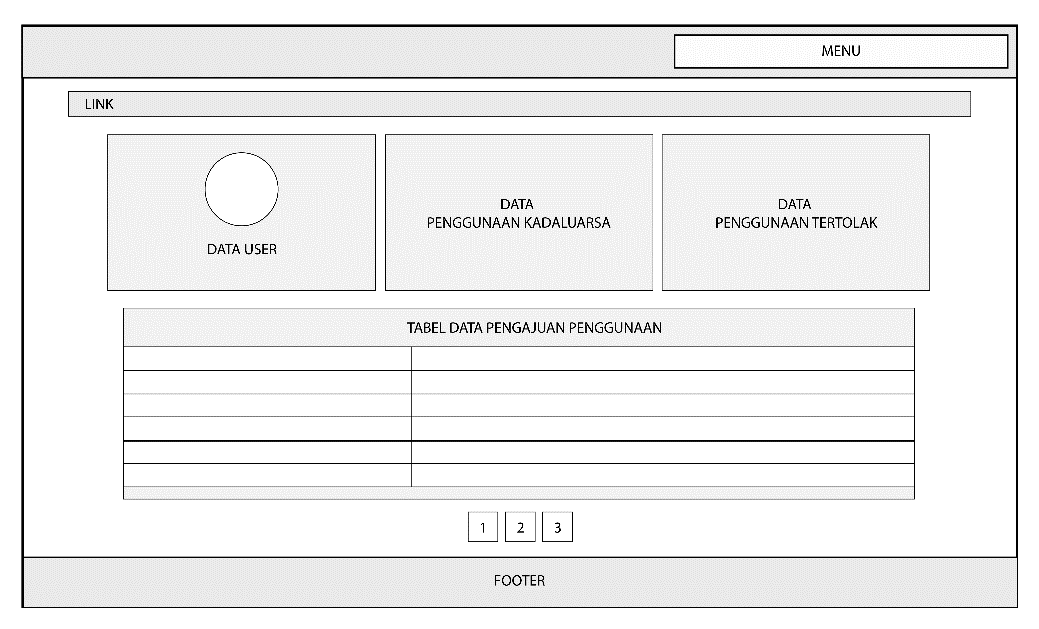
Berikut rancangan tampilan halaman data kategori. Data kategori ditampilkan dengan menggunakan tabel sederhana yang terdiri atas 2 kolom.



Gambar 3. 31 Rancangan Layout Halaman Data Kategori

## **Perancangan *Layout* Halaman Data Pengajuan**

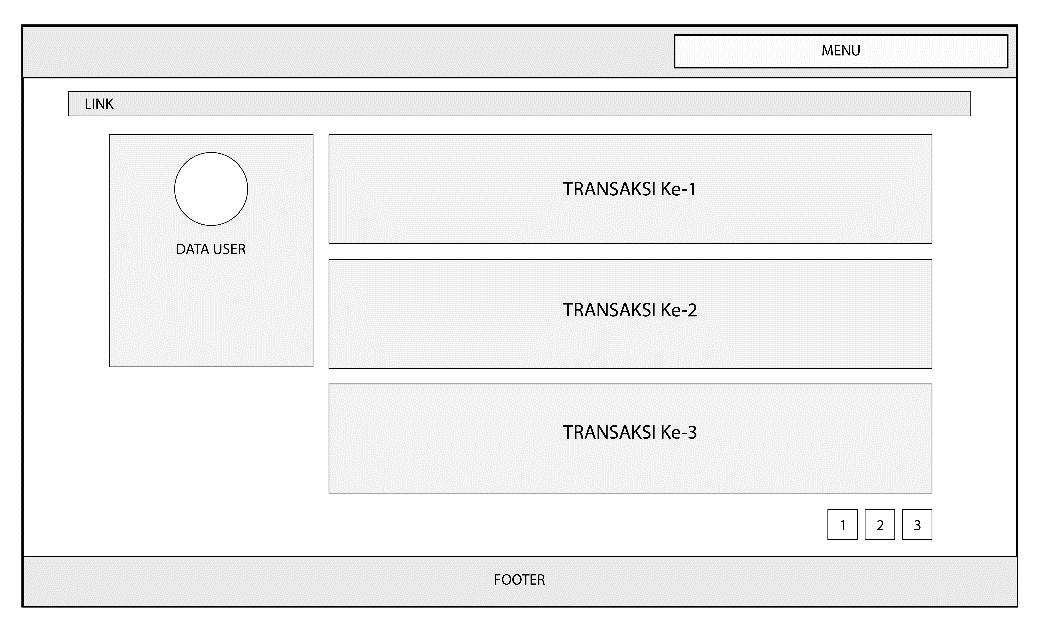
Berikut rancangan tampilan halaman data pengajuan.



Gambar 3. 32 Rancangan Layout Halaman Data Pengajuan

## **Perancangan *Layout* Halaman Data Transaksi**

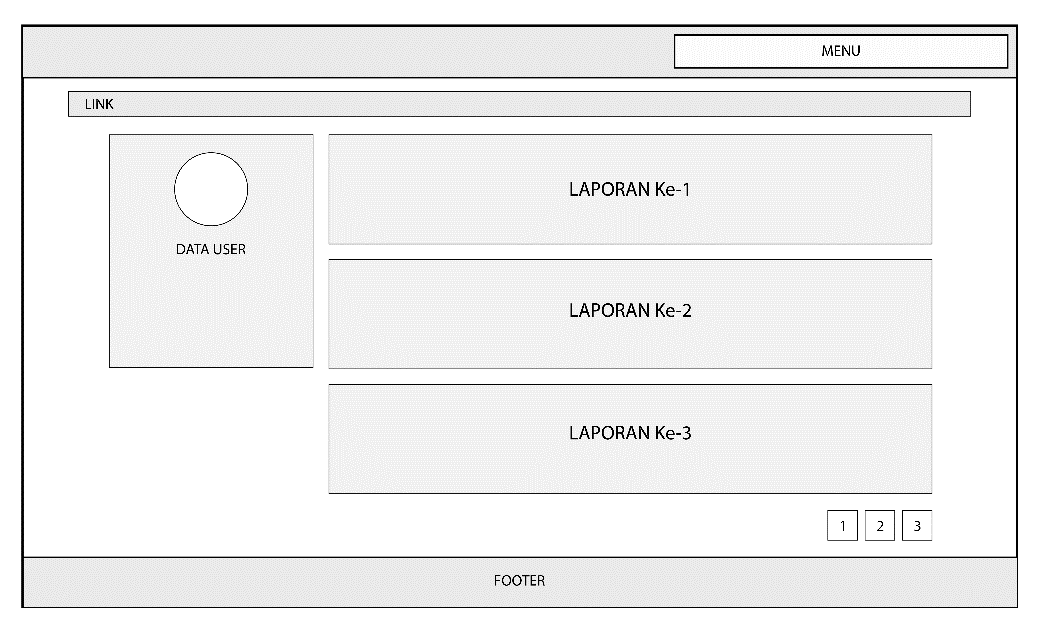
Berikut rancangan tampilan halaman data transaksi. Ditampilkan dengan menggunakna *card* agar mudah untuk dibaca.



Gambar 3. 33 Rancangan Layout Halaman Data Transaksi

## **Perancangan *Layout* Halaman Data Laporan**

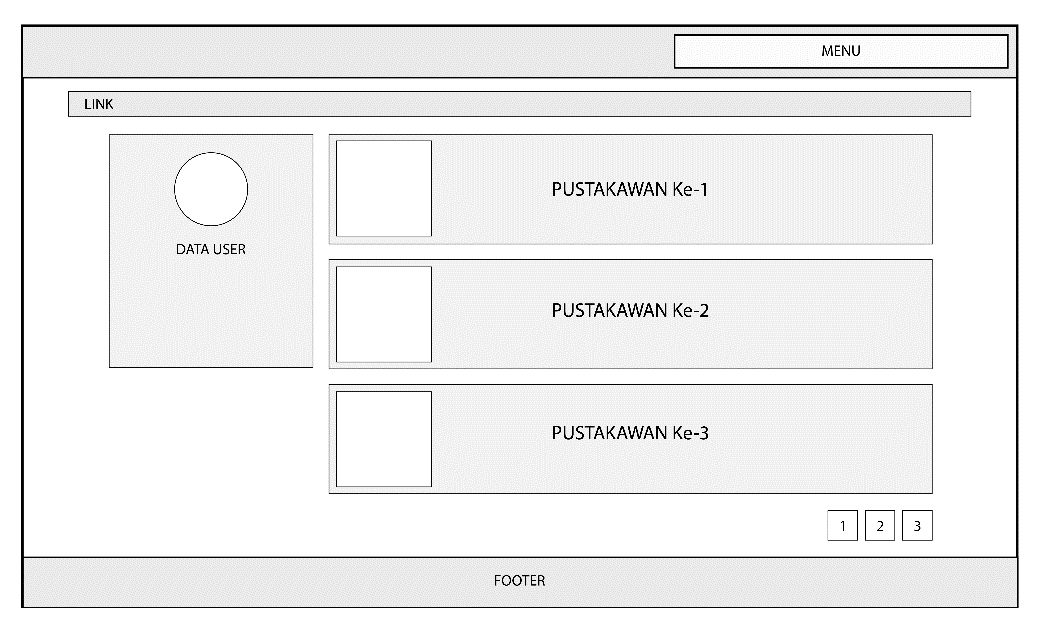
Berikut rancangan tampilan halaman data laporan.



Gambar 3. 34 Rancangan Layout Halaman Data Laporan

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

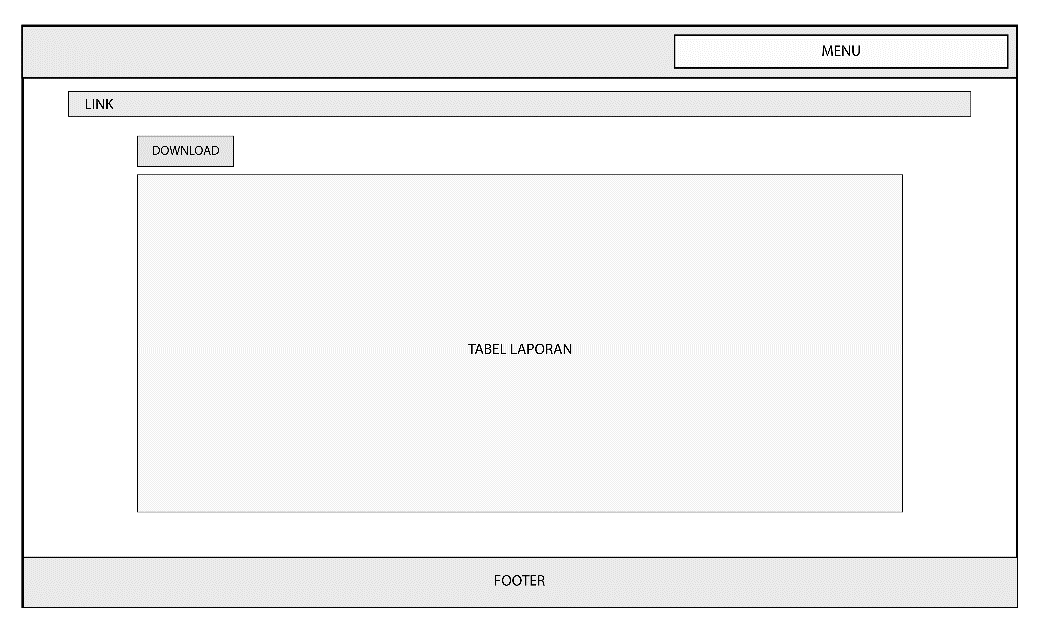
Berikut rancangan tampilan halaman data pustakawan dan *admin*. Mirip dengan data anggota, data *admin*/pustakawan ditampilkan dengan menggunakan *card*.



Gambar 3. 35 Rancangan Layout Halaman Data Pustakawan/Admin

## **Perancangan *Layout* Halaman Cetak Laporan**

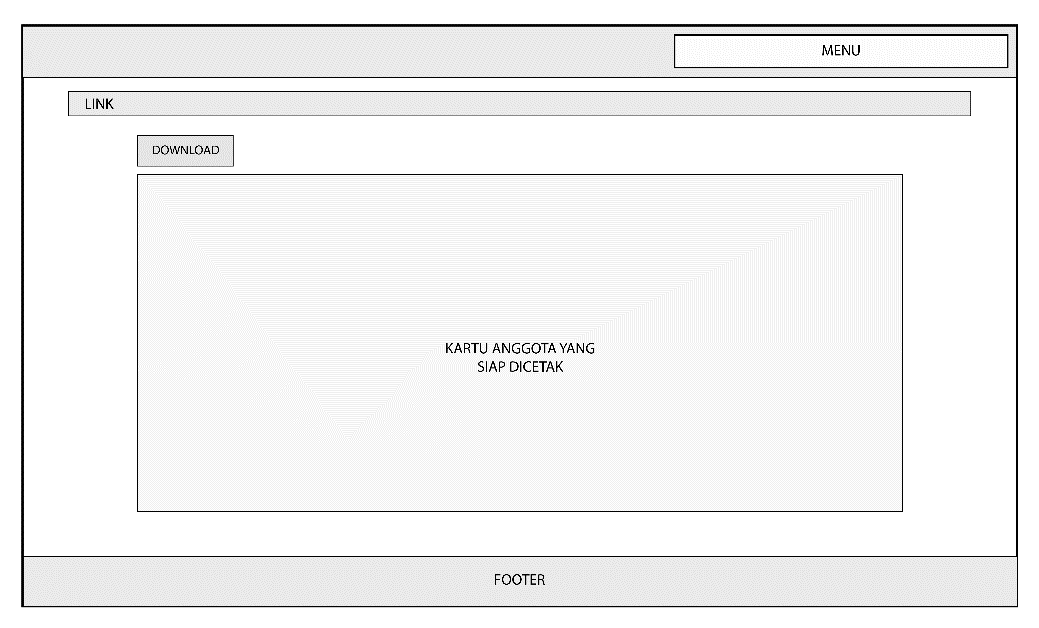
Berikut rancangan tampilan halaman untuk mencetak laporan menjadi *file* PDF. Bagian tengah berisikan *preview* *file* yang siap untuk dicetak.



Gambar 3. 36 Rancangan Layout Halaman Cetak Laporan

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

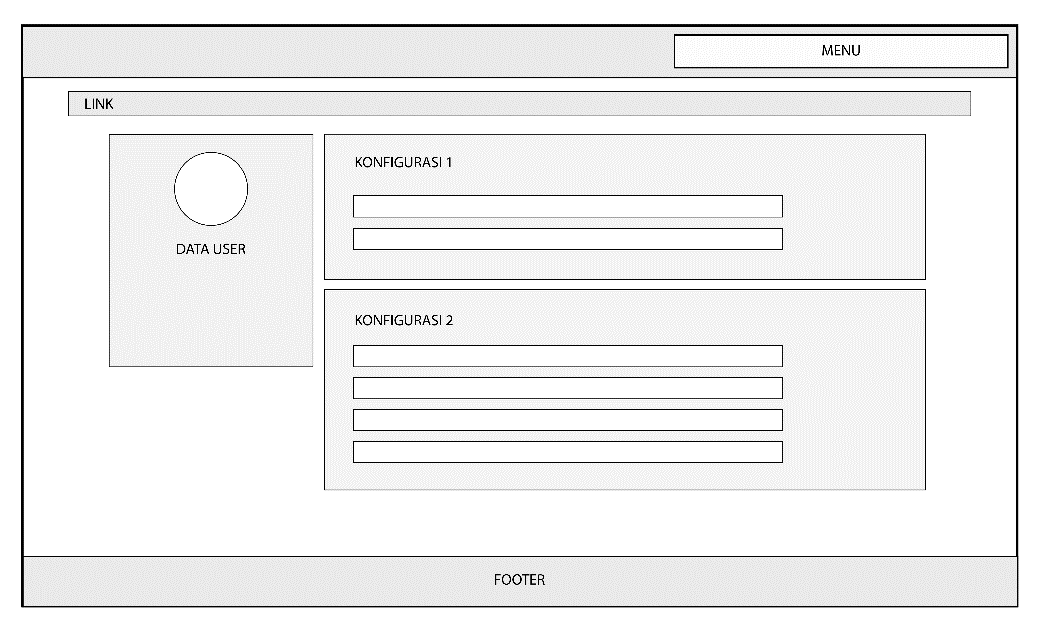
Berikut rancangan tampilan halaman untuk mencetak kartu anggota. Bagian tengah berisikan *preview* *file* yang siap untuk dicetak.



Gambar 3. 37 Rancangan Layout Halaman Cetak Kartu Anggota

## **Perancangan *Layout* Halaman Konfigurasi**

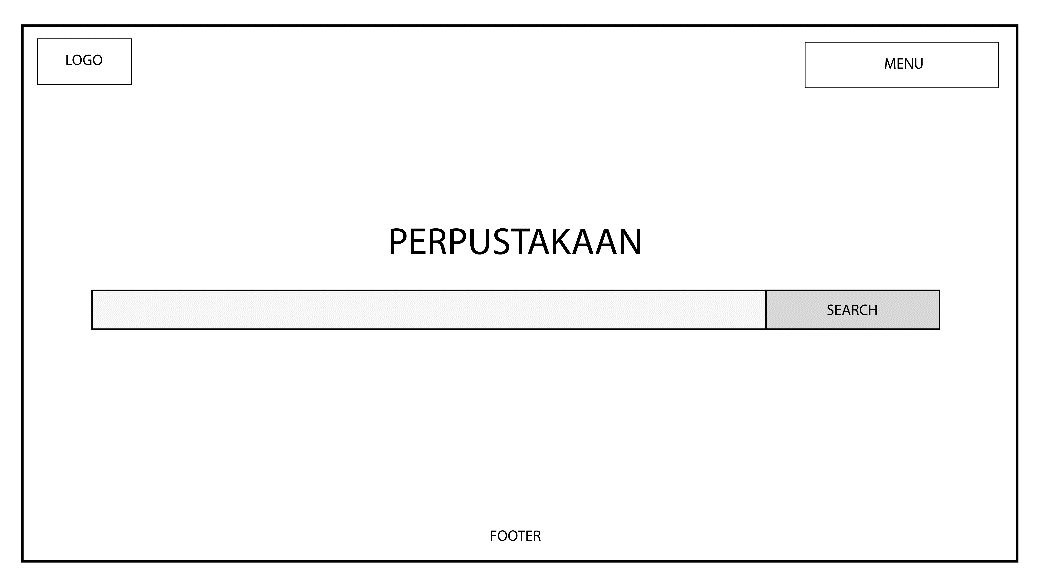
Berikut rancangan tampilan halaman konfigurasi.



Gambar 3. 38 Rancangan Layout Halaman Konfigurasi

## **Perancangan *Layout* Halaman *Dashboard* Anggota**

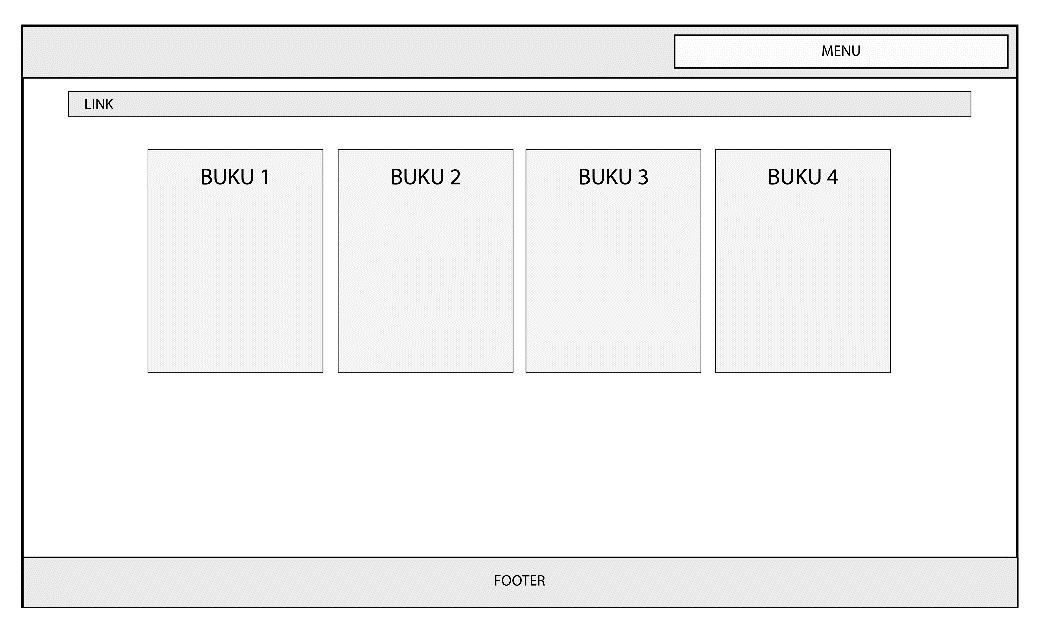
Berikut rancangan tampilan halaman *dashboard* untuk anggota. *Form* *input* di bagian tengah berfungsi untuk kolom *search* buku yang dapat dicari sesuai judul, penerbit atau kategori.



Gambar 3. 39 Rancangan Layout Halaman Dashboard Anggota

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

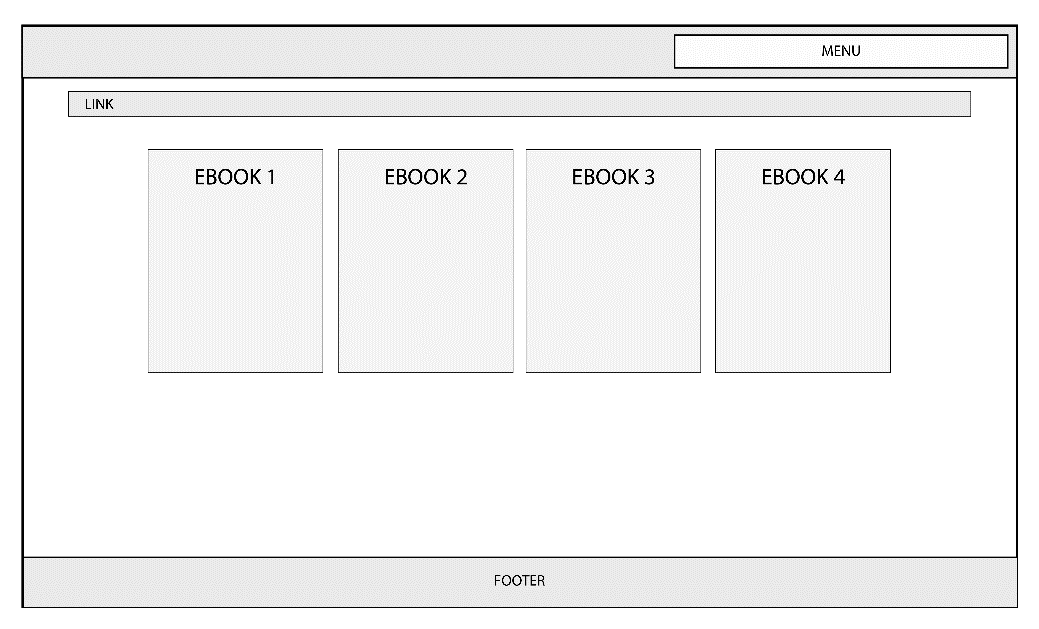
Berikut rancangan tampilan halaman data buku untuk anggota. Tiap data yang tampil hanya berisikan judul dan gambar *cover* buku.



Gambar 3. 40 Rancangan Layout Halaman Data Buku Anggota

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

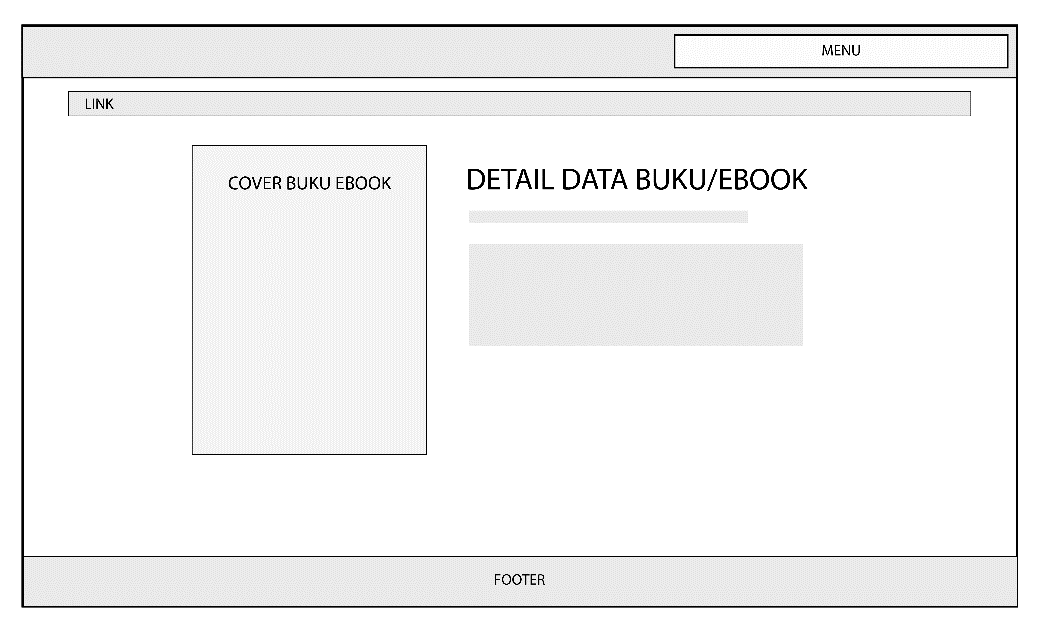
Berikut rancangan tampilan halaman data ebook untuk anggota. Tiap data yang tampil hanya berisikan judul dan gambar *cover* *ebook*.



Gambar 3. 41 Rancangan Layout Halaman Data Ebook Anggota

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

Berikut rancangan tampilan halaman detail buku dan *ebook*.



Gambar 3. 42 Rancangan Layout Halaman Detail Buku/Ebook

# **BAB IV IMPLEMENTASI DAN PENGUJIAN**

## **Implementasi Sistem**

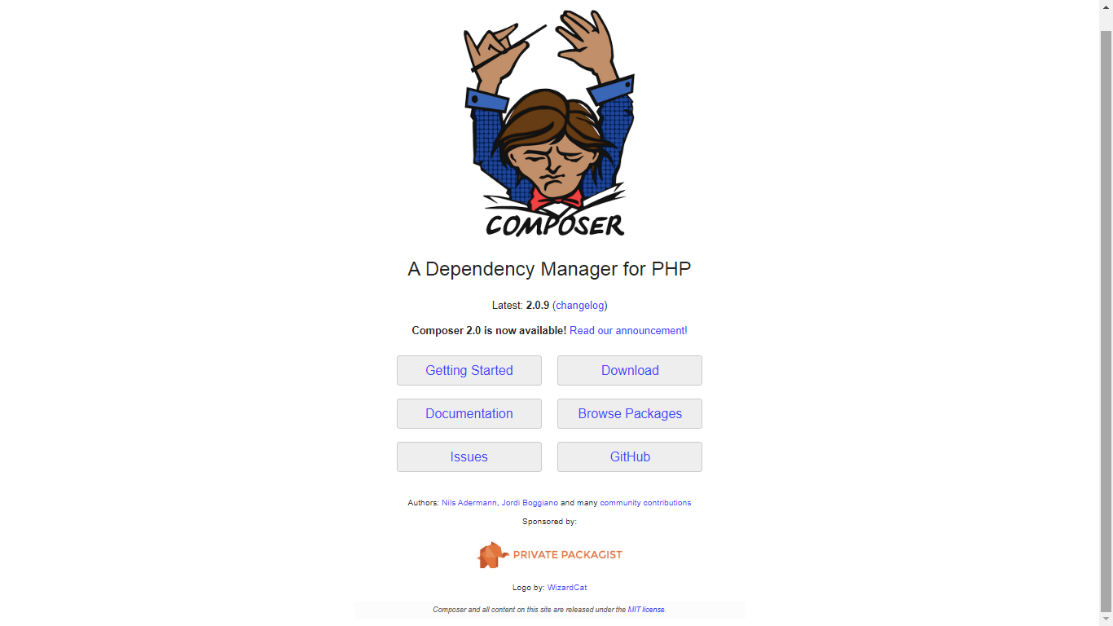
Implementasi sistem merupakan bagian yang menjelaskan mengenai tahap pembuatan sistem aplikasi dari analisis dan perencanaan yang dijelaskan di bab sebelumnya. Terdiri atas instalasi perangkat lunak (*software*) yang dibutuhkan, pembuatan projek dan konfigurasi *database* pada projek.

1. Instalasi *software*

Selama proses pembuatan aplikasi, tak lepas dari beberapa *software* pendukung. Instalasi ini dilakukan karena Laravel yang digunakan dalam projek ini, membutuhkan beberapa *service* yang diberikan oleh *software*-*software* berikut.

*Composer*

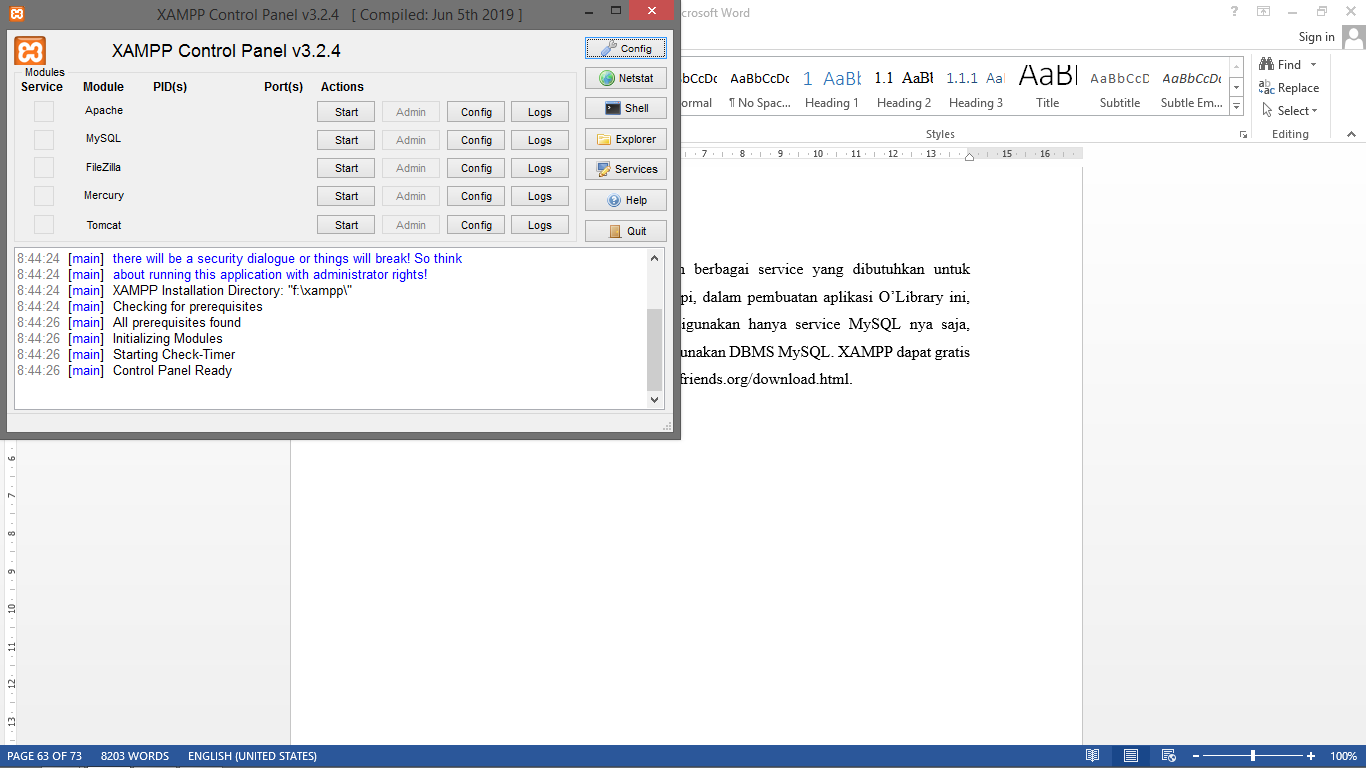
*Composer* merupakan *software* yang sangat dibutuhkan untuk pembuatan projek menggunakan Laravel. Fungsi *composer* yaitu sebagai aplikasi yang mengurusi *dependency* atau *library* Laravel yang dibutuhkan. *Composer* dapat didownload dan diinstall secara gratis di <https://getcomposer.org>.



Gambar 4. 1 Tampilan Composer

XAMPP

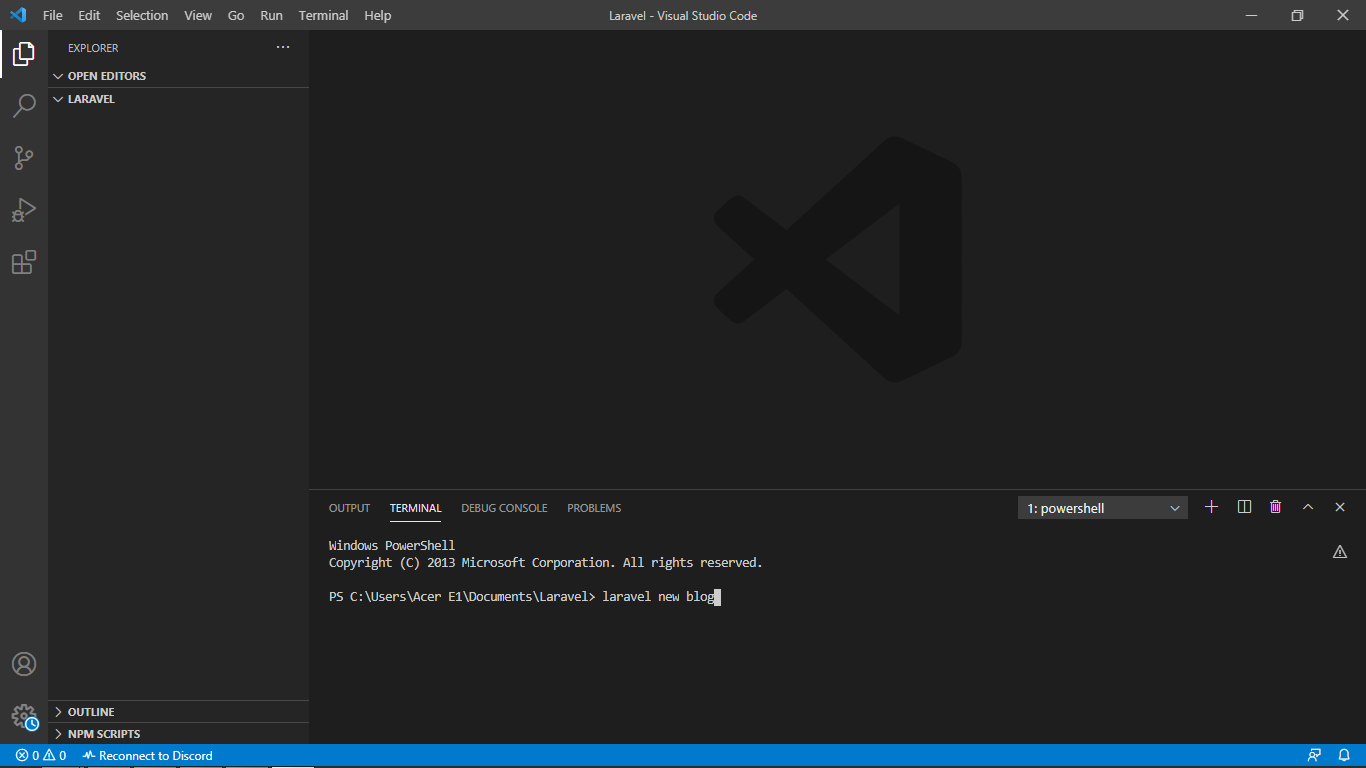
XAMPP menyediakan berbagai *service* yang dibutuhkan untuk pembuatan aplikasi. Tetapi, dalam pembuatan aplikasi O’Library ini, *service* XAMPP yang digunakan hanya *service* MySQL nya saja, karena aplikasi ini menggunakan DBMS MySQL. XAMPP dapat gratis diunduh di <https://apachefriends.org/download.html>.



Gambar 4. 2 Tampilan XAMPP

1. Pembuatan projek laravel

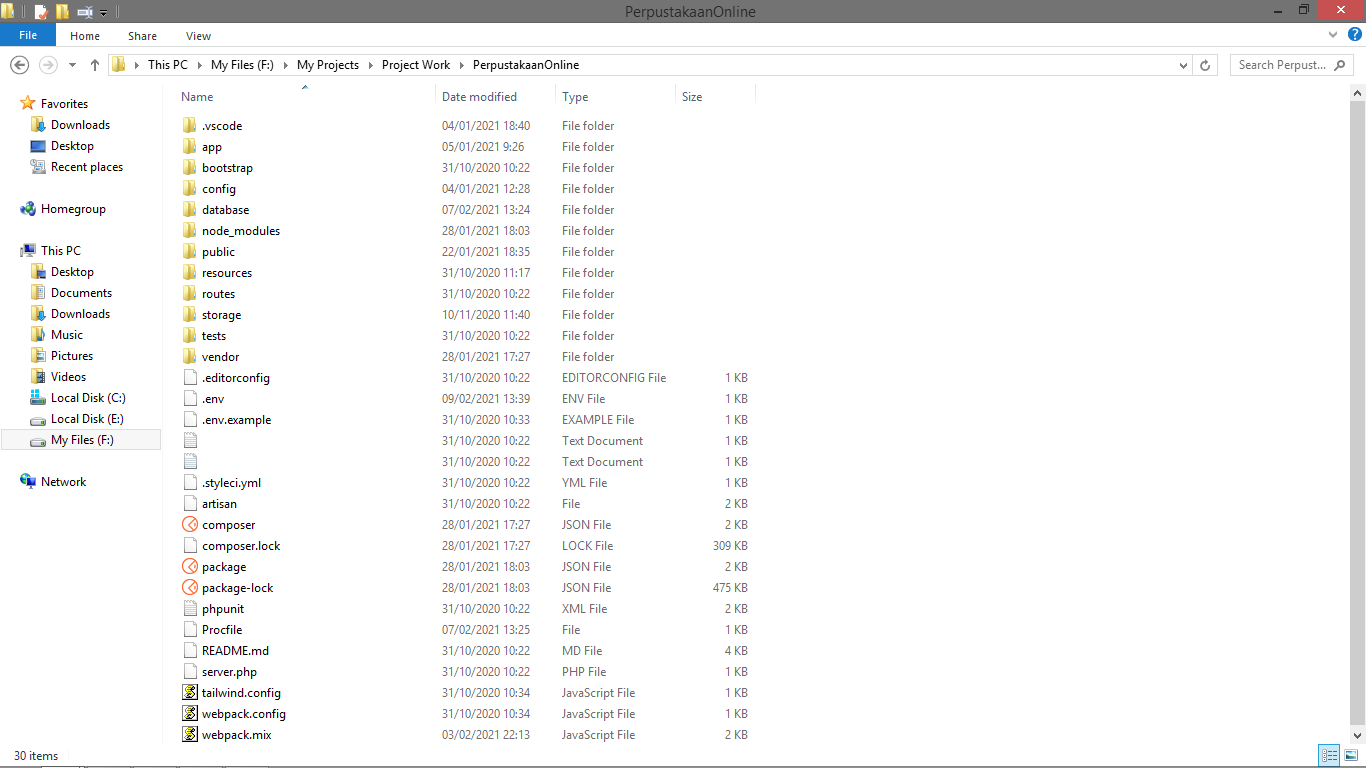
Setelah menginstall semua *software* yang dibutuhkan, langkah selanjutnya yaitu membuat projek laravel. Untuk informasi lebih lanjut mengenai pembuatan projek laravel dapat ditemukan di *website* dokumentasi laravel atau dengan mengikuti perintah berikut.



Gambar 4. 3 Perintah Untuk Membuat Projek Laravel

Dengan mengeksekusi perintah di atas, maka proses pemasangan projek dengan laravel akan dilakukan. Dibutuhkan waktu beberapa menit hingga proses pembuatan projek selesai. Dari perintah di atas, dijelaskan bahwa *user* akan membuat folder projek laravel yang bernama *blog*. *Blog* dari perintah di atas dapat diganti sesuai dengan keinginan *user*.

Setelah proses pembuatan projek selesai, maka akan terlihat susunan direktori sebagai berikut.



Gambar 4. 4 Direktori Projek Laravel

Berikut penjelasan dari setiap *folder*/*file* yang ada dalam projek laravel yang disajikan dalam bentuk tabel.

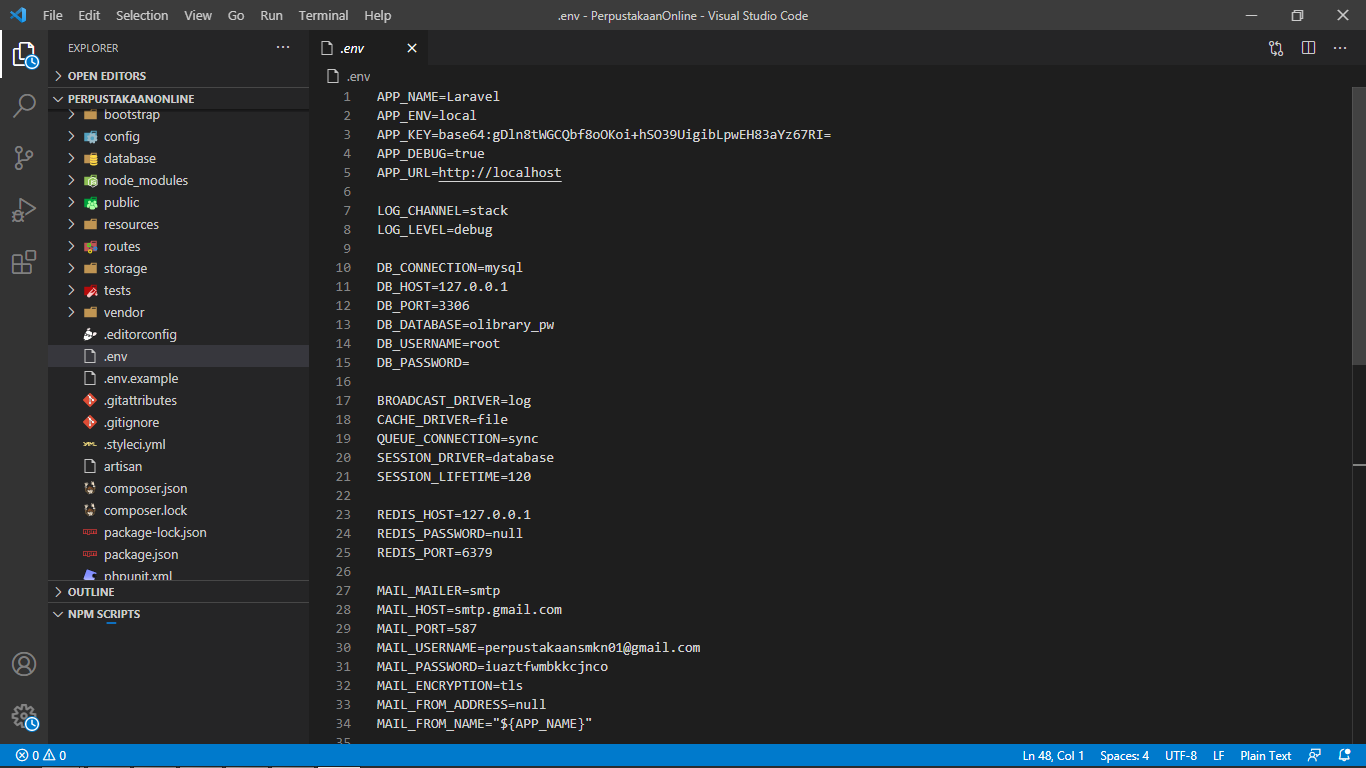
Tabel 4. 1 Penjelasan Direktori Projek Laravel

|  |  |
| --- | --- |
| ***File*/*folder*** | **Penjelasan** |
| *App* | Berisikan *file*-*file* yang berkaitan dengan proses request dan *response* Http, berisi komponen penting MVC seperti *controller* dan *model*. |
| Bootstrap | Berisikan *file-file* *framework* bootstrap |
| *Config* | *Folder* yang berisi *file* untuk mengkonfigurasi aplikasi yang dibuat. |
| *Database* | Berisi *file* *migration* atau *seeder* untuk kebutuhan pembuatan *database*. |
| Node\_modules | *File*-*file* yang dibutuhkan, dihasilkan oleh *package* *manager* seperti npm dan yarn. |
| *Public* | Berisikan *file* utama laravel, projek dijalankan dari direktori ini. |
| *Resources* | *Folder* penting yang berisi kebutuhan aplikasi, seperti *file* css, scss/sass atau js. Terdapat komponen penting MVC di *folder* ini yaitu *view*. |
| *Routes* | Berisikan *file* yang mengurus bagian *route*. |
| *Storage* | Berisi *file* penyimpanan yang dihasilkan dari aplikasi. |
| *Tests* | Berisi *file* *tes* otomatis untuk aplikasi. |
| Vendor | Berisikan dependensi yang dibutuhkan oleh aplikasi. |
| .env | Merupakan *file* konfigurasi sensitif untuk aplikasi. |
| .env.example | *File* yang berisi contoh dasar dari *file* .env. |
| Artisan | *File* yang berfungsi untuk menjalankan perintah artisan. |
| Composer.json | *File* yang mengidentifikasi *package* yang dibutuhkan aplikasi. |
| Composer.lock | *File* yang berisi informasi mengenai *package* terakhir yang dipasang dalam aplikasi. |
| Package.json | *File* yang berisi daftar dependensi yang dibutuhkan oleh laravel. |
| Procfile | *File* yang berisi konfigurasi agar aplikasi dapat dihosting ke Heroku. |
| README.md | Merupakan *file* yang berisi dokumentasi dari laravel. |
| Server.php | *File* yang berisi konfigurasi *server*. |
| Taiwind.config | *File* konfigurasi untuk aplikasi jika menggunakan *framework* tailwind. |
| Webpack.config | *File* konfigurasi untuk aplikasi jika menggunakan *bundler* webpack. |
| Webpack.mix | Berisi daftar *file* yang akan di *bundle* menggunakan webpack. |

1. Konfigurasi *database*

Dalam aplikasi tentu menggunakan *database* sebagai sumber data. Berikut tahap-tahap yang dilakukan *user* sebelum dapat menggunakan *database* (khusus pengguna DBMS MySQL).

1. Konfigurasi *file* .env



Gambar 4. 5 File .env

Dalam *file* .env, terdapat konfigurasi seperti gambar di atas. *User* wajib untuk mengisi konfigurasi tersebut agar *database* dapat terintegerasi dengan baik.

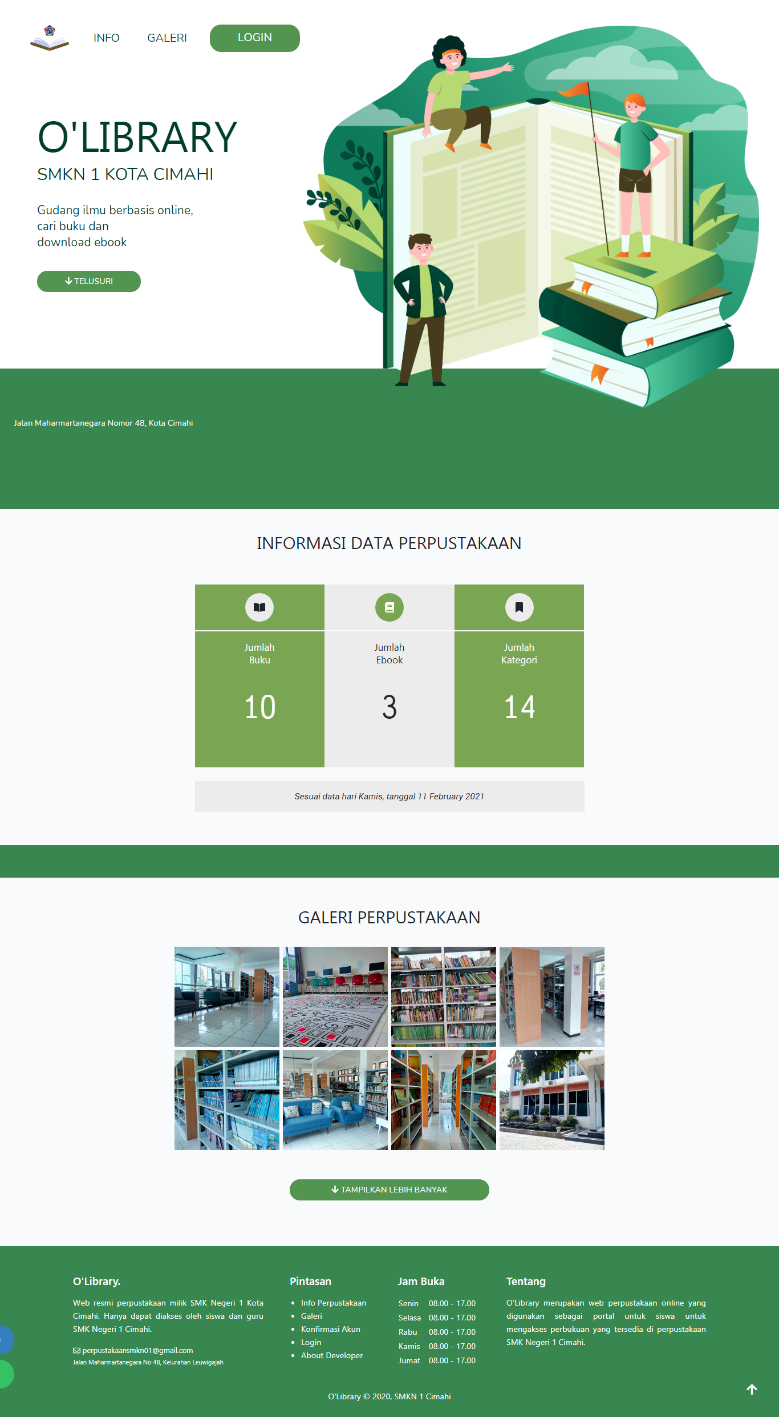
1. *Migrating* dan *seeding* *database*

*Migrate* adalah proses pembuatan struktur tabel dalam *database*, sedangkan *seed* adalah mengisi *record* yang dibutuhkan untuk tabel tersebut. Dengan menggunakan kedua hal ini, *user* tidak perlu untuk membuka *software* pengelola *database* seperti phpmyadmin atau Navicat, karena akan dikelola menggunakan kode PHP.

## **Implementasi Tampilan Aplikasi**

## **Implementasi Tampilan Halaman *Landing* *Page***

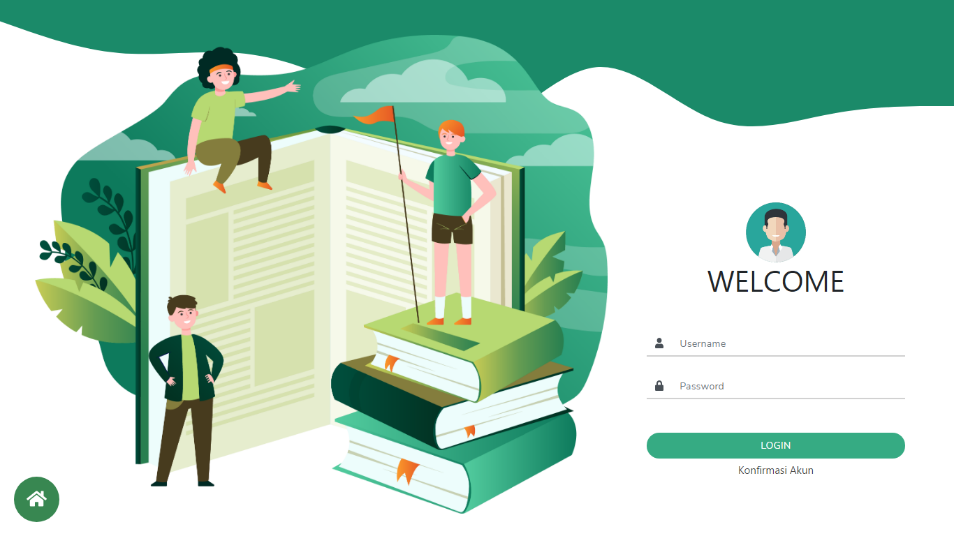
Berikut merupakan implementasi tampilan halaman untuk *landing* *page*. Terdiri atas 3 *section* utama yang memiliki fungsinya masing-masing.



Gambar 4. 6 Implementasi Tampilan Halaman Landing Page

## **Implementasi Tampilan Halaman *Login***

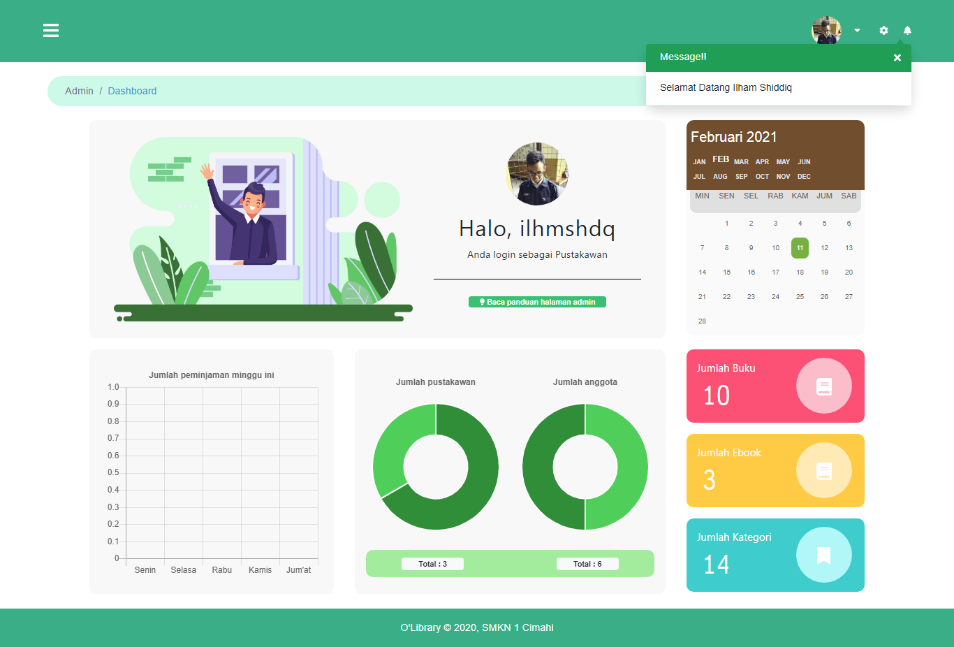
Berikut merupakan implementasi tampilan halaman untuk *login*. Halaman ini menjadi portal *user* sebelum dapat menggunakan *web* aplikasi.



Gambar 4. 7 Implementasi Tampilan Halaman Login

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

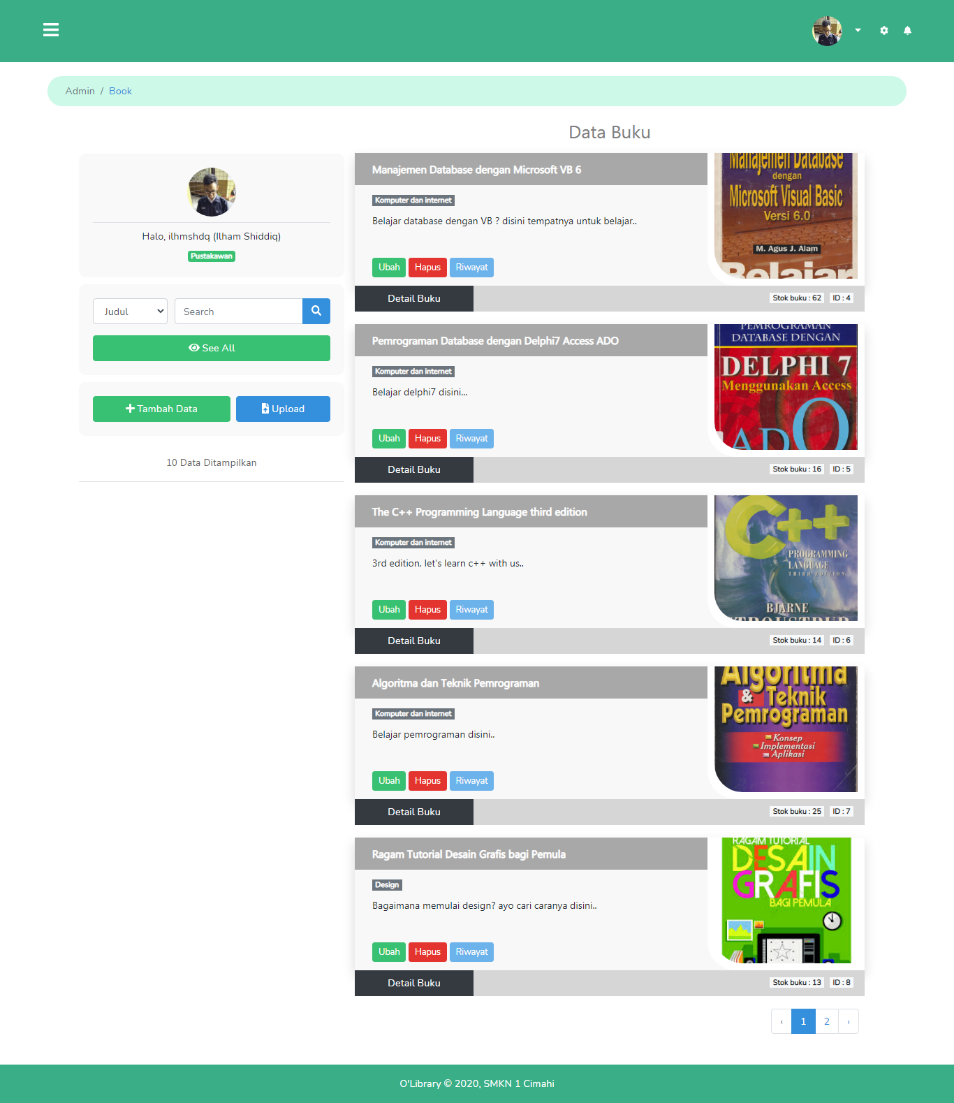
Berikut merupakan implementasi tampilan halaman untuk *dashboard* *admin* dan pustakawan. Halaman ini mengandung beberapa informasi penting yang disajikan dalam bentuk grafik, seperti data buku, transaksi yang terjadi perminggunya, jumlah anggota serta pustakawan.



Gambar 4. 8 Implementasi Tampilan Halaman Dashboard Admin/Pustakawan

## **Implementasi Tampilan Halaman Data Buku**

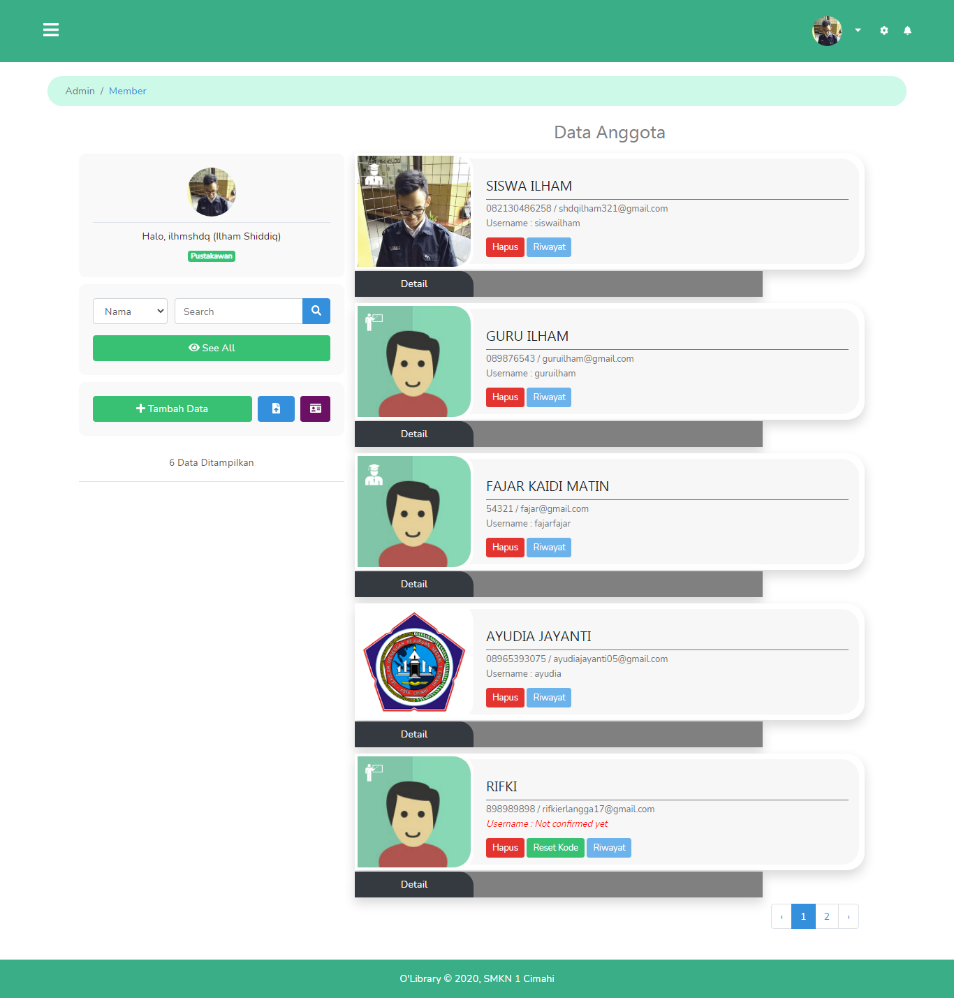
Berikut merupakan implementasi tampilan halaman untuk data buku yang dapat diakses oleh pustakawan. Menampilkan daftar koleksi buku yang tersedia di perpustakaan.



Gambar 4. 9 Implementasi Tampilan Halaman Data Buku

## **Implementasi Tampilan Halaman Data Anggota**

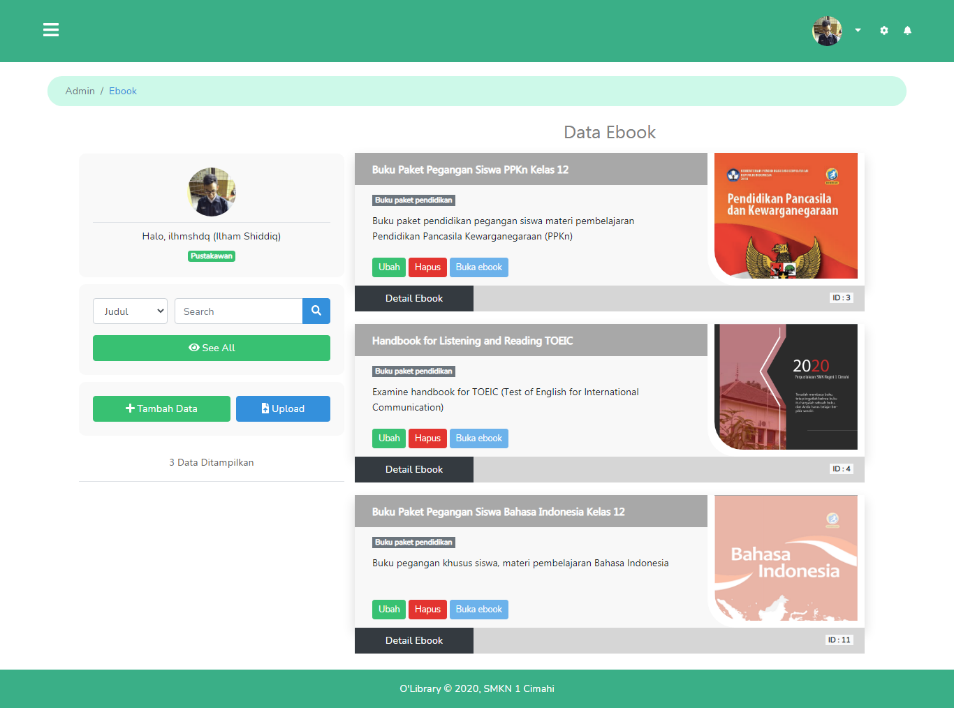
Berikut merupakan implementasi tampilan halaman untuk data anggota yang dapat diakses oleh pustakawan. Menampilkan *list* anggota yang sudah terdaftar.



Gambar 4. 10 Implementasi Tampilan Halaman Data Anggota

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

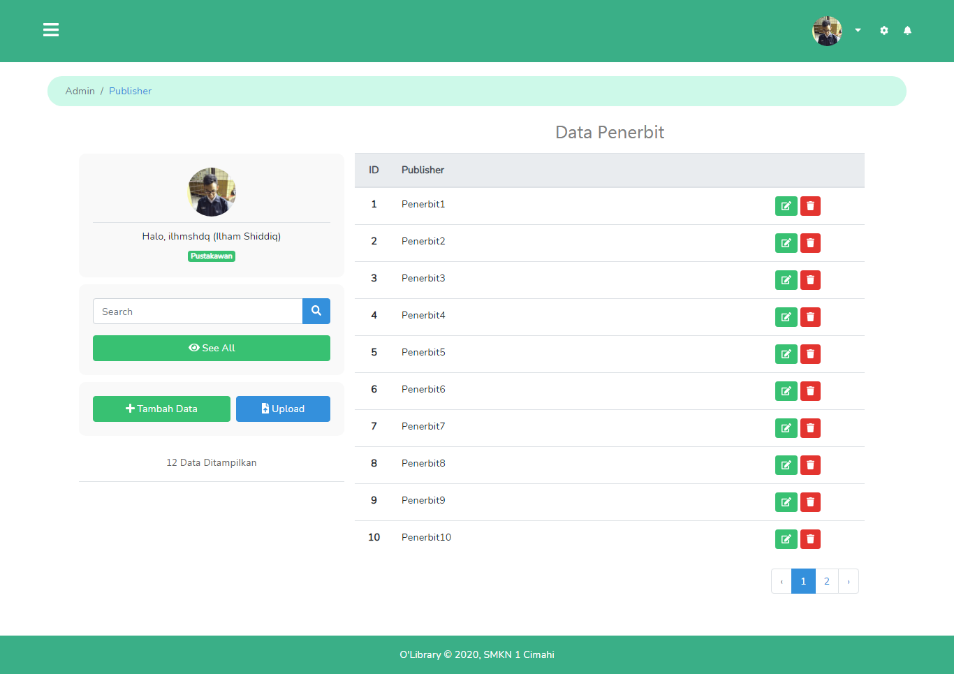
Berikut merupakan implementasi tampilan halaman untuk data anggota yang dapat diakses oleh pustakawan. Menampilkan daftar koleksi *ebook* yang tersedia di perpustakaan. Anggota dapat menggunakan *ebook* ini sesuai dengan persetujuan pustakawan.



Gambar 4. 11 Implementasi Tampilan Halaman Data Ebook

## **Implementasi Tampilan Halaman Data Penerbit**

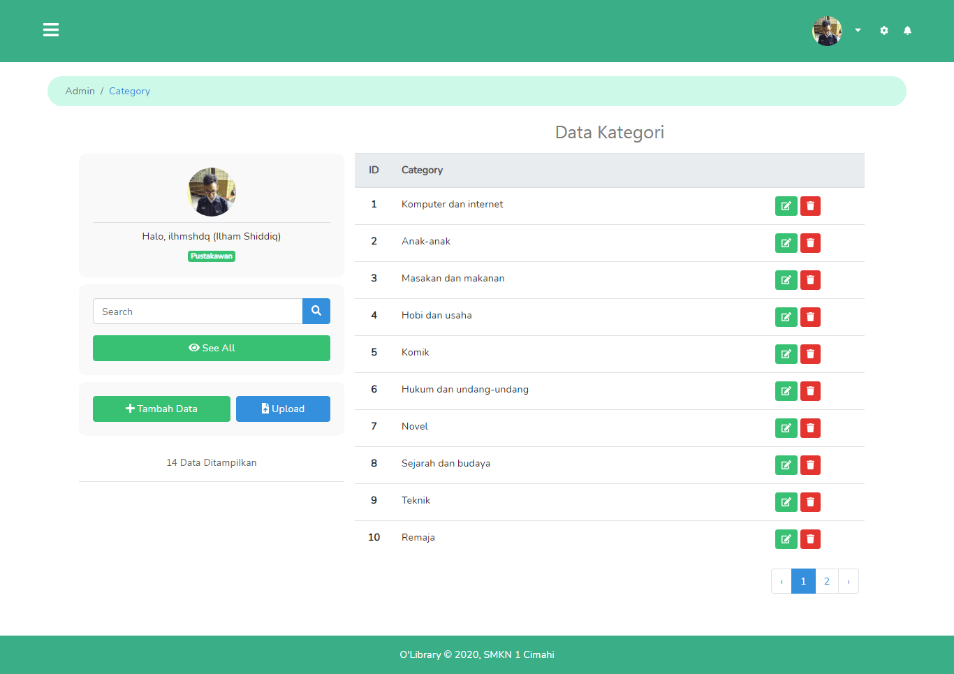
Berikut merupakan implementasi tampilan halaman untuk data penerbit yang dapat diakses oleh pustakawan.



Gambar 4. 12 Implementasi Tampilan Halaman Data Penerbit

## **Implementasi Tampilan Halaman Data Kategori**

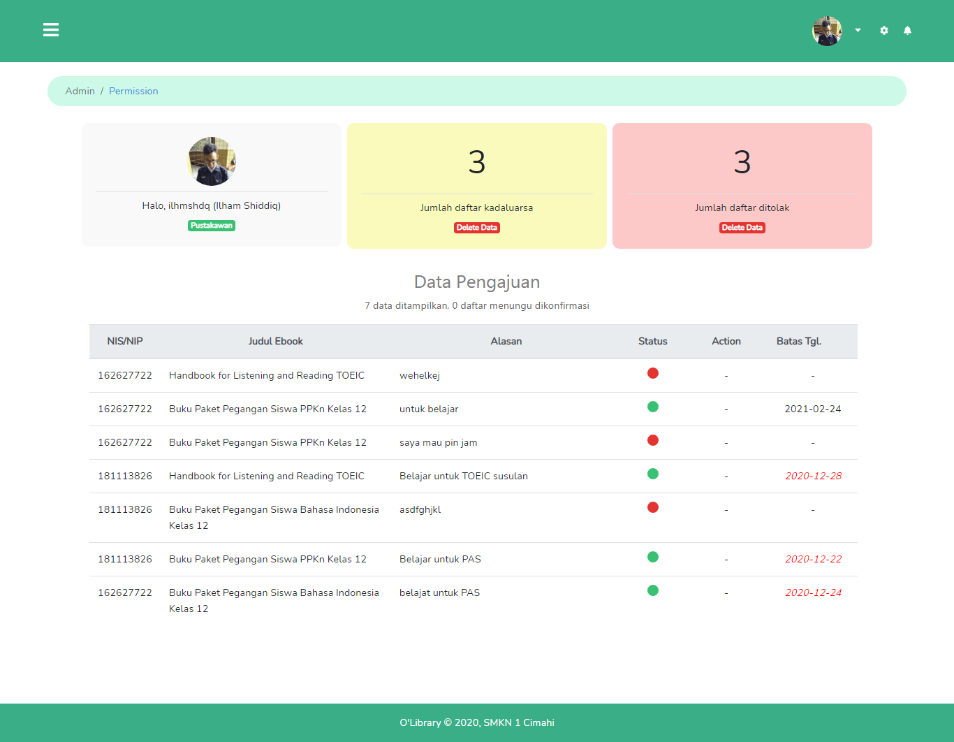
Berikut merupakan implementasi tampilan halaman untuk data kategori yang dapat diakses oleh pustakawan.



Gambar 4. 13 Implementasi Tampilan Halaman Data Kategori

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

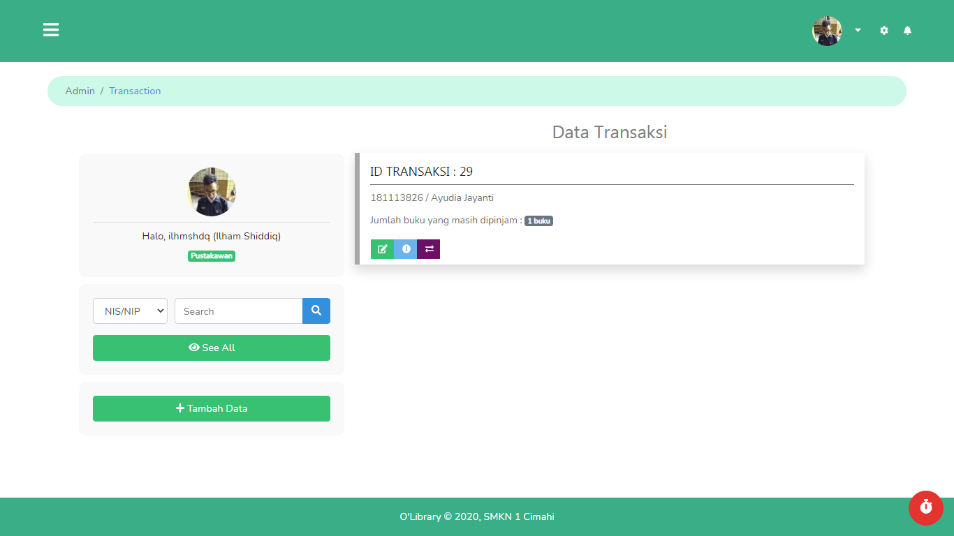
Berikut merupakan implementasi tampilan halaman untuk data pengajuan *ebook* yang dapat diakses oleh pustakawan. Sebelum anggota dapat menggunakan *ebook*, mereka membutuhkan terlebih dahulu persetujuan dari pustakawan, dan di halaman ini, pustakawan akan membuat keputusan apakah pengajuan disetujui atau ditolak sesuai dengan alasan yang diberikan.



Gambar 4. 14 Implementasi Tampilan Halaman Data Pengajuan Ebook

## **Implementasi Tampilan Halaman Data Transaksi**

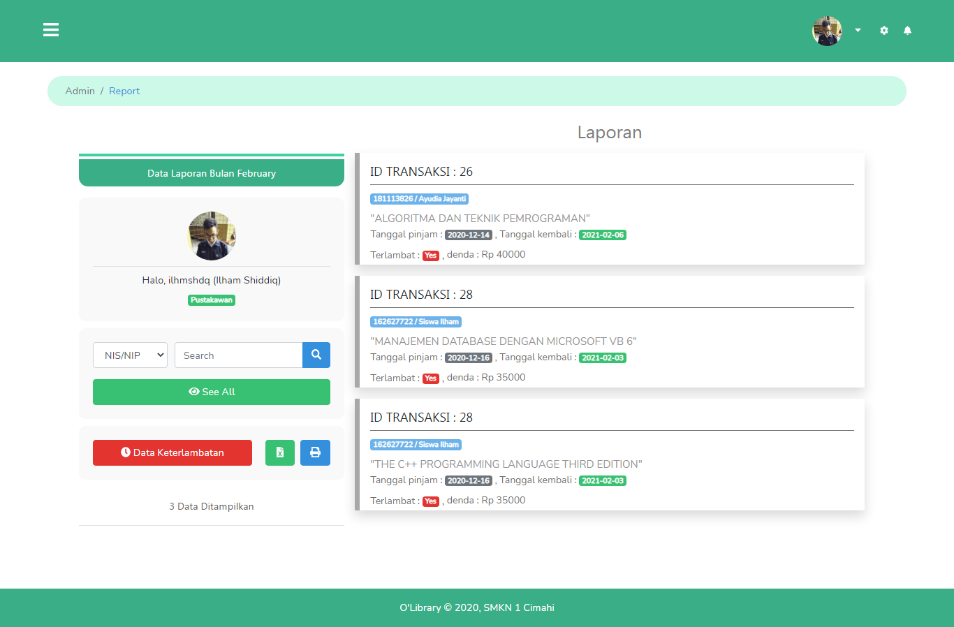
Berikut merupakan implementasi tampilan halaman untuk data transaksi (peminjaman buku) yang dapat diakses oleh pustakawan.



Gambar 4. 15 Implementasi Tampilan Halaman Data Transaksi

## **Implementasi Tampilan Halaman Data Laporan**

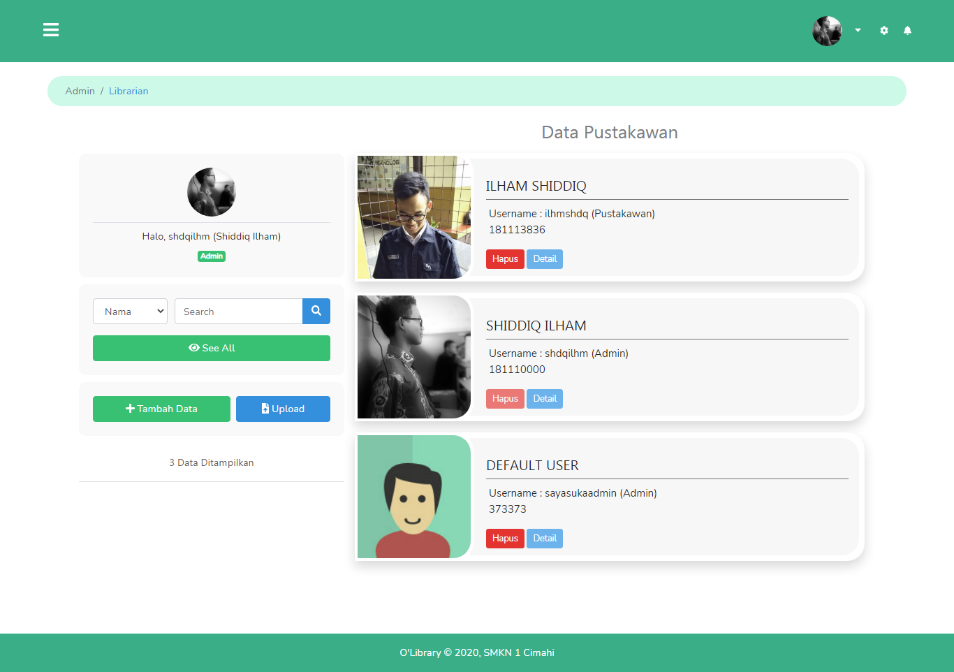
Berikut merupakan implementasi tampilan halaman untuk data laporan yang dapat diakses oleh pustakawan.



Gambar 4. 16 Implementasi Tampilan Halaman Data Laporan

## **Implementasi Tampilan Halaman Data Pustakawan**

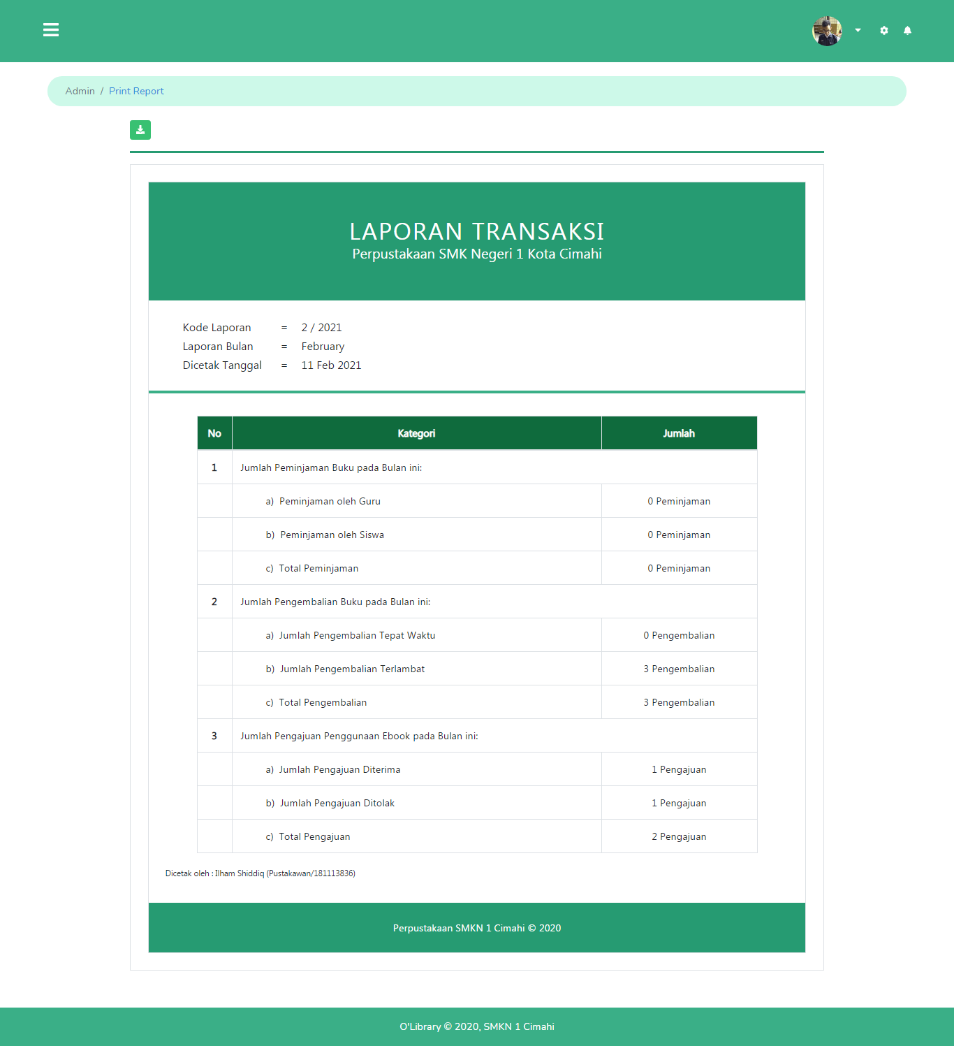
Berikut merupakan implementasi tampilan halaman untuk data pustakawan yang dapat diakses oleh *admin*.



Gambar 4. 17 Implementasi Tampilan Halaman Data Pustakawan

## **Implementasi Tampilan Halaman Cetak Laporan**

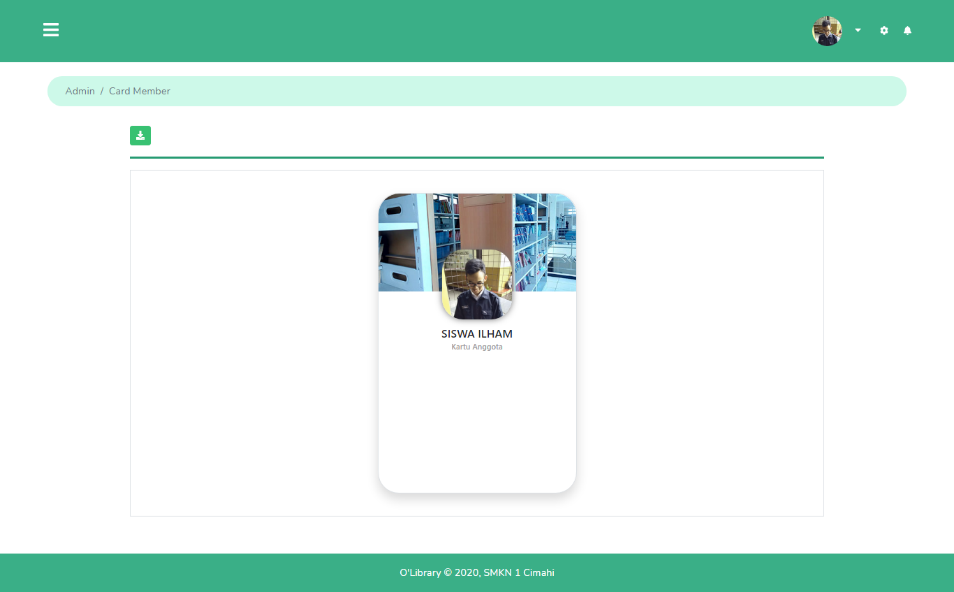
Berikut merupakan implementasi tampilan halaman untuk mencetak laporan dalam *file* pdf yang dapat diakses oleh pustakawan maupun *admin*.



Gambar 4. 18 Implementasi Tampilan Halaman Cetak Laporan

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

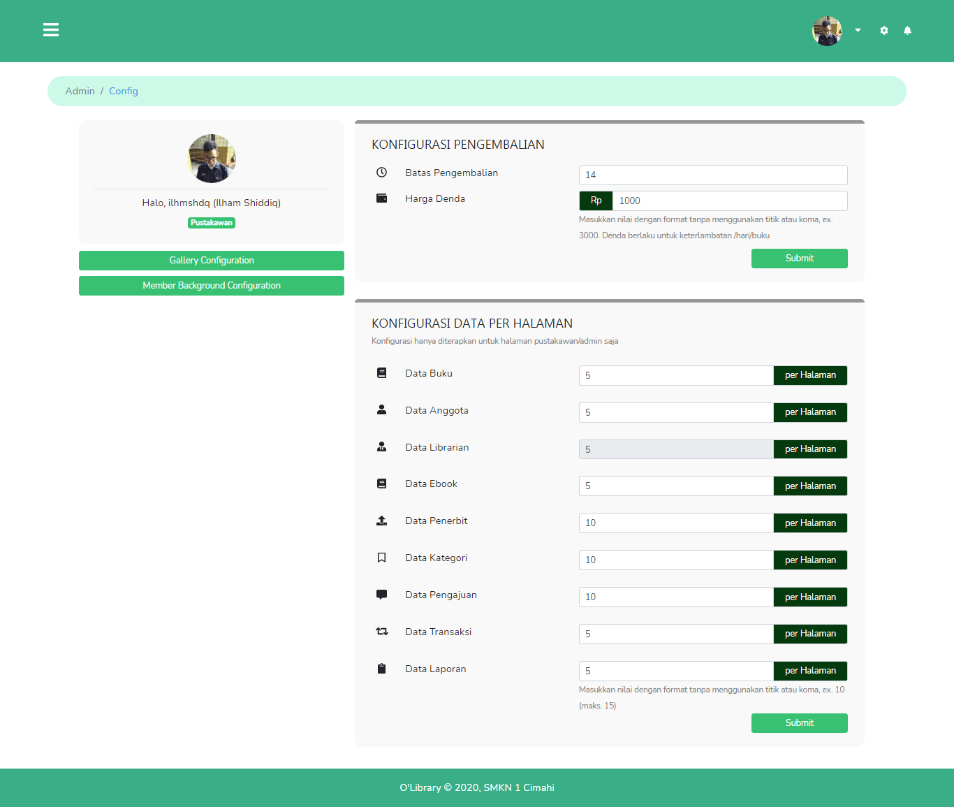
Berikut merupakan implementasi tampilan halaman untuk mencetak kartu anggota dalam *file* pdf yang dapat diakses oleh pustakawan.



Gambar 4. 19 Implementasi Tampilan Halaman Cetak Kartu Anggota

## **Implementasi Tampilan Halaman Konfigurasi**

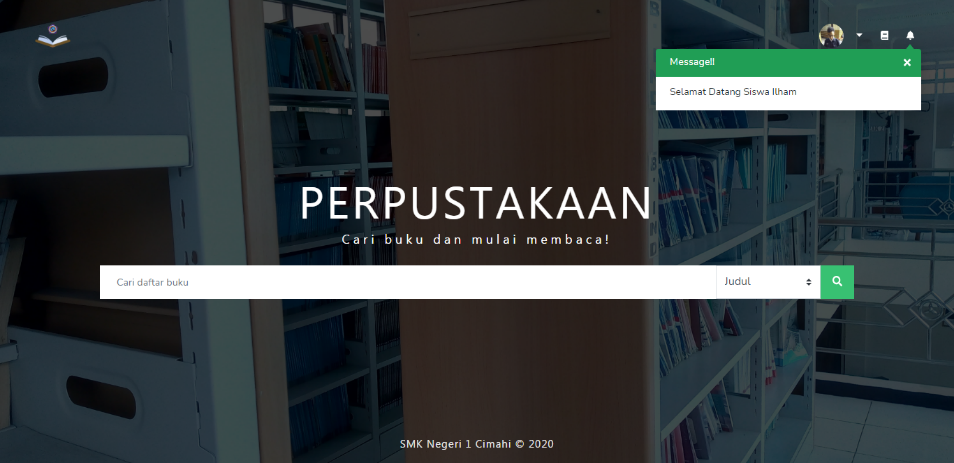
Berikut merupakan implementasi tampilan halaman untuk konfigurasi yang dapat diakses oleh pustakawan maupun *admin*.



Gambar 4. 20 Implementasi Tampilan Halaman Konfigurasi

## **Implementasi Tampilan Halaman Dashboard Anggota**

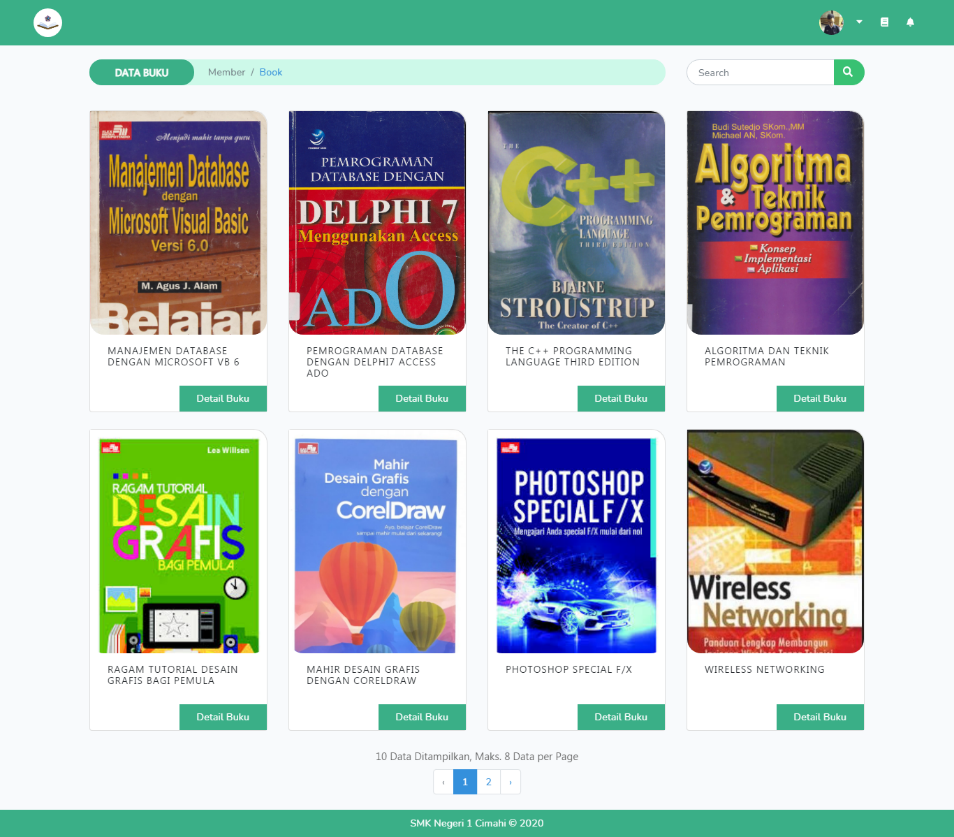
Berikut merupakan implementasi tampilan halaman untuk *dashboard* yang dapat diakses oleh anggota.



Gambar 4. 21 Implementasi Tampilan Halaman Dashboard Anggota

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

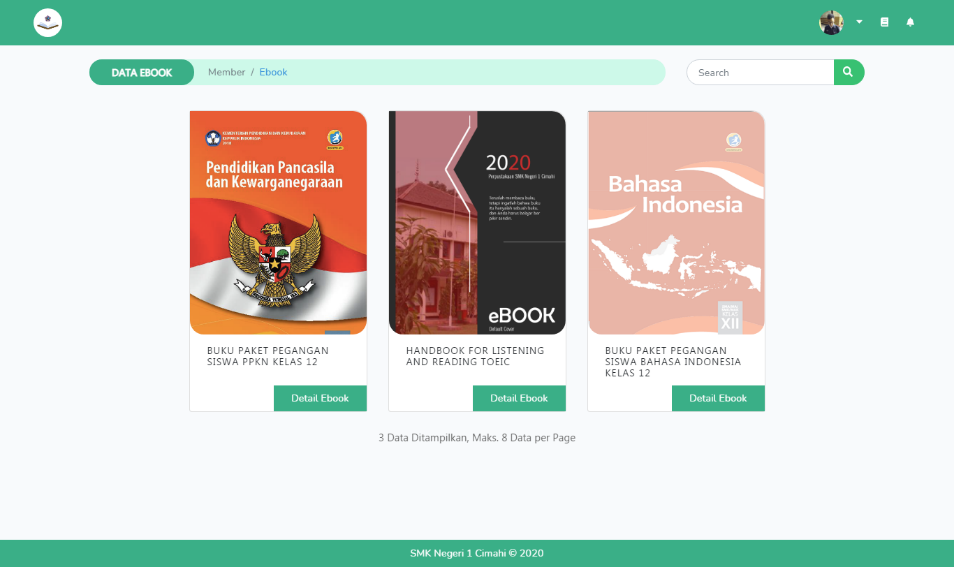
Berikut merupakan implementasi tampilan halaman untuk data buku yang dapat diakses oleh anggota.



Gambar 4. 22 Implementasi Tampilan Halaman Data Buku Anggota

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

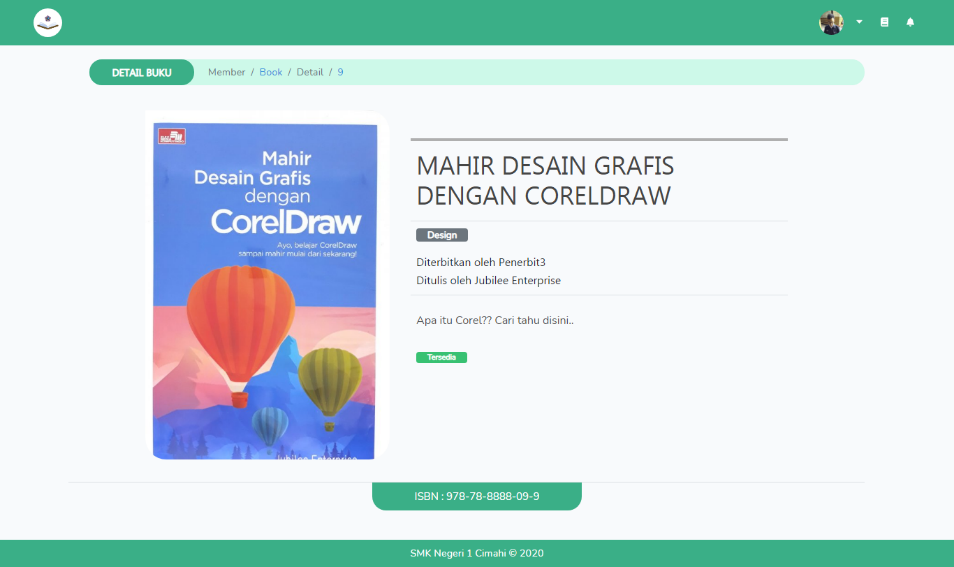
Berikut merupakan implementasi tampilan halaman untuk data *ebook* yang dapat diakses oleh anggota.



Gambar 4. 23 Implementasi Tampilan Halaman Data Ebook Anggota

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

Berikut merupakan implementasi tampilan halaman untuk detail buku/*ebook* yang dapat diakses oleh anggota.



Gambar 4. 24 Implementasi Tampilan Halaman Detail Buku/Ebook

## **Struktur *Database***

Berikut merupakan susunan struktur tiap tabel dalam *database* yang dipakai di aplikasi O’Library, disajikan dalam bentuk tabel.

Tabel *books*

Tabel 4. 2 Struktur Tabel books

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kolom** | **Tipe Data** | **Extra** | **Keterangan** |
| 1 | Id | Bigint(20) | Auto\_increment | PK |
| 2 | Isbn | Char(20) |  |  |
| 3 | Title | Varchar(255) | NOT NULL |  |
| 4 | Publisher\_id | Int(11) | NOT NULL | FK |
| 5 | Author | Varchar(255) | NOT NULL |  |
| 6 | Category\_id | Int(11) | NOT NULL | FK |
| 7 | Qty | Int(11) | NOT NULL |  |
| 8 | Image | Varchar(255) | NOT NULL |  |
| 9 | About | Text |  |  |
| 10 | Publish\_year | Char(4) | NOT NULL |  |
| 11 | Created\_at | Timestamp |  |  |
| 12 | Updated\_at | Timestamp |  |  |

Tabel *categories*

Tabel 4. 3 Struktur Tabel categories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kolom** | **Tipe Data** | **Extra** | **Keterangan** |
| 1 | Id | Bigint(20) | Auto\_increment | PK |
| 2 | Category | Varchar(255) | NOT NULL |  |
| 3 | Created\_at | Timestamp |  |  |
| 4 | Updated\_at | Timestamp |  |  |

Tabel *publishers*

Tabel 4. 4 Struktur Tabel publishers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kolom** | **Tipe Data** | **Extra** | **Keterangan** |
| 1 | Id | Bigint(20) | Auto\_increment | PK |
| 2 | Publisher | Varchar(255) | NOT NULL |  |
| 3 | Created\_at | Timestamp |  |  |
| 4 | Updated\_at | Timestamp |  |  |

Tabel *members*

Tabel 4. 5 Struktur Tabel members

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kolom** | **Tipe Data** | **Extra** | **Keterangan** |
| 1 | Id | Bigint(20) | Auto\_increment | PK |
| 2 | Address | Text | NOT NULL |  |
| 3 | Phone | Char(16) | NOT NULL |  |
| 4 | Status | Varchar(255) | NOT NULL |  |
| 5 | Class | Varchar(255) |  |  |
| 6 | Confirm\_code | Int(11) |  |  |
| 7 | Created\_at | Timestamp |  |  |
| 8 | Updated\_at | Timestamp |  |  |

Tabel *ebooks*

Tabel 4. 6 Struktur Tabel ebooks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kolom** | **Tipe Data** | **Extra** | **Keterangan** |
| 1 | Id | Bigint(20) | Auto\_increment | PK |
| 2 | Isbn | Char(20) |  |  |
| 3 | Title | Varchar(255) | NOT NULL |  |
| 4 | Publisher\_id | Int(11) | NOT NULL | FK |
| 5 | Author | Varchar(255) | NOT NULL |  |
| 6 | Category\_id | Int(11) | NOT NULL | FK |
| 7 | Link | Varchar(255) | NOT NULL |  |
| 8 | Image | Varchar(255) | NOT NULL |  |
| 9 | About | Text |  |  |
| 10 | Publish\_year | Char(4) | NOT NULL |  |
| 11 | Created\_at | Timestamp |  |  |
| 12 | Updated\_at | Timestamp |  |  |

Tabel *transactions*

Tabel 4. 7 Struktur Tabel transactions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kolom** | **Tipe Data** | **Extra** | **Keterangan** |
| 1 | Id | Bigint(20) | Auto\_increment | PK |
| 2 | Member\_id | Int(11) | NOT NULL | FK |
| 3 | Librarian\_id | Int(11) | NOT NULL | FK |
| 4 | Borrow\_date | Timestamp | NOT NULL |  |
| 5 | Created\_at | Timestamp |  |  |
| 6 | Updated\_at | Timestamp |  |  |

Tabel *detail* *transactions*

Tabel 4. 8 Struktur Tabel detail transactions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kolom** | **Tipe Data** | **Extra** | **Keterangan** |
| 1 | Transaction\_id | Int(11) | NOT NULL | FK |
| 2 | Book\_id | Int(11) | NOT NULL | FK |
| 3 | Date\_of\_return | Timestamp |  |  |
| 4 | Status | Char(1) | NOT NULL |  |
| 5 | Created\_at | Timestamp |  |  |
| 6 | Updated\_at | Timestamp |  |  |

Tabel *permissions*

Tabel 4. 9 Struktur Tabel permissions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kolom** | **Tipe Data** | **Extra** | **Keterangan** |
| 1 | Id | Bigint(20) | Auto\_increment | PK |
| 2 | Id\_member | Int(11) | NOT NULL | FK |
| 3 | Id\_ebook | Int(11) | NOT NULL | FK |
| 4 | Reason | Text | NOT NULL |  |
| 5 | Confirmed | Char(1) | NOT NULL |  |
| 6 | Accepted | Char(1) |  |  |
| 7 | Submit\_date | Date | NOT NULL |  |
| 8 | Limit\_date | Date |  |  |
| 9 | Created\_at | Timestamp |  |  |
| 10 | Updated\_at | Timestamp |  |  |

Tabel *users*

Tabel 4. 10 Struktur Tabel users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kolom** | **Tipe Data** | **Extra** | **Keterangan** |
| 1 | Id | Bigint(20) | Auto\_increment | PK |
| 2 | Nomor\_induk | Varchar(255) | NOT NULL |  |
| 3 | Name | Varchar(255) | NOT NULL |  |
| 4 | Username | Varchar(255) |  |  |
| 5 | Password | Varchar(255) |  |  |
| 6 | Role | Varchar(255) | NOT NULL |  |
| 7 | Email | Varchar(255) | NOT NULL |  |
| 8 | Profile\_photo\_  path | Text |  |  |
| 9 | Created\_at | Timestamp |  |  |
| 10 | Updated\_at | Timestamp |  |  |

Tabel *librarians*

Tabel 4. 11 Struktur Tabel librarians

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kolom** | **Tipe Data** | **Extra** | **Keterangan** |
| 1 | Id | Bigint(20) | Auto\_increment | PK |
| 2 | Address | Text | NOT NULL |  |
| 3 | Phone | Char(16) | NOT NULL |  |
| 4 | Confirm\_code | Int(11) |  |  |
| 5 | Created\_at | Timestamp |  |  |
| 6 | Updated\_at | Timestamp |  |  |

Tabel *configs*

Tabel 4. 12 Struktur Tabel configs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kolom** | **Tipe Data** | **Extra** | **Keterangan** |
| 1 | Id | Bigint(20) | Auto\_increment | PK |
| 2 | Late\_charge | Int(11) | NOT NULL |  |
| 3 | Loan\_deadline | Int(11) | NOT NULL |  |
| 4 | Book\_list\_page | Int(11) | NOT NULL |  |
| 5 | Member\_list\_  page | Int(11) | NOT NULL |  |
| 6 | Librarian\_list\_  page | Int(11) | NOT NULL |  |
| 7 | Ebook\_list\_page | Int(11) | NOT NULL |  |
| 8 | Publisher\_list\_  page | Int(11) | NOT NULL |  |
| 9 | Category\_list\_  page | Int(11) | NOT NULL |  |
| 10 | Permission\_list\_  page | Int(11) | NOT NULL |  |
| 11 | Transaction\_list  \_page | Int(11) | NOT NULL |  |
| 12 | Report\_list\_  Page | Int(11) | NOT NULL |  |
| 13 | Bg\_member | Varchar(255) | NOT NULL |  |
| 14 | Created\_at | Timestamp |  |  |
| 15 | Updated\_at | Timestamp |  |  |

Tabel *gallery* *configs*

Tabel 4. 13 Struktur Tabel gallery configs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Kolom** | **Tipe Data** | **Extra** | **Keterangan** |
| 1 | Id | Bigint(20) | Auto\_increment | PK |
| 2 | Gallery\_picture | Varchar(255) | NOT NULL |  |
| 3 | Created\_at | Timestamp |  |  |
| 4 | Updated\_at | Timestamp |  |  |

## **Diagram Relasi**

Berikut merupakan gambar yang menunjukkan hubungan antar tabel yang terdapat dalam *database* yang digunakan aplikasi O’Library. Memperlihatkan ketergantungan antara satu tabel dengan tabel lainnya. Serta terdapat pula tabel yang berdiri sendiri, tidak terhubung dengan tabel manapun.



Gambar 4. 25 Diagram Relasi

## **Pengujian**

Pengujian dalam pembuatan aplikasi memiliki peran untuk menjaga kualitas aplikasi yang dibuat saat digunakan nantinya, agar sesuai dengan kebutuhan aplikasi yang telah dijabarkan sebelumnya. Selain itu, pengujian juga memastikan supaya fitur-fitur yang terdapat dalam aplikasi berjalan dengan baik dan bebas dari *error*. Oleh karena itu, pengujian pada aplikasi sangat dibutuhkan agar aplikasi yang dibuat menjadi lebih baik lagi.

## ***Login***

Berikut merupakan tabel pengujian dari proses *login*.

Tabel 4. 14 Tabel Pengujian Proses Login

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | Memasukkan *username* yang salah saat *login*. | Kembali ke halaman *login* beserta notifikasi yang mengatakan bahwa *username* salah. |  |
| 2 | Memasukkan *username* yang benar, namun passwordnya salah | Kembali ke halaman *login* beserta notifikasi yang mengatakan bahwa *password* salah. |  |
| 3 | Memasukkan *username* dan *password* terdaftar yang benar. | Masuk ke halaman *dashboard* sesuai dengan *role*. |  |
| 4 | Terdapat *form* yang kosong saat mengisi data untuk *login*. | Kembali ke halaman *login* beserta notifikasi yang mengatakan bahwa tidak boleh ada data yang kosong. |  |

## **Konfirmasi Akun**

Berikut merupakan tabel pengujian dari proses saat *user* konfirmasi akun yang dapat dilakukan oleh semua pengguna.

Tabel 4. 15 Tabel Pengujian Proses Konfirmasi Akun

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | Memasukkan nomor induk yang salah/tak terdaftar saat konfirmasi akun. | Kembali lagi ke halaman awal konfirmasi akun dengan notifikasi bahwa nomor induk tak terdaftar di sistem. |  |
| 2 | Memasukkan nomor induk yang benar, tetapi kode konfirmasi salah/tidak terdaftar . | Kembali lagi ke halaman awal konfirmasi akun dengan notifikasi bahwa kode konfirmasi tak terdaftar di sistem. |  |
| 3 | Memasukkan nomor induk dan kode konfirmasi yang benar. | *User* akan diarahkan ke halaman yang berisi *form* input untuk membuat *username* dan passwordnya sendiri. |  |
| 4 | Memasukkan *username* yang telah dipakai oleh *user* lain. | Kembali lagi ke halaman awal konfirmasi akun dengan notifikasi bahwa *username* telah terpakai oleh *user* lain. |  |
| 5 | *User* memasukkan konfirmasi *password* yang berbeda dengan *password* yang ada di atasnya. | Kembali lagi ke halaman awal konfirmasi akun dengan notifikasi bahwa *password* tidak match. |  |
| 6 | Memasukkan *username* yang belum terpakai, serta *password* dan konfirmasi *password* yang sudah *match*. | *User* akan diarahkan ke halaman *login* dengan notifikasi bahwa akun berhasil dibuat dan *user* diharapkan untuk *login*. |  |
| 7 | Saat pengisian data, terdapat *form* *input* yang kosong. | Kembali lagi ke halaman awal konfirmasi akun dengan notifikasi bahwa data tidak boleh kosong. |  |

## **Pendataan Buku**

Berikut merupakan tabel pengujian dari proses pendataan buku yang diakses oleh pustakawan.

Tabel 4. 16 Tabel Pengujian Proses Pendataan Buku

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | Saat memasukkan data yang dibutuhkan, pustakawan memasukkan id penerbit yang tak terdaftar di sistem. | Penambahan data buku akan gagal serta terdapat notifikasi bahwa id penerbit tidak terdaftar di sistem. |  |
| 2 | Saat memasukkan data yang dibutuhkan, pustakawan memasukkan id kategori yang tak terdaftar di sistem. | Penambahan data buku akan gagal serta terdapat notifikasi bahwa id kategori tidak terdaftar di sistem. |  |
| 3 | Pustakawan mengisi semua data dengan benar, serta id penerbit dan id kategorinya sudah terdaftar di sistem. | Penambahan data buku berhasil serta terdapat notifikasi bahwa data buku berhasil ditambahkan. |  |
| 4 | Saat memasukkan data, terdapat data yang kosong. | Penambahan data buku gagal serta terdapat notifikasi bahwa data tidak boleh kosong. |  |
| 5 | Pustakawan mengubah beberapa data dari suatu buku. | Pengubahan data berhasil serta terdapat notifikasi bahwa data berhasil diubah. |  |
| 6 | Pustakawan menghapus data suatu buku. | Penghapusan data berhasil serta terdapat notifikasi bahwa data berhasil dihapus. |  |
| 7 | *Import* data buku dari *file* excel. | Penambahan data berhasil serta terdapat notifikasi bahwa data berhasil ditambah. |  |
| 8 | *Import* data buku dari suatu *file*, tetapi bukan excel. | Penambahan data gagal serta terdapat notifikasi bahwa *file* yang diizinkan hanya *file* excel. |  |

## **Pendataan Anggota**

Berikut merupakan tabel pengujian dari proses pendataan anggota. Halaman ini hanya dapat diakses oleh pustakawan.

Tabel 4. 17 Tabel Pengujian Proses Pendataan Anggota

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | Pustakawan memasukkan semua data yang dibutuhkan di *form* pengisian data. | Penambahan data berhasil serta terdapat notifikasi bahwa data berhasil ditambah. |  |
| 2 | Terdapat data yang kosong saat pustakawan mengisi data. | Penambahan data gagal serta terdapat notifikasi bahwa data tidak boleh kosong. |  |
| 3 | Pustakawan menghapus data seorang anggota. | Penghapusan data berhasil serta terdapat notifikasi bahwa data berhasil dihapus. |  |
| 4 | *Import* data anggota dari *file* excel. | Penambahan data berhasil serta terdapat notifikasi bahwa data berhasil ditambah. |  |
| 5 | *Import* data anggota dari suatu *file*, tetapi bukan excel. | Penambahan data gagal serta terdapat notifikasi bahwa *file* yang diizinkan hanya *file* excel. |  |
| 6 | Pustakawan mencetak kartu member untuk seorang anggota. | Kartu anggota berhasil dikonvert menjadi *file* pdf yang siap untuk diprint. |  |

## **Pendataan *Ebook***

Berikut merupakan tabel pengujian dari proses pendataan *ebook*.

Tabel 4. 18 Tabel Pengujian Proses Pendataan Ebook

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | Saat memasukkan data yang dibutuhkan, pustakawan memasukkan id penerbit yang tak terdaftar di sistem. | Penambahan data ebook akan gagal serta terdapat notifikasi bahwa id penerbit tidak terdaftar di sistem. |  |
| 2 | Saat memasukkan data yang dibutuhkan, pustakawan memasukkan id kategori yang tak terdaftar di sistem. | Penambahan data ebook akan gagal serta terdapat notifikasi bahwa id kategori tidak terdaftar di sistem. |  |
| 3 | Pustakawan mengisi semua data dengan benar, serta id penerbit dan id kategorinya sudah terdaftar di sistem. | Penambahan data *ebook* berhasil serta terdapat notifikasi bahwa data *ebook* berhasil ditambahkan. |  |
| 4 | Saat memasukkan data, terdapat data yang kosong. | Penambahan data *ebook* gagal serta terdapat notifikasi bahwa data tidak boleh kosong. |  |
| 5 | Pustakawan mengubah beberapa data dari suatu *ebook*. | Pengubahan data berhasil serta terdapat notifikasi bahwa data berhasil diubah. |  |
| 6 | Pustakawan menghapus data suatu *ebook*. | Penghapusan data berhasil serta terdapat notifikasi bahwa data berhasil dihapus. |  |
| 7 | Pustakawan membuka *file* *ebook*. | Pustakawan akan diarahkan ke halaman yang berisi *ebook* tersebut. |  |
| 8 | *Import* data *ebook* dari *file* excel. | Penambahan data berhasil serta terdapat notifikasi bahwa data berhasil ditambah. |  |
| 9 | *Import* data *ebook* dari suatu *file*, tetapi bukan excel. | Penambahan data gagal serta terdapat notifikasi bahwa *file* yang diizinkan hanya *file* excel. |  |

## **Pendataan Pustakawan/*Admin***

Berikut merupakan tabel pengujian dari proses pendataan pustakawan/*admin* yang hanya dapat diakses oleh *admin*.

Tabel 4. 19 Tabel Pengujian Proses Pendataan Pustakawan/Admin

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | *Admin* memasukkan semua data yang dibutuhkan di *form* pengisian data. | Penambahan data berhasil serta terdapat notifikasi bahwa data berhasil ditambah. |  |
| 2 | Terdapat data yang kosong saat *admin* mengisi data. | Penambahan data gagal serta terdapat notifikasi bahwa data tidak boleh kosong. |  |
| 3 | *Admin* menghapus data seorang pustakawan. | Penghapusan data berhasil serta terdapat notifikasi bahwa data berhasil dihapus. |  |
| 4 | *Import* data pustakawan dari *file* excel. | Penambahan data berhasil serta terdapat notifikasi bahwa data berhasil ditambah. |  |
| 5 | *Import* data pustakawan dari suatu *file*, tetapi bukan excel. | Penambahan data gagal serta terdapat notifikasi bahwa *file* yang diizinkan hanya *file* excel. |  |

## **Pendataan Penerbit**

Berikut merupakan tabel pengujian dari proses pendataan penerbit.

Tabel 4. 20 Tabel Pengujian Proses Pendataan Penerbit

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | Pustakawan memasukkan semua data yang dibutuhkan di *form* pengisian data. | Penambahan data berhasil serta terdapat notifikasi bahwa data berhasil ditambah. |  |
| 2 | Terdapat data yang kosong saat pustakawan mengisi data. | Penambahan data gagal serta terdapat notifikasi bahwa data tidak boleh kosong. |  |
| 3 | Pustakawan mengubah data dari sebuah penerbit. | Pengubahan data berhasil serta terdapat notifikasi bahwa data berhasil diubah. |  |
| 4 | Pustakawan menghapus data sebuah penerbit. | Penghapusan data berhasil serta terdapat notifikasi bahwa data berhasil dihapus. |  |
| 5 | *Import* data penerbit dari *file* excel. | Penambahan data berhasil serta terdapat notifikasi bahwa data berhasil ditambah. |  |
| 6 | *Import* data penerbit dari suatu *file*, tetapi bukan excel. | Penambahan data gagal serta terdapat notifikasi bahwa *file* yang diizinkan hanya *file* excel. |  |

## **Pendataan Kategori**

Berikut merupakan tabel pengujian dari proses pendataan kategori.

Tabel 4. 21 Tabel Pengujian Proses Pendataan Kategori

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | Pustakawan memasukkan semua data yang dibutuhkan di *form* pengisian data. | Penambahan data berhasil serta terdapat notifikasi bahwa data berhasil ditambah. |  |
| 2 | Terdapat data yang kosong saat pustakawan mengisi data. | Penambahan data gagal serta terdapat notifikasi bahwa data tidak boleh kosong. |  |
| 3 | Pustakawan mengubah data dari sebuah kategori. | Pengubahan data berhasil serta terdapat notifikasi bahwa data berhasil diubah. |  |
| 4 | Pustakawan menghapus data sebuah kategori. | Penghapusan data berhasil serta terdapat notifikasi bahwa data berhasil dihapus. |  |
| 5 | *Import* data kategori dari *file* excel. | Penambahan data berhasil serta terdapat notifikasi bahwa data berhasil ditambah. |  |
| 6 | *Import* data kategori dari suatu *file*, tetapi bukan excel. | Penambahan data gagal serta terdapat notifikasi bahwa *file* yang diizinkan hanya *file* excel. |  |

## **Pendataan Pengajuan Penggunaan *Ebook***

Berikut merupakan tabel pengujian dari proses pendataan pengajuan penggunaan *ebook*. Data pengajuan merupakan data yang berasal saat anggota mengirim permintaan untuk menggunakan suatu *ebook*.

Tabel 4. 22 Tabel Pengujian Pendataan Pengajuan Penggunaan Ebook

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | Menerima ajuan penggunaan *ebook* dari anggota. | Ajuan diterima, *email* pemberitahuan terkirim serta anggota dapat menggunakan *ebook* tersebut. |  |
| 2 | Menolak ajuan penggunaan *ebook* dari anggota. | Ajuan ditolak, *email* pemberitahuan terkirim serta anggota tidak dapat menggunakan *ebook* tersebut. |  |
| 3 | Menghapus data penggunaan *ebook* yang telah kadaluarsa. | Data berhasil terhapus. |  |
| 4 | Menghapus data penggunaan *ebook* yang telah ditolak. | Data berhasil terhapus. |  |

## **Pendataan Transaksi**

Berikut merupakan tabel pengujian dari proses pendataan transaksi.

Tabel 4. 23 Tabel Pengujian Proses Pendataan Transaksi

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | Pustakawan mengisi data yang dibutuhkan untuk menambah data transaksi, namun nomor induk yang dimasukkan salah/tak terdaftar di sistem. | Penambahan data akan gagal serta terdapat notifikasi bahwa nomor induk tidak terdaftar di sistem. |  |
| 2 | Pustakawan mengisi data yang dibutuhkan untuk menambah data transaksi, namun id buku yang dimasukkan salah/tak terdaftar di sistem. | Penambahan data akan gagal serta terdapat notifikasi bahwa buku tidak terdaftar di sistem. |  |
| 3 | Pustakawan mengisi data yang dibutuhkan untuk menambah data transaksi, serta nomor induk dan id buku sudah terdaftar di sistem. | Penambahan data berhasil serta terdapat notifikasi bahwa data berhasil ditambahkan. |  |
| 4 | Pustakawan mengisi data yang dibutuhkan untuk menambah data transaksi, terdapat data yang kosong. | Penambahan data akan gagal serta terdapat notifikasi bahwa data tidak boleh ada yang kosong. |  |
| 5 | Pustakawan mengubah data dari sebuah transaksi. | Pengubahan data berhasil serta terdapat notifikasi bahwa data berhasil diubah. |  |
| 6 | Anggota mengembalikan buku dan pustakawan mendatanya. | Data transaksi ditandai telah dikembalikan dan sistem menghitung denda yang harus dibayar anggota jika terlambat mengembalikan. |  |
| 7 | Pustakawan mengirim peringatan kepada anggota yang telah melewati batas peminjaman namun belum mengembalikan. | Pemberitahuan akan terkirim kepada anggota melalui *email*. |  |

## **Laporan**

Berikut merupakan tabel pengujian dari laporan transaksi. Halaman ini dapat diakses oleh *admin* maupun pustakawan.

Tabel 4. 24 Tabel Pengujian Proses Pembuatan Laporan

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | Melihat daftar keseluruhan laporan. | Menanmpilkan semua data laporan pengembalian, menampilkan informasi anggota serta buku yang dipinjam. |  |
| 2 | Melihat daftar laporan keterlambatan. | Menanmpilkan semua data laporan pengembalian, menampilkan informasi anggota, buku yang dipinjam serta denda yang didapat. |  |
| 3 | Cetak data menjadi *file* excel. | Otomatis akan mendownload *file* excel yang berisi data-data yang ingin dicetak. |  |
| 4 | Cetak data laporan menjadi *file* pdf. | Otomatis akan mendownload *file* pdf yang berisi data laporan. |  |

## **Konfigurasi**

Berikut merupakan tabel pengujian dari proses perubahan konfigurasi.

Tabel 4. 25 Tabel Pengujian Proses Konfigurasi

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | Admin/pustakawan mengubah isi konfigurasi yang tersedia. | Konfigurasi akan langsung diterapkan ke aplikasi. |  |
| 2 | Saat mengisi data, terdapat data yang dikosongkan. | Pengubahan konfigurasi gagal serta terdapat notifikasi bahwa data tak boleh kosong. |  |
| 3 | Saat mengisi data, terdapat konfigurasi yang melebihi batas seharusnya. | Pengubahan konfigurasi gagal serta terdapat notifikasi bahwa data tak boleh melebih batas yang telah ditentukan. |  |
| 4 | Mengubah gambar-gambar yang ada di pengaturan *gallery*. | Gambar-gambar akan otomatis terganti serta dapat langsung dilihat perubahannya di *landing* *page*. |  |
| 5 | Mengubah gambar *background* yang ada di *dashboard* anggota. | Gambar akan otomatis terganti serta dapat langsung dilihat perubahannya di *dashboatd* anggota. |  |

## **Pengajuan *Ebook***

Berikut merupakan tabel pengujian dari proses pengajuan *ebook*.

Tabel 4. 26 Tabel Pengujian Proses Pengajuan Ebook

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | Member mengajukan penggunaan sebuah *ebook*. | Pengajuan berhasil dilakukan dan data akan terlihat di halaman pengajuan pustakawan. |  |
| 2 | Member mengajukan penggunaan sebuah *ebook* tetapi sebelumnya *ebook* tersebut sudah pernah diajukan. | Pengajuan gagal dilakukan serta terdapat notifikasi bahwa *ebook* tersebut sudah pernah diajukan sebelumnya. |  |

## **Keamanan**

Berikut merupakan tabel pengujian dari keamanan website.

Tabel 4. 27 Tabel Pengujian Keamanan

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | Saat pengguna berada di halaman *landing* *page* lalu mencoba untuk mengakses halaman yang perlu *authetication* tanpa *login*. | Pengguna akan diarahkan ke halaman login karena halaman yang dituju perlu adanya *authentication* sebagai *user* yang terdaftar dalam sistem. |  |
| 2 | Setelah pengguna keluar aplikasi (*logout*) lalu menekan tombol kembali yang ada di *browser*. | Pengguna tidak akan diarahkan kemana pun karena setelah *logout* pengguna tidak diizinkan mengakses halaman aplikasi. |  |
| 3 | Mengakses halaman yang bukan rolenya. | Pengguna akan diarahkan ke halaman *dashboard* sesuai rolenya, karena dilarang untuk mengakses halaman yang bukan untuk rolenya. |  |

## **Mengganti *Password***

Berikut merupakan tabel pengujian saat user mengganti *password* milikinya.

Tabel 4. 28 Tabel Pengujian Proses Mengganti Password

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kasus** | **Hasil yang diharapkan** | **Status** |
| 1 | *Password* lama yang diisikan tidak sesuai dengan *password* yang terdaftar di sistem. | Proses mengganti *password* gagal serta terdapat notifikasi bahwa *password* yang lama tidak terdaftar di sistem. |  |
| 2 | Saat mengisi, *password* yang diisikan kurang dari 8 karakter. | Proses mengganti *password* gagal serta terdapat notifikasi bahwa *password* harus lebih dari 8 karakter. |  |
| 3 | Konfirmasi *password* tidak cocok dengan *password* baru yang dibuat. | Proses mengganti *password* gagal serta terdapat notifikasi bahwa *password* tidak cocok. |  |
| 4 | *User* tidak menandai *checkbox* persetujuan bahwa *user* tersebut benar-benar ingin mengganti passwordnya. | Proses mengganti *password* gagal serta terdapat notifikasi bahwa *checkbox* persetujuan harus ditandai. |  |
| 5 | Data yang dibutuhkan diisi lengkap, *password* baru sudah match dan *checkbox* telah ditandai. | Proses mengganti *password* berhasil, data di *database* akan ter-update serta terdapat notifikasi bahwa *password* berhasil diganti. |  |
| 6 | Saat mengisi data, terdapat data yang tak terisi. | Proses mengganti *password* gagal serta terdapat notifikasi bahwa tidak boleh ada data yang kosong. |  |

# **BAB V PENUTUP**

## **Kesimpulan**

Beberapa kesimpulan yang dapat diambil terkait pembuatan *web* aplikasi O’Library ini adalah:

1. O’Library ini mengurusi pendataan di bidang perbukuan yang ada di perpustakaan sekolah.
2. Penggunaan O’library ini ditujukan kepada 3 *user*, yaitu anggota, *admin* dan pustakawan.
3. O’Library akan memudahkan pengurus perpustakaan untuk mendata koleksi buku atau *ebook* yang tersedia di perpustakaan.
4. O’Library akan memudahkan anggota perpustakaan untuk melihat daftar buku/*ebook* yang tersedia di perpustakaan.
5. O’Library akan memudahkan pengguna untuk mengakses buku digital (*ebook*) secara gratis.
6. O’Library akan memudahkan dalam urusan pembuatan laporan, karena laporan dapat dicetak dengan mudah dalam bentuk *file* digital.

## **Saran**

## Berikut beberapa saran yang dapat diberikan *developer* agar *web* aplikasi yang dibuat dapat memiliki kualitas dan harga jual yang lebih tinggi lagi.

## Sistem penginputan data dilakukan secara otomatis dengan menggunakan alat bantu, seperti *scan bar code*.

## Proses peminjaman buku yang dilakukan anggota dilakukan secara otomatis, seperti dengan menggunakan *scan* QR yang tersedia di kartu anggota agar proses semakin cepat.

# **DAFTAR PUSTAKA**

Ahmad, Daniyal. (2017). *Webpack apa itu*?. Diakses pada 23 September 2020, dari <https://medium.com/codelabs-unikom/webpack-apa-itu-b6272cd16322>.

Blogger, Iqram. (2013). *Flow Map-Definisi, Simbol dan Penggunannya*. Diakses pada 3 Februari 2021, dari http://andiiqram.blogspot.com/2013/02/flow-map-definisi-simbol-dan.html.

C, Ariata. (2019). *Apa itu AJAX dan Bagaimana Cara Kerjanya?.* Diakses pada 16 Januari 2021, dari <https://www.hostinger.co.id/tutorial/apa-itu-ajax/>.

C, Ariata. (2019). *Apa itu Jquery?.* Diakses pada 23 September 2020, dari https://www.hostinger.co.id/tutorial/apa-itu-jquery/.

Dosenpendidikan. (2020). *Pengertian Buku*. Diakses pada 23 September 2020, dari <https://www.dosenpendidikan.co.id/pengertian-buku/>.

Ibeng, Parta. (2021). *Pengertian Data, Fungsi Data, dan Macam Jenisnya*. Diakses pada 15 Januari 2021, dari https://pendidikan.co.id/pengertian-data/.

MateriDosen. (2017). *Use Case Diagram, Lengkap Studi Kasus dan Contoh Use Case*. Diakses pada 3 Februari 2021, dari <http://www.materidosen.com/2017/04/use-case-diagram-lengkap-studi-kasus.html>.

Muhardian, Ahmad. (2019). *Tutorial Dasar SASS untuk Pemula*. Diakses pada 23 September 2020, dari <https://www.petanikode.com/sass-untuk-pemula/>.

Nabila, Gina. (2018). *Laporan Praktek Kerja Industri - Aplikasi Marketing Project Desk (Madesk) Menggunakan Laravel Berbasis Website*.

Putra, Reno. (2017). *Kenalan dengan ES6 Javascript (Introduction, variable, arrow function) #part1*. Diakses pada 23 September 2020, dari https://medium.com/@renopp/kenalan-dengan-es6-javascript-introduction-variable-arrow-function-part1-6bd5c148473b.

Team, Dewaweb. (2020). *Pengertian, Fungsi, Jenis, dan Manfaat Website yang Perlu Kamu Ketahui*. Diakses pada 23 September 2020, dari <https://www.dewaweb.com/blog/pengertian-website/>.

Team, Dicoding. (2020). *Ngoding Web Dinamis atau Statis*. Diakses pada 23 September 2020, dari <https://www.dicoding.com/blog/ngoding-web-dinamis-atau-statis/>.

Wikipedia. (2020). *Aplikasi Web*. Diakses pada 23 September 2020, dari <https://id.wikipedia.org/wiki/Aplikasi_web>.

Wikipedia. (2021). *Perpustakaan*. Diakses pada 23 September 2020, dari https://id.wikipedia.org/wiki/Perpustakaan.

Wikipedia. (2020). *Pangkalan Data*. Diakses pada 23 September 2020, dari <https://id.wikipedia.org/wiki/Pangkalan_data>.

Wikipedia. (2019). *Buku Elektronik*. Diakses pada 23 September 2020, dari https://id.wikipedia.org/wiki/Buku\_elektronik.