**RANCANG BANGUN APLIKASI PENGELOLAAN DAN PERAWATAN LABORATORIUM POLITEKNIK NEGERI INDRAMAYU**

**BERBASIS WEBSITE**

**Tugas Akhir**

diajukan untuk memenuhi sebagian persyaratan mencapai

gelar Ahli Madya pada jenjang Diploma III

Jurusan Teknik Informatika

****

Oleh :

**JENIA ADELLIA PUSPITA**

**NIM. 1603102**

**JURUSAN TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI INDRAMAYU**

**2019**

**HALAMAN PENGESAHAN**

Tugas Akhir ini diajukan oleh :

Nama : Jenia Adellia Puspita

NIM : 1603102

Program Studi : Teknik Informatika

Judul : Rancang Bangun Aplikasi Pengelolaan dan Perawatan

Laboratorium Politeknik Negeri Indramayu

Pembimbing : 1. A. [Lubis Ghozali, S.Kom., M.Kom](http://gaweweb.com) ........................

2. Adi Suheryadi, S.ST., M.Kom ........................

Telah berhasil dipertahankan dihadapan dewan penguji pada tanggal 26 Agustus 2019 dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Ahli Madya Program Studi Teknik Informatika, Jurusan Teknik Informatika, Politeknik Indramayu.

DEWAN PENGUJI

|  |  |  |  |
| --- | --- | --- | --- |
| Nama | Jabatan | Tandatangan | Tanggal |
| 1. .................................. | Ketua Penguji | .......................... | 26 Agustus 2019 |
| 1. .................................. | Sekretaris Penguji | .......................... | 26 Agustus 2019 |
| 1. .................................. | Anggota Penguji | .......................... | 26 Agustus 2019 |

Indramayu, 26 Agustus 2019

Ketua Jurusan Teknik Informatika

**Munengsih Sari Bunga, S.Kom., M.Eng**

**NIK. 08098543­­**

**PERNYATAAN KEASLIAN**

Saya menyatakan dengan sesungguhnya bahwa Tugas Akhir ini adalah asli hasil karya saya sendiri, dan sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis atau dipublikasikan oleh orang lain, kecuali yang secara tertulis disebutkan sumbernya dalam naskah dan dalam daftar pustaka.

Indramayu, 26 Agustus 2019

Yang menyatakan,

**Jenia Adellia Puspita**

NIM. 1603102

**ABSTRAK**

Dalam pengisian pemakaian ruang laboratorium sehari-hari, kurangnya perhatian dari mahasiswa menggunakan kertas dan terbatas kesediaan mengakibatkan laporan pemakaian ruang laboratorium sehari-hari terbengkalai. Padahal data pemakaian ruang laboratorium dibutuhkan untuk laporan dalam penunjang sarana dan prasana agar lebih baik kedepannya, dan perawatan pada ruang laboratorium dengan keterlibatan mahasiswa yang kompensasi masih kurang, serta proses peminjaman barang hingga pengembalian barang yang baik. Dari permasalahan tersebut memunculkan gagasan untuk membuat suatu aplikasi berbasis *Website*, yang di dalamnya dapat memberikan pengelolaan dan perawatan pada ruang laboratorium. Pengelolaan terkait dengan proses kegiatan yang ada di ruang laboratorium dari *logbook* laboratorium, inventaris alat dan barang yang ada di ruang laboratorium sebagai penunjang dalam belajar dan mengajar di kampus, dan peminjaman barang pada ruang laboratorium untuk mahasiswa. Sedangkan perawatan ruang laboratorium bentuk upaya membersihkan ruang laboratorium yang terhubung dengan mahasiswa yang kompensasi. sehingga permasalahan yang ada membuat sebuah penelitian yang diambil dengan judul “Rancang Bangun Aplikasi Pengelolaan dan Perawatan Laboratorium Politeknik Negeri Indramayu”.

Kata Kunci: Pengelolaan, Perawatan, Laboratorium.

***ABSTRACT***

*In filling the use of laboratory space everyday, lack of attention from students using paper and limited willingness to result in reports of daily use of laboratory space abandoned. Though data on laboratory space usage is needed for reports in supporting facilities and infrastructures to be better in the future, and maintenance in laboratory space with the involvement of students whose compensation is still minimal, as well as the process of borrowing goods to return good items. Of these problems raises the idea of ​​creating a Website-based application, which can provide management and maintenance in the laboratory space. Management is related to the activity processes in the laboratory room of the laboratory logbook, inventory of tools and goods in the laboratory room as support in learning and teaching on campus, and borrowing goods in the laboratory room for students. While laboratory room care is an effort to clean up the laboratory space that is connected with students who are compensated. so that the existing problems make a study taken with the title "Designing Application for Management and Maintenance of Indramayu State Polytechnic Laboratory".*

*Keyword: Management, Maintenance, Laboratory.*

**KATA PENGANTAR**

*Assalamu’alaikum Warahmatullahi Wabarakatuh.*

Dengan mengucap puji kami panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik. Adapun judul tugas akhir ini yaitu “Rancang Bangun Aplikasi Pengelolaan dan Perawatan Laboratorium Politeknik Negeri Indramayu Berbasis *Website*”.

Tujuan penulisan tugas akhir ini dibuat sebagai salah satu syarat kelulusan Diploma tiga Teknik Informatika Politeknik Negeri Indramayu. Sebagai bahan penulisan diambil berdasarkan penelitian, observasi dan beberapa sumber yang turut mendukung dalam penulisan ini. Penulis menyadari bahwa tanpa bimbingan dan dorongan dari semua pihak, maka tugas akhir ini tidak dapat terselesaikan dengan baik. Oleh karena itu penulis ingin menyampaikan ucapan terimakasih kepada:

1. Bapak Casiman Sukardi, ST., MT selaku Direktur Politeknik Negeri Indramayu.
2. Ibu Munengsih Sari Bunga, S.Kom., M.Eng selaku Ketua Jurusan Teknik Informatika.
3. Bapak A. [Lubis Ghozali, S.Kom., M.Kom](http://gaweweb.com) selaku pembimbing I.
4. Bapak Adi Suheryadi, S.ST., M.Kom selaku pembimbing II.
5. Semua Dosen Teknik Informatika yang telah memberi banyak ilmu dan pengetahuan.
6. Teman-teman Teknik Informatika angkatan 2016 yang telah banyak membantu dalam penyelesaian tugas akhir ini.
7. Kedua orang tua yang selalu memberi dukungan penuh dan menjadi motivasi semangat untuk dapat menyelesaikan tugas akhir ini.
8. Teman kosan yang selalu ada menjadi tempat untuk berbagi.
9. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang terlibat dalam penyelesaian tugas akhir ini.

Penulis telah berusaha semaksimal mungkin dalam menyelesaikan laporan tugas akhir ini dengan baik, tetapi manusia hanyalah salah dan keliru karena kesempurnaan hanya milik Allah SWT. Apabila terdapat kesalahan dalam penulisan tugas akhir ini penulis berterimakasih dengan senang hati pembaca memberikan kritik dan saran yang membangun tentunya. Kritik dan saran yang membangun akan menjadi bekal penulis agar lebih baik dalam penulisan. Semoga laporan tugas akhir ini dapat bermanfaat bagi pembacanya.

*Wassalamu’alaikum Warahmatullahi Wabarakatuh.*

Indramayu, 26 Agustus 2019

Penulis

**DAFTAR ISI**

**DAFTAR GAMBAR**

**DAFTAR LAMPIRAN**

**BAB I**

**PENDAHULUAN**

1. **Latar Belakang Masalah**

Laboratorium komputer adalah tempat riset, eksperimen, pengukuran ataupun pelatihan ilmiah yang berhubungan dengan ilmu komputer dan dalam ruang tersebut memiliki beberapa komputer dalam satu jaringan untuk penggunaan sesuai yang dibutuhkan. Di dalam laboratorium terdapat kegiatan praktikum, penelitian, dan bentuk kegiatan lainnya yang digunakan untuk proses belajar mengajar. (Nanik, dkk. 2012).

Pada laboratorium Jurusan terdapat alat dan bahan yang menunjang sebagai sarana belajar mengajar, di Laboratorium Politeknik Negeri Indramayu saat ini sudah menggunakan komputasi sebagai sarana untuk kegiatan yang berkaitan dengan pengelolaan dan perawatan, namun masih sebatas untuk mengetik data inventaris laboratorium (Meja, Kursi, AC, CCTV, CPU, LCD Monitor, Sound, LCD proyektor, papan tulis dan alat-alat praktikum) kegiatan tersebut tidak berjalan dengan baik dan program yang digunakan masih terbatas untuk membuat laporan yang berhubungan dengan pengolahan data maupun inventaris sehingga mengakibatkan keterlambatan, maka diperlukan sistem yang cakupan luas dengan menggunakan program untuk mengolah data yang meliputi pengelolaan data inventaris alat dan bahan, data laporan kegiatan, perawatan laboratorium, dan informasi kompensasi mahasiswa yang berkaitan dengan kebersihan pada ruang laboratorium.

Dengan adanya sistem baru akan mempermudah informasi laboratorium yang cepat dan tepat dalam dikembangkan sistem pengolahan data agar lebih baik. Karena itu harus diperlukan sistem yang dapat melakukan tugas-tugas tersebut secara cepat dan akurat yaitu sistem komputer yang mempunyai keunggulan dalam pemrosesan data baik yang harus diolah, disimpan, disajikan, dianalisa ketepatannya dan kecepatan hasil yang dicapai.

1. **Rumusan Masalah**

Berdasarkan latar belakang yang telah dijelaskan di atas, maka penulis mengambil beberapa rumusan masalah yang terkait dengan pengelolaan dan perawatan pada lab komputer antara lain sebagai berikut :

1. Bagaimana merancang sistem pengelolaan yang terstruktur untuk kebutuhan dan hanya terfokuskan pada ruang laboratorium di Politeknik Negeri Indramayu?
2. Bagaimana membuat aplikasi pengelolaan dan perawatan laboratorium?
3. Bagaimana mengimplementasikan pengisian *logbook* komputer di ruang laboratorium untuk mahasiswa?
4. **Batasan Masalah**

Permasalahan yang akan diselesaikan dalam penelitian ini agar terarah dan mencakup serta sesuai dengan judul Tugas Akhir yang diajukan, dibatasi oleh ruang lingkup sebagai berikut :

1. Mencakup mulai dari pengelolaan pemakaian komputer laboratorium dan data peminjaman di *Maintenance,* dan data kompensasi.
2. Aplikasi ini menggunakan java sebagai dekstop, Yii Framework sebagai web, dan mysql sebagai database.
3. **Tujuan penelitian**

Tujuan yang akan dicapai dalam penelitian tugas akhir ini adalah :

1. Mengelola data pemakaian komputer pada laboratorium secara terpusat.
2. Peningkatan pelayanan agar data pemakaian komputer terstruktur untuk pelaporan.
3. Mengelola data peminjaman barang sesuai dengan jumlah barang yang ada di *maintenance*.
4. **Manfaat penelitian**

Berdasarkan pembahasan diatas terdapat manfaat dari penelitian tugas akhir ini adalah :

1. Memudahkan pengelolaan pemakaian komputer di ruang laboratorium untuk data laporan.
2. Memudahkan peminjaman barang utnuk mahasiswa dalam menggunakan aplikasi ini.
3. **Sistematika Penulisan**

Dalam sistematika penulisan laporan tugas akhir ini dibagi menjadi beberapa bagian bab, di mana pada setiap bagian membuat pembahasan yang berbeda namun masih memiliki keterkaitan satu sama lain. Untuk memudahkan penulisan laporan tugas akhir ini penulis menguraikan dan mengurutkan bab-babnya secara sistematis. Adapun urutan penulisannya sebagai berikut:

**BAB I PENDAHULUAN**

Bab pendahuluan berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

**BAB II LANDASAN TEORI**

Bab ini menguraikan tentang teori dasar yang berkaitan secara langsung dengan masalah yang diteliti. Diusahakan mengambil referensi dari buku atau sumber lainnya. Pada bagian ini juga bisa diuraikan penelitian-penelitian lain yang terkait dengan penelitian.

**BAB III METODE PENELITIAN**

Bagian ini berisi penjelasan tentang tahapan dan metode penelitian yang ditempuh untuk mencapai tujuan yang ditetapkan. Diagram alur (flow chart) harus dibuat untuk menjelaskan tahapan tersebut.

**BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini dibahas hasil-hasil dari tahapan penelitian, dari tahap analisis, desain, implementasi desain, hasil pengujian dan implementasinya, berupa penjelasan teoritik, baik secara kualitatif, kuantitatif, atau secara statistik. Disamping itu, sebaiknya hasil penelitian juga dibandingkan dengan hasil penelitian terdahulu yang sejenis atau keadaan sebelumnya.

**BAB V PENUTUP**

Berisi kesimpulan dan saran. Kesimpulan akan mengemukakan kembali masalah penelitian (menjawab rumusan masalah), bukti-bukti yang dihasilkan dan akhirnya menarik kesimpulan apakah penelitian atau kegiatan yang dilakukan sudah memberi manfaat nyata bagi objek penelitian.

**BAB II**

**LANDASAN TEORI**

1. **Laboratorium**

Laboratorium (disingkat lab) adalah tempat riset ilmiah, eksperimen, pengukuran ataupun pelatihan ilmiah dilakukan. Laboratorium biasanya dibuat untuk memungkinkan dilakukannya kegiatan-kegiatan tersebut secara terkendali. Laboratorium ilmiah biasanya dibedakan menurut disiplin ilmunya, misalnya laboratorium fisika, laboratorium kimia, laboratorium biokimia, laboratorium komputer, dan laboratorium bahasa. (Garniardi, dkk. 2009).

1. **Website**

Website adalah suatu kumpulan halaman yang menampilkan berbagai macam informasi teks, data, gambar diam ataupun bergerak, data animasi, suara, video maupun gabungan dari semuanya, baik itu yang bersifat statis maupun yang dinamis, yang dimana membentuk satu rangkaian bangunan yang saling berkaitan dimana masing-masing dihubungkan dengan jaringan halaman atau hyperlink. (Sora, 2014).

Definisi website lainnya adalah kumpulan dari berbagai macam halaman situs, yang terangkum didalam sebuah domain atau juga subdomain, yang lebih tempatnya berada di dalam WWW *(World Wide Web*) yang tentunya terdapat di dalam Internet. Halaman website biasanya berupa dokumen yang ditulis dalam format *Hyper Text Markup Language* (HTML), yang bisa diakses melalui HTTP, HTTP adalah suatu protokol yang menyampaikan berbagai informasi dari server website untuk ditampilkan kepada para user atau pemakai melalui web browser (Sora, 2014).

1. **Java**

Java merupakan teknologi yaitu mencakup sebagai bahasa pemograman yang memiliki sintaks dan aturan pemograman tersendiri, juga memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan program yang ditulis dengan bahasa pemograman java. Java merupakan bahasa pemrograman yang berorientasi objek yang diciptakan oleh Sun Microsystem pada tahun 1995. Java dapat membuat seluruh bentuk aplikasi, desktop, web dan lainnya sebagaimana dibuat dengan menggunakan bahasa pemrograman yang lain. Java dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan java tidak hanya terfokus pada suatu sistemoperasi, tetapi dikembangkan untukberbagai sistem operasi dan bersifat *open source*. (Hariyanto, 2004).

1. **Yii Framework**

Yii adalah kerangka kerja pemrograman web umum, yang berarti bahwa hal itu dapat digunakan untuk mengembangkan semua jenis aplikasi Web yang menggunakan PHP. Karena arsitektur berbasis komponen dan dukungan caching yang canggih, Yii sangat cocok untuk mengembangkan aplikasi skala besar seperti portal, forum, sistem manajemen konten (CMS), proyek ecommerce, layanan web REST, dan sebagainya.(Brandt. dkk, 2008).

Gambar 2. 1 Yii framework (Eshill, 2008)

1. **GitHub**

GitHub adalah [layanan penanganan web](https://id.wikipedia.org/wiki/Layanan_penginangan_web) bersama untuk proyek pengembangan [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) yang menggunakan sistem [pengontrol versi](https://id.wikipedia.org/wiki/Pengontrol_versi) [*Git*](https://id.wikipedia.org/wiki/Git) dan [layanan *hosting* internet](https://id.wikipedia.org/wiki/Layanan_hosting_internet). Github memberikan kontrol akses dan beberapa fitur kolaborasi seperti pelacakan *bug*, permintaan fitur, manajemen tugas pada setiap proyek yang dikerjakan. (Williams, 2012).



Gambar 2. 2 GitHub (Friedman, 2008)

1. Git

Git adalah version control system dimana bisa mengingat setiap perubahan *file*. Git sangat efektif dan cepat untuk membantu kita khususnya dalam pembuatan projek yang besar dan dengan tidak berbayar. (Farenza, 2018).



Gambar 2. 3 Git (Firdaus, 2016)

1. **XAMPP**

XAMPP adalah [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak_bebas) yang mendukung banyak [sistem operasi](https://id.wikipedia.org/wiki/Sistem_operasi), merupakan kompilasi dari beberapa [program](https://id.wikipedia.org/wiki/Program). Fungsinya adalah sebagai [*server*](https://id.wikipedia.org/wiki/Server) yang berdiri sendiri ([localhost](https://id.wikipedia.org/w/index.php?title=Localhost&action=edit&redlink=1)), yang terdiri atas program [*apache* HTTP *server*](https://id.wikipedia.org/wiki/Apache_HTTP_Server), [MySQL](https://id.wikipedia.org/wiki/MySQL) [*database*](https://id.wikipedia.org/wiki/Database), dan [penerjemah](https://id.wikipedia.org/wiki/Penerjemah) [bahasa](https://id.wikipedia.org/wiki/Bahasa) yang ditulis dengan [bahasa](https://id.wikipedia.org/wiki/Bahasa) [pemrograman](https://id.wikipedia.org/wiki/Pemrograman) [PHP](https://id.wikipedia.org/wiki/PHP) dan [Perl](https://id.wikipedia.org/wiki/Perl). Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), [Apache](https://id.wikipedia.org/wiki/Apache), [MySQL](https://id.wikipedia.org/wiki/MySQL), [PHP](https://id.wikipedia.org/wiki/PHP) dan [Perl](https://id.wikipedia.org/wiki/Perl). [Program](https://id.wikipedia.org/wiki/Program) ini tersedia dalam [*GNU General Public License*](https://id.wikipedia.org/wiki/GNU_General_Public_License)dan [bebas](https://id.wikipedia.org/wiki/Bebas). XAMPP merupakan [web server](https://id.wikipedia.org/wiki/Web_server) yang mudah digunakan yang dapat melayani tampilan halaman [web](https://id.wikipedia.org/wiki/Web) dinamis. Untuk *install* XAMPP dapat mengunduh langsung dari web resminya. ([Schaferhoff](https://torquemag.io/author/nschaferhoff1234/), 2015).



Gambar 2. 4 XAMPP (Seidler, 2015)

1. **MySQL**

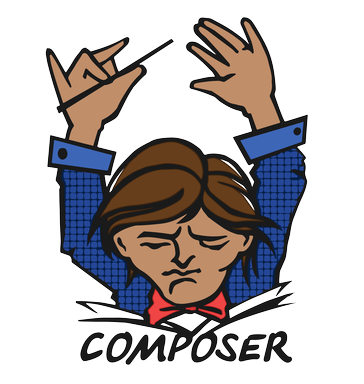
MYSQL adalah sebuah perangkat lunak system yang menangani manajemen basis data SQL atau *Database Management System* yang multialur dan multipengguna. MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL). Pada saat ini MySQL merupakan database server yang sangat terkenal di dunia, semua itu tak lain karena bahasa dasar yang digunakan untuk mengakses *database* yaitu MySQL. Elemen dari SQL yang paling dasar antara lain pernyataan, nama, tipe data, ekspresi, konstanta dan fungsi bawaan pada *database*. (Fikriansyah, 2017).



Gambar 2. 5 MySQL (Ronhjones, 2018)

1. **Composer**

Composer adalah aplikasi [*package manager*](https://id.wikipedia.org/wiki/Package_manager) bahasa pemrograman *Hypertext Preprocessor* ([PHP](https://id.wikipedia.org/wiki/PHP)) yang menyediakan format standar untuk mengelola dependensi dan pustaka yang diperlukan dalam membuat web. Composer memungkinkan untuk membuat *library* pada projek yang dibuat dan composer sendiri akan *install* atau *updat*e secara otomatis tanpa harus melakukan manual. (Adermann, 2012).



Gambar 2. 6 Composer (Adermann, 2012)

Composer berjalan melalui [baris perintah](https://id.wikipedia.org/wiki/Antarmuka_baris_perintah) dan *install* otomatis dependensi (pustaka perangkat lunak) untuk aplikasi. Hal ini juga memungkinkan pengguna untuk menginstal aplikasi PHP yang tersedia di "*Packagist*"yang mana itu adalah repositori utamanya. (Adermann, 2012).

1. **Netbeans**

Adapun netbeans adalah sebuah aplikasi IDE (*Integrated Development Environment*) besutan SUN Microsystems. aplikasi ini berbasiskan Java dan berjalan di atas *swing. Swing* sendiri adalah sebuah teknologi Java untuk mengembangkan sebuah aplikasi Desktop yang dapat dijalankan di berbagai macam OS. yaitu Windows, Linux, MAC OS ,Solaris dan OS lain yang mendukung suatu JVM yang sepadan.

Pada awalnya, aplikasi ini diperuntukkan dalam suatu pengembangan pemrograman Java. Namun, aplikasi ini juga mendukung bahasa pemrograman lainnya, secara khusus seperti PHP, C/C++ dan HTML5.

Fitur - fitur yang terdapat pada Netbeans antara lain diambil dari (Dictio, 2017):

1. Smart Code Completion, untuk mengusulkan nama variabel dari suatu tipe, melengkapi *keyword* dan mengusulkan tipe parameter dari sebuah *method*.
2. Bookmarking, fitur yang digunakan untuk menandai baris yang suatu saat hendak kita modifikasi.
3. Go to commands, fitur yang digunakan untuk jump ke deklarasi variabel, *source code* atau *file* yang ada pada projek yang sama.
4. Code generator, jika kita menggunakan fitur ini kita dapat membuat konstruktur*, setter and getter method* dan yang lainnya.
5. Error stripe, fitur yang akan menandai baris yang eror dengan memberi *highlight* berwarnamerah.



Gambar 2. 7 Netbeans (Kurniawan, 2018)

Kelebihan netbeans mempunyai *Garbage Collection* yaitu dapat membuat program berbasis visual dan *event driven*, sehingga programmer tidak perlu membuat teks program secara manual baris per baris untuk membuat dialog atau pengguna antarmuka nantinya akan menghasilkan teks program secara otomatis, di dalam Netbeans juga terdapat *Compailer,Builder,dan Debugger* internal.

1. **Sublime Text 3**

*Sublime text* adalah sebuah teks editor yang elegan, kaya fitur, *cross platform*, mudah dan simpel yang cukup terkenal di kalangan *Developer* (pengembang). *Programmer* biasanya menggunakan *sublime text* untuk menyunting *source code* yang sedang kerjakan. Sampai saat ini *sublime text* sudah mencapai versi 3. *Sublime* *text* mempunyai beberapa keunggulan-keunggulan yang dapat membantu pengguna dalam membuat sebuah *web development*. Berikut keunggulan-keunggulan fitur yang dimiliki *sublime text* 3 diambil dari (Karmelia, 2015) adalah :

1. *Multiple* *Selection*

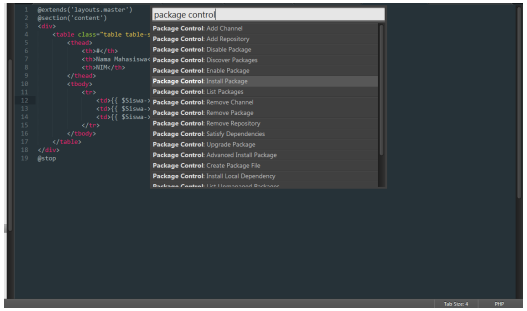
*Multiple* *Selection* mempunyai fungsi untuk membuat perubahan pada sebuah kode pada waktu yang sama dan dalam baris yang berbeda. *Multiple* *selection* ini juga merupakan salah satu fitur unggulan dari *sublime text* 3. dapat meletakkan kursor pada kode yang akan diubah, lalu tekan Ctrl+klik atau blok kode yang akan diubah kemudian Ctrl+D setelah itu kita dapat merubah kode secara bersamaan.



Gambar 2. 8 Multiple Selection

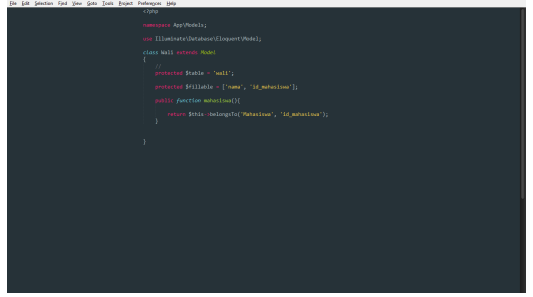
1. *Command Pallete*

*Command Pallete* mempunyai fungsi untuk mengakses *file* shortcut dengan mudah. Untuk mencari *file* tersebut kita dapat tekan Ctrl+Shift+P, kemudian cari perintah yang diinginkan.

**

Gambar 2. 9 *Command Pallete*

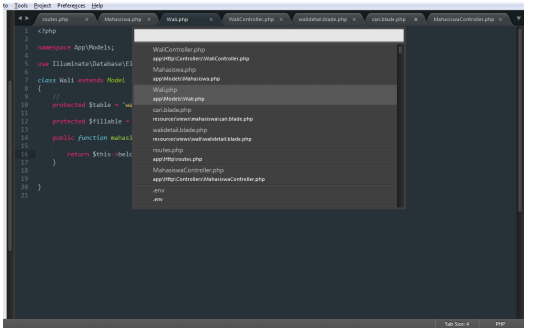
1. *Distraction Free Mode*

Fitur ini mempunyai fungsi untuk merubah tampilan layar menjadi penuh dengan menekan SHIFT + F11. Fitur ini dibutuhkan ketika pengguna ingin fokus pada pekerjaan yang sedang dikerjakannya. *Distraction Free Mode* menampilkan *file* berupa layar penuh, dengan apapun kecuali teks yang ditampilkan di tengah monitor. semua layanan web tersembunyi tetapi dapat diakses. *Distraction Free Mode* dapat dimasukkan kedalam item menu lihat atau yang biasa disebut *Distraction Free Mode*.

Gambar 2. 10 *Distraction*

1. *Find in project*

Fitur *find in project* dapat mencari atau membuka *file* di dalam sebuah projek dengan cepat dan mudah. Hanya dengan menekan Ctrl+P langsung dapat mencari *file* yang diinginkan.



Gambar 2. 11 *Find in Project*

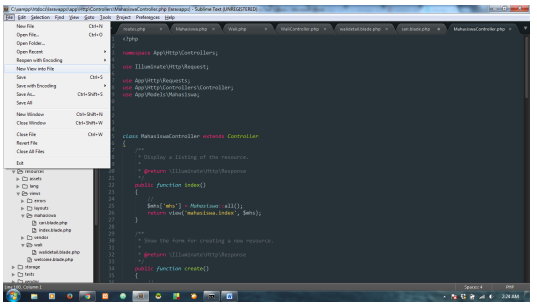
1. *Plugin API* *Switch*

*Sublime Text* mempunyai keunggulan dengan *plugin* yang berbasis *Python Plugin* *API*. Teks editor ini juga mempunyai *plugin* yang sangat beragam, dan ini dapat memudahkan pengguna dalam mengembangkan perangkat lunak.

1. *Drag and Drop*

Dalam teks editor ini pengguna dapat *drag and drop* *file* teks ke dalam editor yang akan membuka tab baru secara otomatis.

1. *Split Editing*

Di dalam fitur ini kita dapat mengedit *file* secara berdampingan dengan klik *file* -> *New menu into file*.

Gambar 2. 12 *Split Editing*

1. *Multi Platform*

*Sublime text* juga mempunyai keunggulan dalam berbagai *platform*. *Sublime text* sendiri sudah tersedia dalam berbagai *platform* sistem operasi, yaitu Windows, Linux, dan iOS.

Teks editor *sublime text* mempunyai banyak keunggulan dan membuat para pengguna teks editor ini mudah saat pembuatan sebuah aplikasi maupun sebuah web. Setelah mengetahui fitur diatas, ada fitur dari *sublime text* yaitu *Package Control*. *Package Control* dari berisikan beberapa *package* yang bisa diinstal pada *sublime text* 3. (Karmelia, 2015).

1. ***UML***

*UML* singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. Menurut Chonoles (2003) mengatakan sebagai bahasa berarti *UML* memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep *UML* ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang dibuat berhubungan satu dengan lainnya dan mengikuti standar yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. *UML* diaplikasikan untuk tujuan dari berikut diambil dari (Oktafianto, 2017) :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisis dan mencari apa yang diperlukan oleh sistem.
4. Mendokumentasi sistem yang ada, proses dan organisasinya.

* **Jenis-Jenis Diagram UML**

1. *Use Case Diagram*

*Use Case Diagram* merupakan alat komunikasi untuk mewakili persyaratan dari sebuah sistem. Pada use case diagram menunjukan interaksi antara pengguna dengan entitas lainnya dalam sistem yang ingin dikembangkan, penyajian *use case diagram* berupa interaksi antara *use case* dan *actor*. Berikut ini adalah beberapa simbol dari *use case diagram*:

Tabel 2. 1 simbol *use case*  dan pengertian

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | *Actor* merupakan spesifikasi pada peran pengguna ketika berhubungan dengan *use case.* |
|  | *Extend* merupakan menunjukan bahwa suatu *use case* tambahan fungsional dari *use case* lainnya jika suatu kondisi terpenuhi. |
|  | *Include* merupakan menunjukan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya. |
|  | *Association* merupakan abstraksi dari penghubung antara *actor* dan *use case*. |
|  | *Use case* merupakan abstraksi dari interaksi antara sistem dan *actor*. |
|  | *System* menspesifikasikan paket yang menampilkan sistem secara terbatas. |

Sumber: (edu.blog, 2015)

1. *Activity Diagram*

*Activity diagram* merupakan gambaran dari alur aktivitas dalam sebuah sistem yang sedang dirancang dan cara agar bagaimana alur aplikasi dapat berjalan dengan baik. *Activity diagram* bisa mempunyai kemiripan dengan *flowchart* padahal fungsi nya berbeda. *Activity diagram* bisa mendukung perilaku paralel dan juga mempunyai *state* diagram yang khusus, dimana sebagian besar *state* adalah aksi. Berikut beberapa simbol dari *activity diagram*:

Tabel 2. 2 Simbol-simbol *activity* *diagram* dan penjelasan

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | *Activity* untuk memperlihatkan masing-masing kelas antarmuka saling berinteraksi satu sama lain. |
|  | *Action* untuk *state* dari sistem yang mencerminkan eksekusi dari suatu aksi. |
|  | *Actifity Final Node* untuk objek yang dibentuk dan diakhiri. |
|  | *Initial Node* untuk objek yang dibentuk atau diawali. |
|  | *Fork Node* untuk satu aliranpada tahap tertentu berubah menjadi beberapa aliran. |

Sumber: (Nugroho, 2010)

1. *Sequence Diagram*

*Sequence diagram* menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosisasi dengan *use case* diagram. *Sequence diagram* memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam sequence diagram. Berikut beberapa simbol dari *sequence diagram*:

Tabel 2. 3 Simbol-simbol *sequnece* *diagram* dan penjelasan

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | *Anchor Note to item* adalah melekatkan catatan pada *use case* atau *actor* tertentu dalam diagram. |
|  | *Object* untuk menambahkan objek baru pada diagram. |
|  | *Object Message;* menggambarkan pesan dua objek. |
|  | *Return Message* untuk menggambarkan pengembalian diri pemanggilan prosedur. |
|  | *Message to self* untuk menggambarkan pesan yang menuju dirinya sendiri. |

Sumber: (Nugroho, 2010)

1. *Class Diagram*

*Class diagram* membantu programer dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* memperlihatkan hubungan antarkelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem. Berikut beberapa simbol dari *class diagram*:

Tabel 2. 4 Simbol-simbol *class diagram* dan penjelasan

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | *Interface* untuk menambahkan kelas antarmuka pada diagram. |
|  | *Class* untuk menambahkan kelas baru pada diagram. |
|  | *Association* untuk menggambarkan relasi asosiasi. |

Sumber: (Nugroho, 2010)

1. **Normalisasi Data**

Perancangan basis data diperlukan agar bisa memiliki basis data yang kompak dan efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan dan mudah dalam pemanipulasian (tambah, ubah, hapus) data. Dalam merancang basis data, kita dapat melakukannya dengan menerapkan normalisasi terhadap struktur tabel yang telah diketahui atau dengan langsung membuat model *Entity-Relationship .*perancang basis data seringkali diasosiasikan dengan pembuatan model Entity-Relationship (model E-R), dimana kelompok-kelompok data dan relasi antarkelompok data tersebut diwujudkan dalam bentuk diagram. Normalisasi sendiri merupakan cara pendekatan lain dalam membangun lojik basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal. Namun demikian, dalam pelaksanakannya desain lojik basis data relasional yang didasari baik oleh prinsip normalisasi maupun yang didasari oleh transformasi secara hati-hati dari model E-R ke bentuk fisik akan menghasilkan hasil yang mirip. Dalam uraian tentang normalisasi basis data menggunakan istilah seperti Atribut, Key, Domain dan ketergantungan fungsional.

1. Atribut Tabel (*Table Attribute)*

Atribut yang sebenarnya identik dengan pemakaian istilah kolom data. Istilah atribut ini lebih umum digunakan dalam perancangan basis data, karena istilah itu lebih impresif menunjukkan fungsinya sebagai pembentuk karakteristik (sifat-sifat) yang melekat pada sebuah tabel. Penerapan aturan-aturan normalisasi terhadap atribut-atribut pada sebuah tabel bisa berdampak pada penghilangan kolom tertentu, penambahan kolom baru, atau bahkan penambahan tabel baru.

Contoh:

* Tabel Mahasiswa memiliki 4 buah atribut, yaitu nim, nama\_mhs, alamat\_mhs, dan tgl\_lahir.
* Tabel Dosen memiliki 2 buah atribut, yaitu nama\_dosen dan alamat\_dosen.
* Tabel Kuliah memiliki 4 buah atribut, yaitu kode\_kul, nama\_kul, sks dan semester.

1. Domain dan tipe data

Domain memiliki banyak kesamaan pengertian dengan fungsi tipe data tersebut. Akan tetapi tipe data lebih merujuk pada kemampuan penyimpanan data yang mungkin bagi suatu atribut secara fisik, tanpa melihat layak atau tidaknya data tersebut bila dilihat dari kenyataan pemakaiannya. Sementara domain nilai lebih ditekankan pada batas-batas nilai yang diperbolehkan bagi suatu atribut, dilihat dari kenyataan yang ada.

1. Ketergantungan Fungsional (*Functional Dependency*)

Diberikan sebuah tabel T berisi paling sedikit 2 buah atribut, yaitu A dan B dengan dapat menyatakan notasi berikut ini:

A→B

Yang berarti A secara fungsional menentukan B atau B secara fungsional tergantung pada A, jika dan hanya jika untuk setiap kumpulan baris data (*row*) di tabel T dengan nilai untuk A yang sama, maka nilai untuk B pasti juga sama. Definisi yang lebih formal untuk itu adalah:

Diberikan 2 *row* rl dan r2 dalam tabel T dimana A→B, jika rl (A) = r2 (A), maka r1 (B) = r2 (B).

1. Normalisasi dengan Ketergantungan Fungsional

Dalam perspektif normalisasi, sebuah basis data dapat dikatakan baik, jika setiap tabel yang menjadi unsur pembentuk basis data tersebut juga telah berada dalam keadaan baik atau normal. Selanjutnya sebuah tabel dapat dikategorikan baik (efisien) atau normal, jika telah memenuhi tiga kriteria berikut:

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition)*.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).
3. Tidak melanggar *Boyce-Code Normal Form* (*BCNF*).

Jika kriteria ketiga (*BCNF*) tidak dapat terpenuhi, maka paling tidak harus diupayakan agar tabel tersebut tidak melanggar bentuk normal tahap ketiga (*3rd Normal Form/*3NF).

**Normalisasi *database***sendiri terdiri dari banyak bentuk, dalam ilmu basis data ada setidaknya 9 bentuk normalisasi yang ada yaitu 1NF, 2NF, 3NF, EKNF, BCNF, 4NF, 5NF, DKNF, dan 6NF. Namun dalam prakteknya dalam dunia industri bentuk normalisasi ini yang paling sering digunakan ada sekitar 3 bentuk yaitu *First Normal Form*, *Second Normal Form*, *Third Normal Form*.

* ***First Normal Form* (1NF)**

Bentuk normal yang pertama atau 1NF mensyaratkan beberapa kondisi dalam sebuah database, yaitu fungsi dari bentuk normal pertama adalah menghilangkan duplikasi kolom dari tabel yang sama, dan buat tabel terpisah untuk masing-masing kelompok data terkait dan mengidentifikasi setiap baris dengan kolom yang unik atau *primary key*. (Karjono, 2018).

Tabel 2. 5 Normalisasi 1NF

|  |  |  |
| --- | --- | --- |
| IDBuku | Nama\_buku | Tgl\_terbit |
| B01 | Komputer | 20-Jan-17 |
| B02 | AC | 23-Mar-17 |
| B03 | Proyektor | 25-Jan-18 |
| B04 | CPU | 03-Jun-18 |

Sumber: (Karjono, 2017)

Pada intinya bentuk normalisasi 1NF mengelompokkan beberapa tipe data atau kelompok data yang sejenis agar dapat dipisahkan sehingga anomali data dapat di atasi. Contoh ketika menghapus, mengupdate, atau menambahkan data peminjam, maka kita tidak bersinggungan dengan data buku atau data terbit. Sehingga inkonsistensi data dapat dijaga dengam baik. (Karjono, 2018).

* ***Second normal form* (2NF)**

Syarat untuk menerapkan normalisasi bentuk kedua ini adalah data telah dibentuk dalam 1NF, berikut adalah beberapa fungsi normalisasi 2NF diambil dari (Karjono, 2018):

1. Menghapus beberapa subset data yang ada pada tabel dan menempatkan mereka pada tabel terpisah.
2. Menciptakan hubungan antara tabel baru dan tabel lama dengan menciptakan foreign key.
3. Tidak ada atribut dalam tabel yang secara fungsional bergantung pada candidate key tabel tersebut.

Tabel 2. 6 Normalisasi 2NF

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID\_TRX | Judul\_Buku | IDBuku | IDPeminjam | IDPenerbit | Nm\_penerbit |
| 1111 | Basis data | B01 | P01 | P01 | Erlangga |
| 2222 | Basis data | B01 | P01 | P02 | Erlangga |

Sumber: (Karjono, 2017)

Tabel 2. 7 Bentuk 2NF dari tabel diatas

|  |  |  |  |
| --- | --- | --- | --- |
| ID\_TRX | IDBuku | IDPeminjam | IDPenerbit |
| 1111 | B01 | P01 | P01 |
| 2222 | B01 | P01 | P02 |

Sumber: (Karjono, 2017)

Pada intinya bentu kedua ini adalah tidak boleh ada *field* yang berhubungan dengan *field* lainnya secara fungsional. Contoh nya dalam tabel diatas yaitu judul buku tergantung dengan IDBuku sehingga dalam bentuk 2NF judul buku dapat di hilangkan karena telah memiliki tabel master tersendiri.

* ***Third Normal Form* (3NF)**

Normalisasi *database* dalam bentuk 3NF bertujuan untuk menghilangkan seluruh atribut atau *field* yang tidak berhubungan dengan *primary key*. Syarat dari bentuk normal ketiga atau 3NF adalah :

1. Memenuhi semua persyaratan dari bentuk normal kedua.
2. Menghapus kolom yang tidak tergantung pada primary key.

Tabel 2. 8 Normalisasi 3NF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OrderID | CustID | Harga | Jumlah | Total |
| 112 | C111 | 1000 | 21 | 21000 |
| 113 | C112 | 1000 | 22 | 22000 |

Sumber: (Karjono, 2017)

Tabel 2. 9 Bentuk 3NF dari tabel diatas

|  |  |  |  |
| --- | --- | --- | --- |
| OrderID | CustID | Harga | Jumlah |
| 112 | C111 | 1000 | 21 |
| 113 | C112 | 1000 | 22 |

Sumber: (Karjono, 2017)

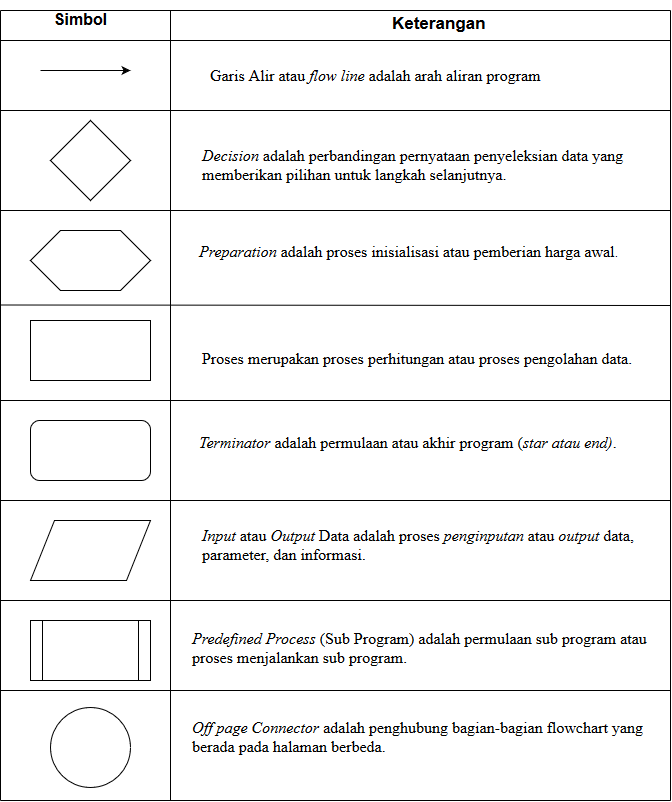
Tidak semua kolom tergantung pada *primary key* tetapi pada satu *field* total yang bergantung pada harga dan jumlah, total dapat dihasilkan dengan mengalikan harga dan jumlah. Bentuk 3NF dalam tabel di atas dapat dilakukan dengan membuang *field* total.

1. ***Flowchart***

*Flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara jelas pada hubungan antara suatu proses berupa intruksi dengan proses lainnya dalam suatu program. (Arifianto, 2014).

Tujuan pembuatan *flowchart* adalah untuk menggambarkan suatu tahapan masalah secara sederhana dan jelas dengan menggunakan simbol-simbol standar pada *flowchart.* Dengan adanya *flowchart* setiap urutan proses dapat digambarkan menjadi lebih jelas. Berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu *flowchart:*

Tabel 2. 10 simbol-simbol *Flowchart* dan penjelasan



Sumber: (bukubiru.com)

1. ***ERD* (*Entity Relationship Diagram*)**

Pengertian *ERD* atau *entity relationship diagram* adalah pemodelan data utama yang membantu mengorganisasikan data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Sistem adalah kumpulan elemen yang setiap elemen memiliki fungsi masing-masing dan secara bersama-sama mencapai tujuan dari sistem tersebut. Kebersamaan dari sistem yang dilambangkan dengan saling berelasinya antara satu dan entitas dengan entitas yang lainnya. Entitas (*entity*  atau *entity set*), memiliki banyak istilah di dalam ilmu komputer, seperti table (*table*), berkas (data *file*), penyimpanan data (data *store*), dan sebagainya. (Satria, 2012).

Berikut beberapa simbol dari *Entity Relationship Diagram:*

Tabel 2. 11 simbol-simbol *Entity Relationship Diagram* dan penjelasan

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | Entitas adalah representasi dari hampir semua informasi gabungan yang harus dipahami oleh perangkat lunak. |
|  | Relasi adalah hubungan yang ditunjukkan dengan garis yang diberi label yang menghubungkan objek. |
|  | Link adalah tanda garis yang digunakan untuk menghubungkan komponen-kompenen ERD. |
|  | Atribut adalah untuk menentukan properti suatu objek data dan mengambil salah satu dari tiga karakteristik yang berbeda. |

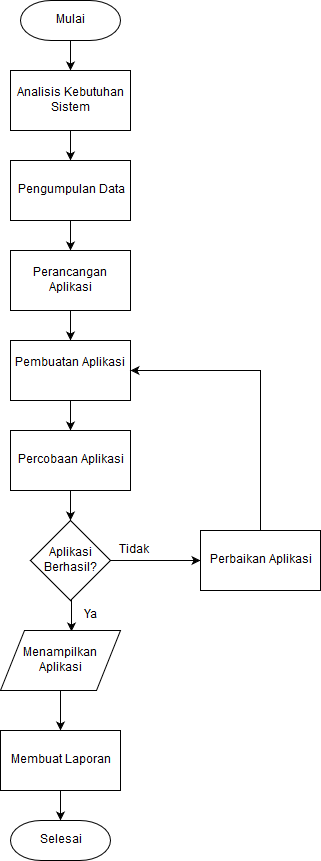
Sumber: (Satria, 2012)

**BAB III**

**METODOLOGI PENELITIAN**

1. **Metode Penelitian**

Metodologi penelitian berisi tahap yang digunakan untuk mendapatkan informasi dalam penelitian ini agar terstruktur dengan baik. Dengan sistematika ini proses penelitian dapat dipahami dan sesuai dengan masalah yang ada. Penelitian yang dilakukan untuk merancang sistem diperoleh dari pengamatan data-data yang sudah dikumpulkan dari berbagai sumber. Dalam penelitian memiliki beberapa tahapan dalam prosesnya tahapan-tahapan itu meliputi analisis kebutuhan sistem, pengumpulan data, perancangan aplikasi, pembuatan aplikasi, percobaan aplikasi, menampilkan aplikasi, dan pembuatan laporan untuk evaluasi. Untuk lebih jelasnya dapat dilihat pada *flowchart* penelitian pada Gambar berikut.



Gambar 3. 1 *Flowchart* Penelitian

1. **Analisa Kebutuhan Sistem**

Pada pembuatan Rancang Bangun Aplikasi Pengelolaan dan perawataan Laboratorium Politeknik Negeri Indramayu terdapat beberapa kebutuhan, dari kebutuhan dalam bentuk *hardware* (perangkat keras) maupun kebutuhan dalam bentuk *software* (perangkat lunak).

1. **Kebutuhan Hardware (Perangkat Keras)**

Kebutuhan hardware untuk implementasi kebutuhan sistem dapat dilihat pada Tabel 3.1 Kebutuhan *Hardware*.

Tabel 3. 1 Kebutuhan *Hardware*

|  |  |  |
| --- | --- | --- |
| **No.** | **Jenis Perangkat Keras** | **Kebutuhan Perangkat Keras** |
| 1. | *Processor* | Minimal *Dual Core LGA* 775 |
| 2. | Memori *(RAM)* | *RAM* 2 *GB* |
| 3. | Penyimpan *(Harddisk)* | 200 *GB* |

1. **Kebutuhan Software (Perangkat Lunak)**

Kebutuhan hardware untuk implementasi kebutuhan sistem dapat dilihat pada Tabel 3.2 Kebutuhan *Software*.

Tabel 3. 2 Kebutuhan *Software*

|  |  |  |
| --- | --- | --- |
| **No.** | **Jenis Perangkat Lunak** | **Kebutuhan Perangkat Lunak** |
| 1. | Sistem Operasi | *Microsoft Windows 7 or Higher* |
| 2. | Desain Aplikasi | *Balsamiq Mockup* |
| 3. | Bahasa Pemrograman | *PHP, CSS, HTML, Javascript, Java* |
| 4. | *Software* Pengolah | *Sublime Text 3, Netbeans* |
| 5. | Penyimpan Data | *MySQL* |
| 6. | *Web Browser* | *Google Chrome, Mozile Firefox, IE,* dan lain-lain |

1. **Metode Pengumpulan Data**

Metode Pengumpulan Data terdiri dari 2 cara yaitu tahap kuisioner, kaji pustaka diataranya sebagai berikut:

1. **Kuisioner**

Penulis membuat kuisioner yang berisi pertanyaan-pertanyaan yang berkaitan dengan sistem yang akan dibuat khususnya seputar tentang proses di ruang laboratorium sehingga penulis mendapatkan data-data yang dapat menjadi acuan dalam pengimplementasian sistem aplikasi.

Tabel 3. 3 Kuisioner

|  |  |
| --- | --- |
| **Kebutuhan Aplikasi** | |
| No. | **Pertanyaan** |
| 1 | Bagaimana respon (mahasiswa) apabila *logbook* komputer di laboratorium menggunakan kertas? |
| 2. | Bagaimana sistem peminjaman alat di *maintenance*? |
| 3. | Apakah sebelumnya belum ada yang membuat aplikasi untuk laboratorium? |

1. **Kaji Pustaka**

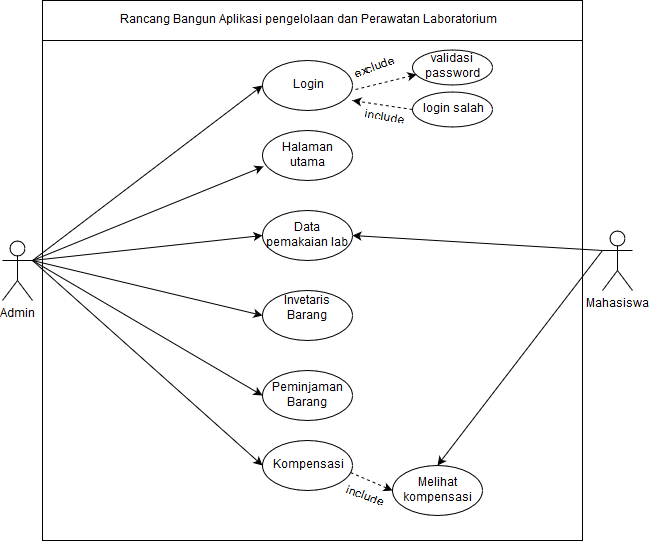
Pada tahapan ini penulis mempelajari referensi yang relevan berupa makalah, laporan tugas akhir, dan buku. Fasilitas internet juga digunakan sebagai media untuk mencari data atau informasi yang ada di dunia maya yang berkaitan dengan objek penelitian.

1. **Perancangan Sistem**

Perancangan sistem ini menggunakan UML (*Unified Modelling Language*). Pada perancangan sistem digambarkan secara garis besar mengenai rancang bangun aplikasi pengelolaan dan perawatan laboratorium Politeknik Negeri Indramayu untuk kebutuhan yang akan dijelaskan pada sub bab berikutnya adalah:

1. ***Use* *Case Diagram***

Pada *use case diagram* yang dibuat disini dalam pembuatan rancang bangun aplikasi pengelolaan dan perawatan laboratorium ini memiliki beberapa aktor diantaranya aktor *administrator*, dan aktor mahasiswa, dimana setiap aktor memiliki hak akses yang berbeda-beda. Aktor *administrator* memiliki hak akses dalam mengelola semua master data, sedangkan mahasiswa hanya dapat mengisi form pemakaian lab dan melakukan peminjaman barang di *maintenance.*



Gambar 3. 2 *Use Case Diagram*

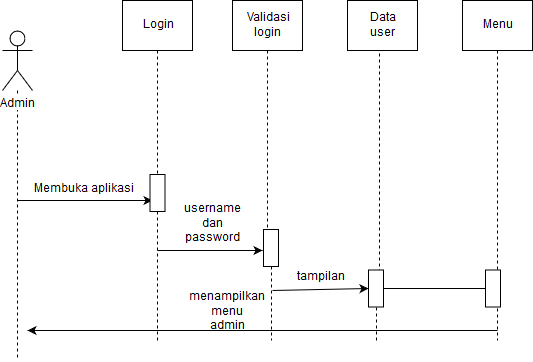
1. ***Sequence Diagram***

*Sequence diagram* menggambarkan kolaborasi dinamis antara sejumlah objek. *sequence diagram* merupakan langkah-langkah pada sistem sebagai respon untuk menghasilkan output. Kegunaan dari *sequence diagram* adalah menunjukan rangkaian pesan yang dikirim antara objek yang berinteraksi dengan objek.

1. ***Sequence diagram* admin masuk ke sistem**

Pada sequence diagram admin menerangkan bagaimana langkah-langkah admin masuk ke sistem, untuk masuk ke sistem memerlukan login terlebih dahulu, lalu sistem merespon dengan memvalidasi login apabila username dan password yang dimasukan benar maka menampilkan sesuai dengan data user dan menu utama.

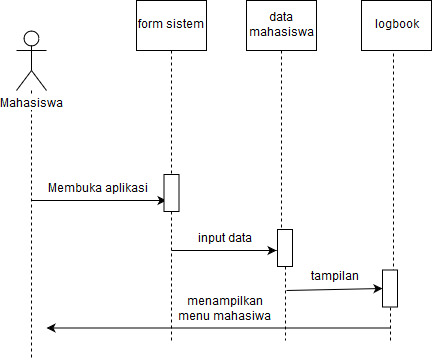
Jika validasi login berhasil maka menampilkan proses menu yaitu terdapat halaman dashboard, data pemakaian komputer di Lab, invetaris barang, dan peminjaman barang. Namun apabila login validasi salah atau gagal maka kembali ke halaman login.



Gambar 3. 3 sequence diagram admin

1. ***Sequence diagram* mahasiswa**

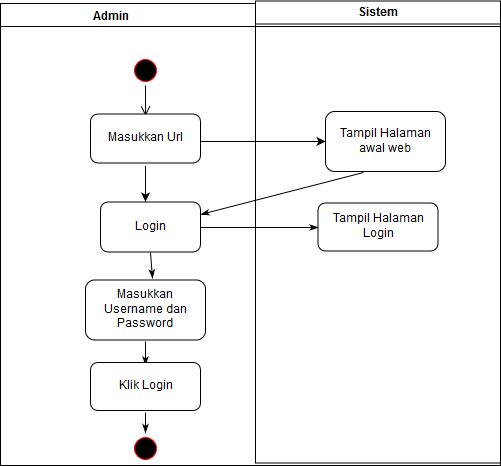
Mahasiswa membuka aplikasi dekstop yang sudah tersedia di komputer laboratorium, lalu sistem menampilkan form input nama, kelas, mata kuliah, tanggal, dan waktu mulai hingga waktu selesai.



Gambar 3. 4 *sequence diagram* mahasiswa

1. ***Activity Diagram***
2. *Activity Diagram* *Login*

*Activity Diagram* *Login* yaitu langkah-langkah admin untuk masuk ke dalam sistem seperti gambar dibawah berikut.



Gambar 3. 5 *Activity Diagram Login*

1. *Activity Diagram* Invetaris Barang

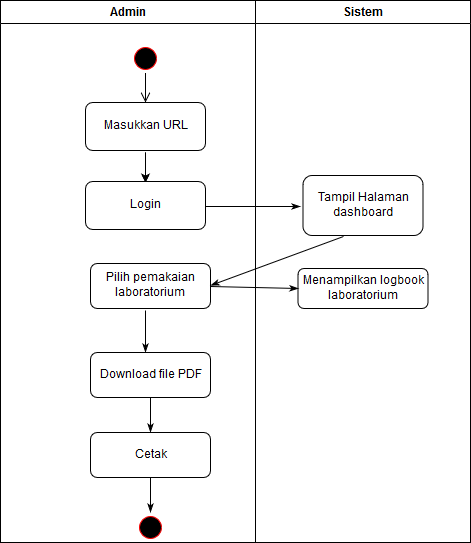
*Activity Diagram* Inventaris Barang yaitu langkah-langkah admin untuk menambahkan data barang seperti gambar dibawah berikut.



Gambar 3. 6 *Activity Diagram* Invetaris Barang

1. *Activity Diagram* Pemakaian Laboratorium

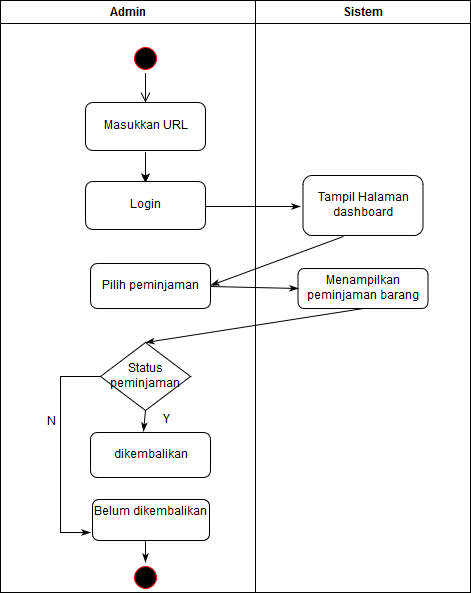
*Activity Diagram* Pemakaian Laboratorium yaitu langkah-langkah admin untuk melihat data pemakaian laboratorium yang telah diisi dari user seperti gambar dibawah berikut.



Gambar 3. 7 *Acitivity Diagram* Pemakaian Laboratorium

1. *Activity Diagram* Peminjaman

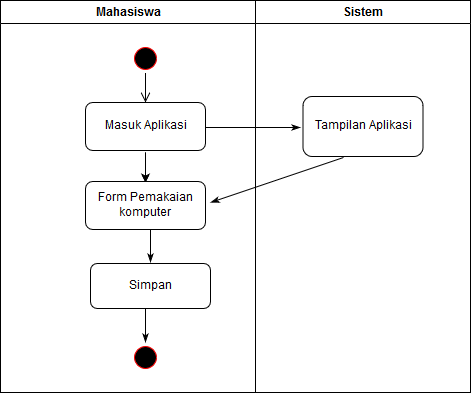
*Activity Diagram* Peminjaman yaitu langkah-langkah admin untuk memverifikasi peminjaman seperti gambar dibawah berikut.



Gambar 3. 8 *Activity Diagram* Peminjaman

1. *Activity Diagram User*

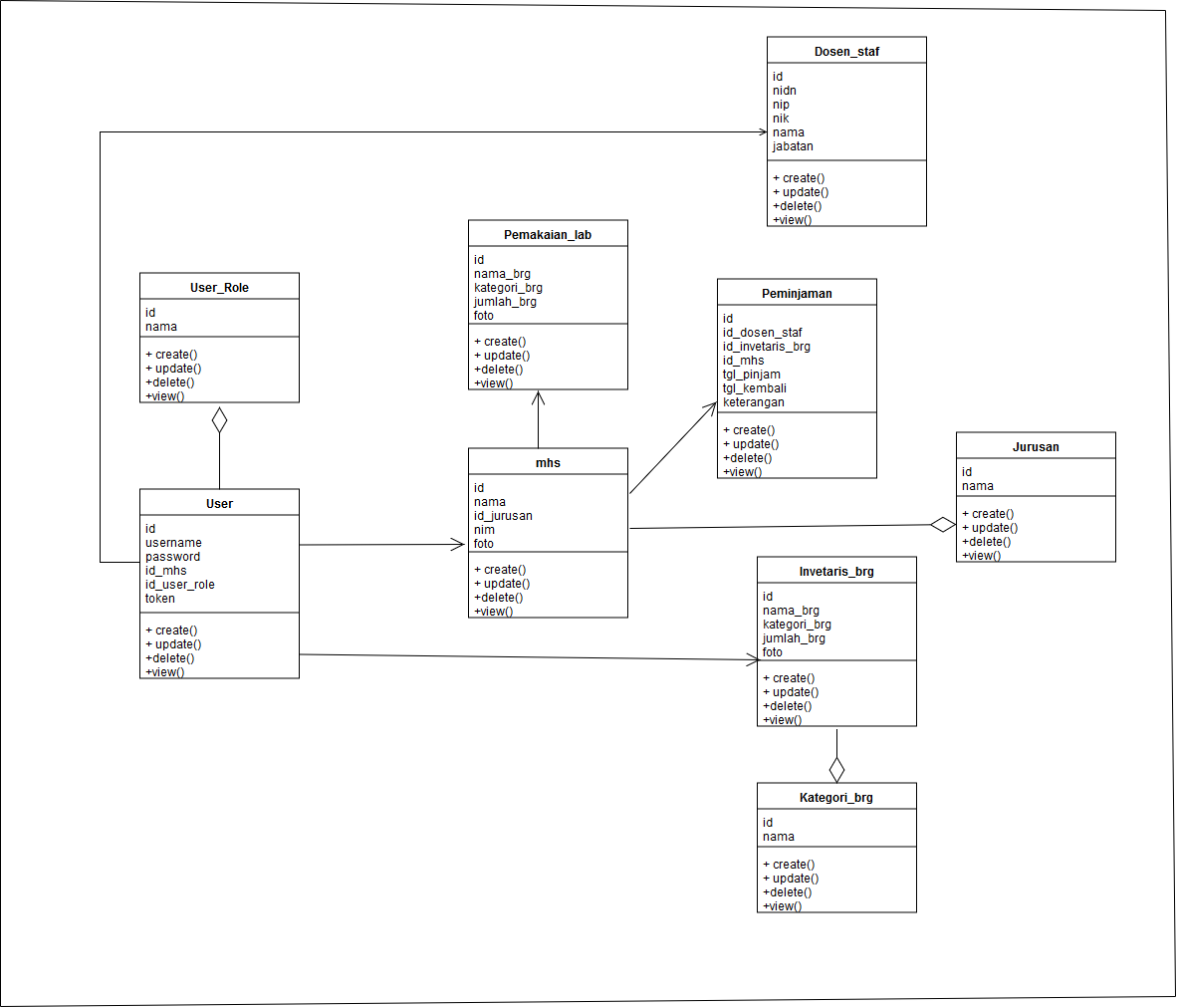
*Activity Diagram* *User* yaitu langkah-langkah *user* untuk memasukkan data pada komputer di pemakaian laboratorium seperti gambar dibawah berikut.



Gambar 3. 9 Activity Diagram User

1. ***Class Diagram***

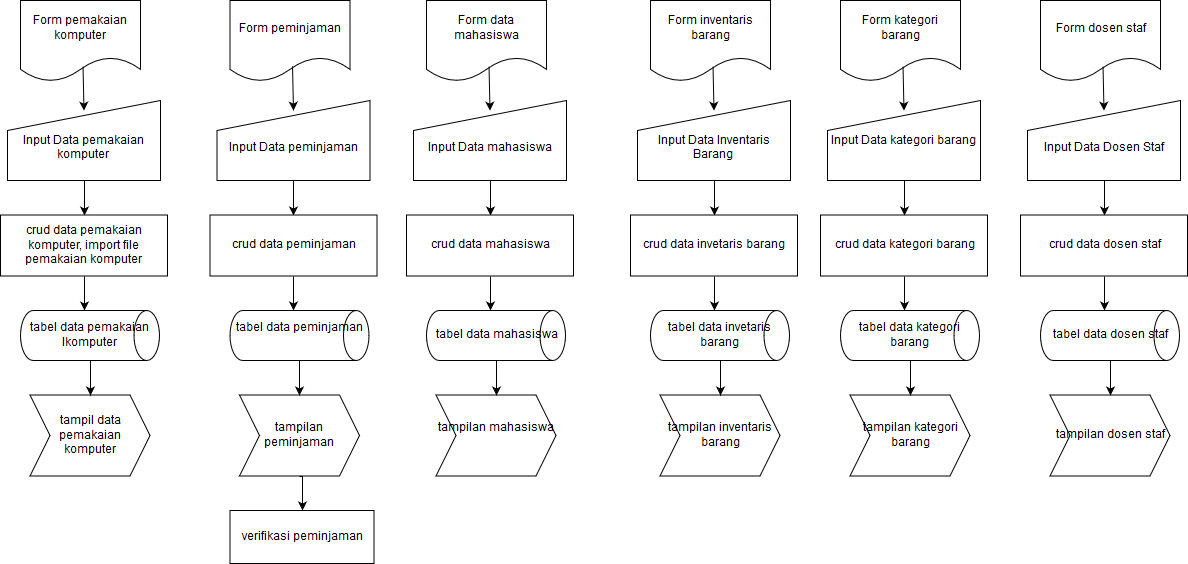
*Class diagram* menunjukan interaksi antar kelas dalam sistem. Class diagram juga memperlihatkan struktur dari sistem berupa sistem kelas, attribut, dan hubungan antar kelas. Adapun class diagram dari aplikasi ini terdapat pada Gambar di bawah berikut.

Gambar 3. 10 *Class Diagram*

1. ***Flowchart***
2. ***Flowchart System***

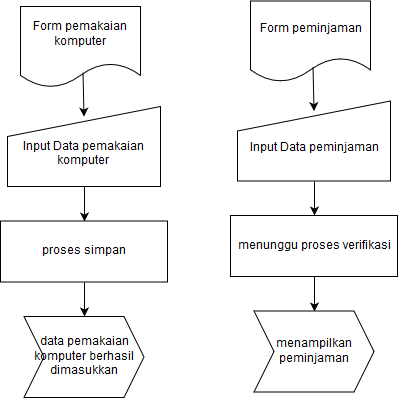
*Flowchart system* dari pembuatan aplikasi ini yaitu suatu bagan yang memperlihatkan urutan alur kerja dan proses dari beberapa *file* di dalam media tertentu. Melalui *flowchart* ini terlihat jenis media penyimpanan yang dipakai dalam pengolahan data. Adapun *flowchart system* dari pembuatan aplikasi ini terdapat pada Gambar.

1. ***Flowchart System* Admin**

****

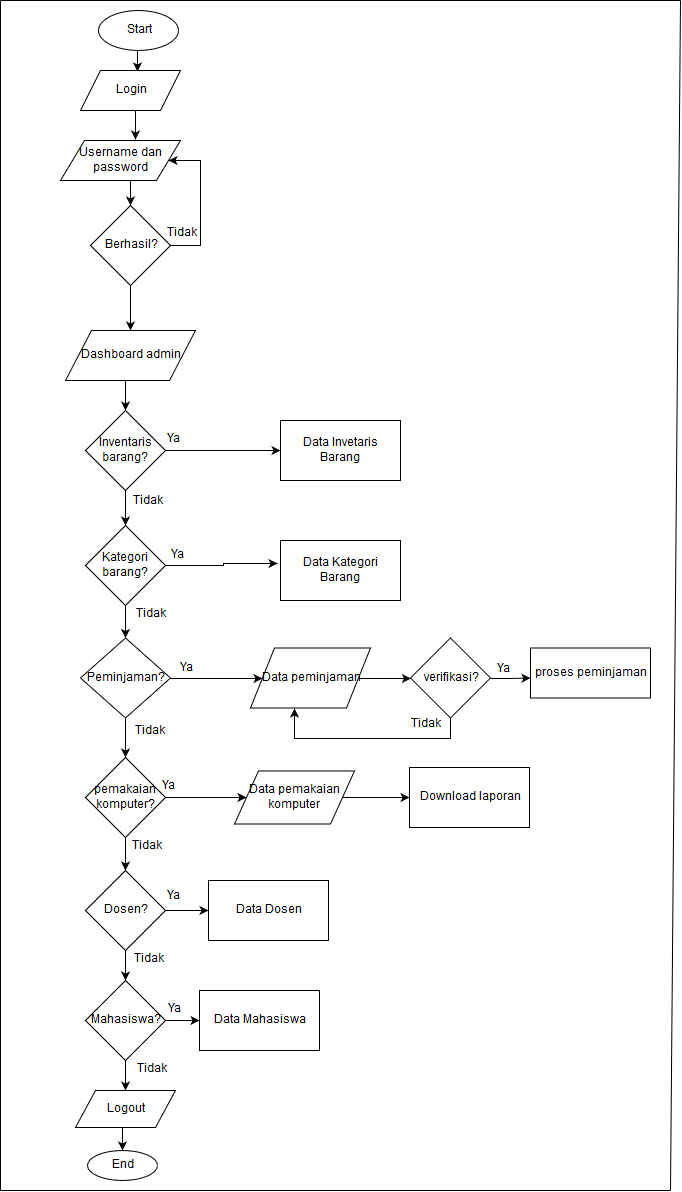
Gambar 3. 11 *Flowchart System* Admin

1. ***Flowchart System Mahasiswa***

****

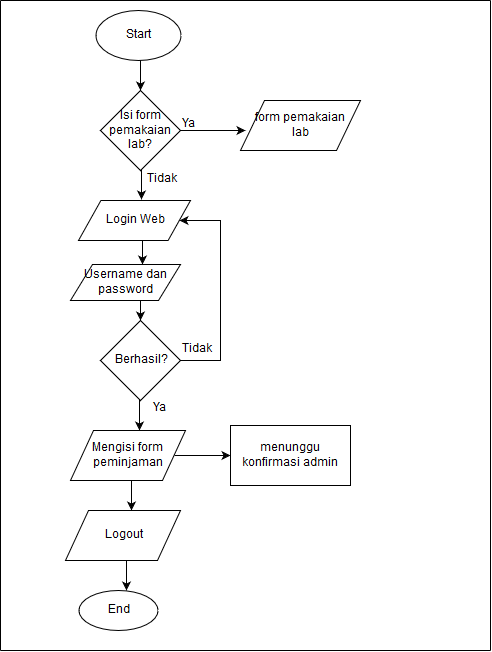
Gambar 3. 12 *Flowchart System* Mahasiswa

1. ***Flowchart Program***
2. ***Flowchart* Admin**

****

Gambar 3. 13 *Flowchart* Admin

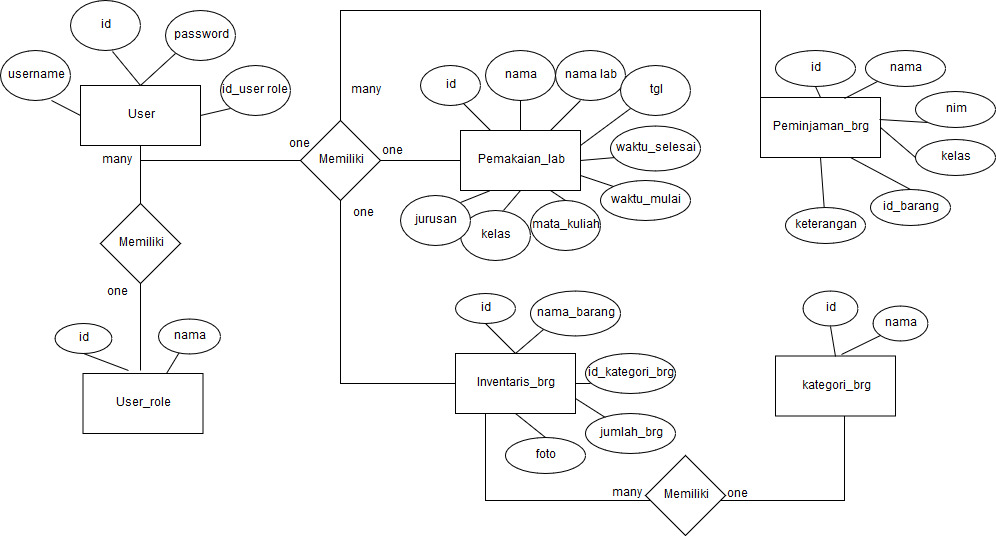
1. ***Flowchart* Mahasiswa**

****

Gambar 3. 14 *Flowchart* Mahasiswa

1. ***Entity Relationship Diagram***

ERD adalah salah satu model yang digunakan untuk mendesain database dengan tujuan menggambarkan data yang berelasi pada sebuah database. ERD mempunyai objek-objek yang saling berkaitan dan berelasi. Dalam bagan ini terdapat tabel user, user role, pemakaian lab, iventaris barang, kategori barang, dan peminjaman barang yang saling berelasi agar jelas nya dapat dilihat pada Gambar 3.15 *Entity Relationship Diagram*.



Gambar 3. 15 *Entity Relationship Diagram*

1. **Peracangan Antar Muka (*Interface*)**
2. **Rancangan Halaman *Login***

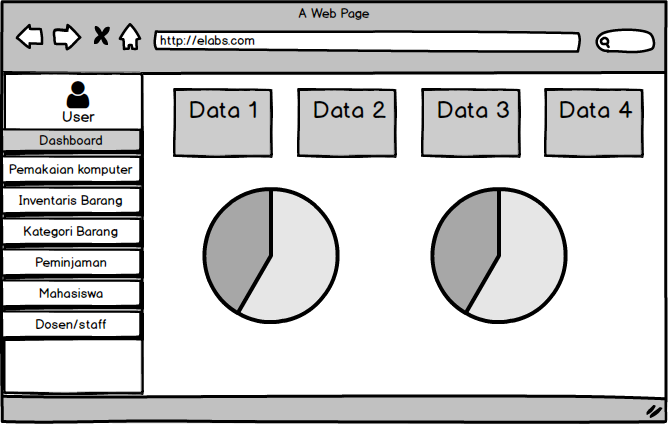
Berikut ini adalah rancangan tampilan halaman *login* dapat dilihat pada Gambar 3.16 Halaman rancangan *Login*.



Gambar 3. 16 Halamanrancangan *Login*

1. **Rancangan Halaman *Dashboard* Admin**

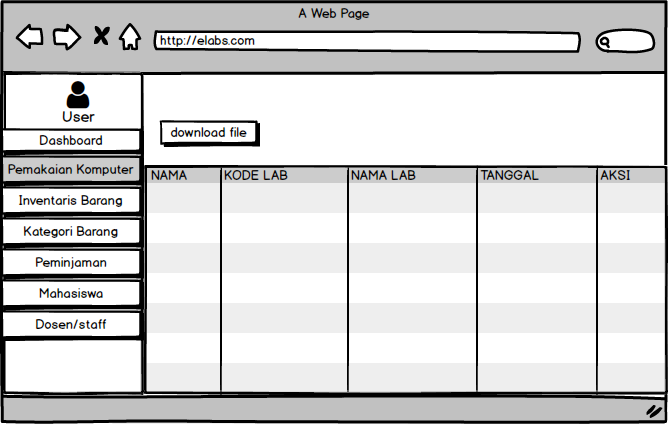
Berikut ini adalah rancangan tampilan halaman *dashboard* pada admin dapat dilihat pada Gambar 3.17 Halaman rancangan *Dashboard* Admin.



Gambar 3. 17 Halaman rancangan *Dashboard* Admin

1. **Rancangan Halaman Pemakaian Komputer**

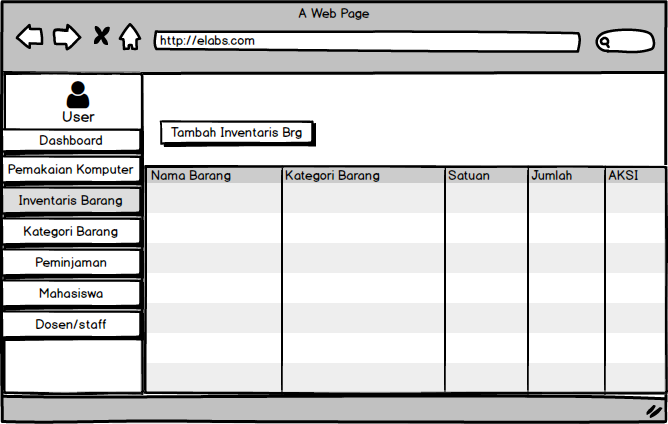
Berikut ini adalah rancangan tampilan halaman tabel pada data pemakaian komputer di laboratorium dapat dilihat pada Gambar 3.18 Halaman rancangan Data Pemakaian Komputer.



Gambar 3. 18 Halaman rancangan Data Pemakaian Komputer

1. **Rancangan Halaman Invetaris Barang**

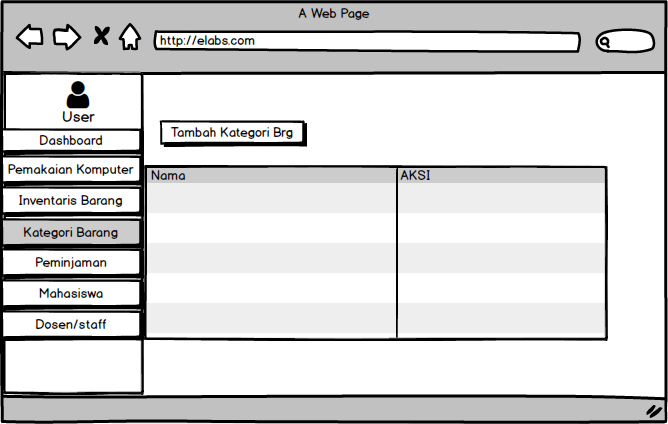
Berikut ini adalah rancangan tampilan halaman tabel pada data inventaris barang dapat dilihat pada Gambar 3.19 Halaman rancangan Inventaris Barang.



Gambar 3. 19 Halaman rancangan Inventaris Barang

1. **Rancangan Halaman Kategori Barang**

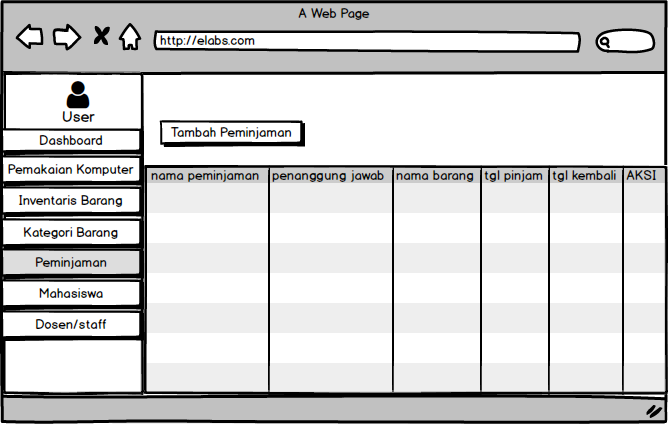
Berikut ini adalah rancangan tampilan halaman tabel pada data kategori barang dapat dilihat pada Gambar 3.20 Halaman rancangan Kategori Barang.



Gambar 3. 20 Halaman rancangan Kategori Barang

1. **Rancangan Halaman Peminjaman**

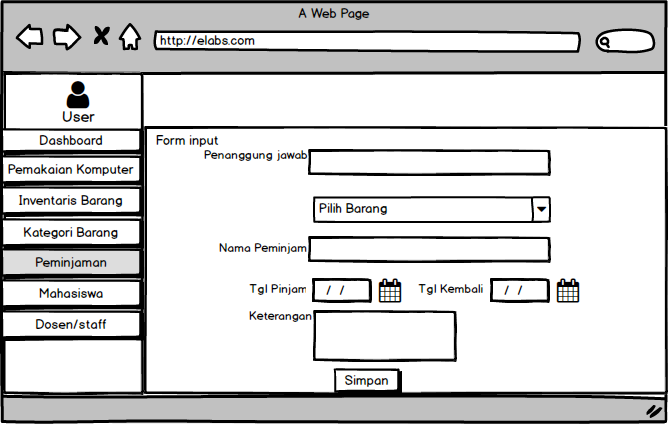
Berikut ini adalah rancangan tampilan halaman tabel pada data peminjaman barang dapat dilihat pada Gambar 3.21 Halaman rancangan Peminjaman.



Gambar 3. 21 Halaman rancangan Peminjaman

1. **Rancangan Halaman Form Peminjaman**

Berikut ini adalah rancangan tampilan halaman form peminjaman barang dapat dilihat pada Gambar 3.22 Halaman rancangan *Form* Peminjaman.

****

Gambar 3. 22 Halaman rancangan *Form* Peminjaman

1. **Rancangan Halaman *Form* *User* untuk pemakaian lab**

Pada rancangan halaman *form user* untuk pemakaian lab terdapat pada dekstop yang nanti nya terhubung pada *website*



Gambar 3. 23 Rancangan Halaman *User* dekstop

1. **Perancangan *Database***
2. **Tabel *User***

Pada Tabel *user* digunakan untuk menyimpan data yang terhubung dengan *user* yaitu mahasiswa dan rancangan *database* tabel *user* dapat dilihat pada Tabel

Tabel 3. 4 Rancangan Tabel *User*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | Id | Int(11) | Sebagai *primary key*, id merupakan identitas utama dari tabel *user*. |
| 2 | Username | Varchar(255) | Untuk menyimpan nama *user*. |
| 3 | Password | Varchar(255) | Untuk menyimpan *password user.* |
| 4 | Id\_mhs | Varchar(11) | Untuk menyimpan relasi mhs ke *user*. |
| 5 | Id\_user\_role | Varchar(11) | Untuk menyimpan relasi *user role* ke *user.* |
| 6 | Token | Varchar(100) | Untuk menyimpan foto *user .* |

1. **Tabel mahasiswa**

Pada tabel mahasiswa digunakan untuk menyimpan data mahasiswa dan rancangan *database* tabel mahasiswa dapat dilihat pada Tabel 3.5 Rancangan Tabel Mahasiswa.

Tabel 3. 5 Rancangan Tabel Mahasiwa

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | Id | Int(11) | Sebagai *primary key*, id merupakan identitas utama dari tabel Mahasiswa. |
| 2 | Nama | Varchar(255) | Untuk menyimpan nama. |
| 3 | Id\_jurusan | Int(11) | Untuk menyimpan relasi tabel jurusan ke tabel mahasiswa*.* |
| 4 | Nim | Varchar(11) | Untuk menyimpan nim. |
| 5 | Foto | Varchar(11) | Untuk menyimpan berkas foto. |

1. **Tabel dosen staf**

Pada tabel dosen staf digunakan untuk menyimpan data dosen staf dan rancangan *database* tabel dosen staf dapat dilihat pada Tabel 3.6 Rancangan Tabel Dosen Staf.

Tabel 3. 6 Rancangan Tabel Dosen Staf

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | Id | Int(11) | Sebagai *primary key*, id merupakan identitas utama dari tabel Dosen staf. |
| 2 | Nidn | Varchar(100) | Untuk menyimpan nidn. |
| 3 | Nip | Varchar(100) | Untuk menyimpan nip*.* |
| 4 | Nama | Varchar(255) | Untuk menyimpan nama. |
| 5 | Jabatan | Varchar(255) | Untuk menyimpan jabatan. |

1. **Tabel Inventaris Barang**

Pada tabel inventaris barang digunakan untuk menyimpan data inventaris barang dan rancangan *database* tabel inventaris barang dapat dilihat pada Tabel 3.7 Rancangan Tabel inventaris barang.

Tabel 3. 7 Rancangan Tabel inventaris barang

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | Id | Int(11) | Sebagai *primary key*, id merupakan identitas utama dari tabel Inventaris Barang. |
| 2 | Nama Barang | Varchar(255) | Untuk menyimpan nama barang. |
| 3 | Kategori Barang | Varchar(100) | Untuk menyimpan kategori barang*.* |
| 4 | Jumlah Barang | Varchar(255) | Untuk menyimpan jumlah barang.. |
| 5 | Foto | Varchar(255) | Untuk menyimpan berkas foto. |

1. **Tabel Pemakaian Laboratorium**

Pada tabel pemakaian lab digunakan untuk menyimpan data pemakaian lab dan rancangan *database* tabel pemakaian lab dapat dilihat pada Tabel 3.8 pemakaian lab.

Tabel 3. 8 Rancangan Tabel Pemakaian Lab

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | Id | Int(11) | Sebagai *primary key*, id merupakan identitas utama dari tabel Pemakaian Lab. |
| 2 | Kode Lab | Varchar(4) | Untuk menyimpan kode lab. |
| 3 | Nama Lab | Varchar(255) | Untuk menyimpan nama lab. |
| 4 | Mata Kuliah | Varchar(255) | Untuk menyimpan mata kuliah. |
| 5 | Tanggal | Date | Untuk menyimpan tanggal. |
| 6 | Nama | Varchar(255) | Untuk menyimpan nama. |
| 7 | Kelas | Varchar(255) | Untuk menyimpan kelas. |

1. **Tabel Peminjaman**

Pada tabel peminjaman digunakan untuk menyimpan data peminjaman dan rancangan *database* tabel peminjaman dapat dilihat pada Tabel 3.9 Rancangan Tabel Peminjaman.

Tabel 3. 9 Rancangan Tabel Peminjaman

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | Id | Int(11) | Sebagai *primary key*, id merupakan identitas utama dari tabel Peminjaman. |
| 2 | Id dosen staf | Int (11) | Untuk menyimpan relasi tabel dosen staf ke tabel peminjaman*.* |
| 3 | Id invetaris barang | Int(11) | Untuk menyimpan relasi tabel inventaris barang ke tabel peminjaman*.* |
| 4 | Id Mahasiswa | Int (11) | Untuk menyimpan relasi tabel mahasiswa ke tabel peminjaman. |
| 5 | Tanggal pinjam | Date | Untuk menyimpan tanggal. |
| 6 | Tanggal kembali | Date | Untuk menyimpan tanggal. |
| 7 | Keterangan | Varchar(255) | Untuk menyimpan data keterangan. |

1. **Tabel *User Role***

Pada tabel *user role* digunakan untuk menyimpan data *user role* untuk membuat level pada *user* dan rancangan *database* tabel *user role* dapat dilihat pada Tabel 3.10 Rancangan Tabel *User Role.*

Tabel 3. 10 Rancangan Tabel *User Role*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | Id | Int(11) | Sebagai *primary key*, id merupakan identitas utama dari tabel *User Role*. |
| 2 | Nama | Varchar (255) | Untuk menyimpan nama *User Role.* |

1. **Tabel Jurusan**

Pada tabel jurusan digunakan untuk menyimpan data jurusan untuk membuat relasi di tabel mahasiswa dan rancangan *database* tabel jurusan dapat dilihat pada tabel 3.11 Rancangan tabel jurusan.

Tabel 3. 11 Rancangan Tabel Jurusan

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | Id | Int(11) | Sebagai *primary key*, id merupakan identitas utama dari tabel Jurusan. |
| 2 | Nama | Varchar(255) | Untuk menyimpan nama jurusan*.* |

1. **Tabel Kategori Barang**

Pada tabel kategori barang digunakan untuk menyimpan data kategori barang untuk menampilkan relasi pada tabel inventaris barang dan rancangan *database* tabel kategori barang dapat dilihat pada Tabel 3.12 Rancangan Tabel Kategori Barang.

Tabel 3. 12 Rancangan Tabel Kategori Barang

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | Id | Int(11) | Sebagai *primary key*, id merupakan identitas utama dari tabel kategori. |
| 2 | Nama | Varchar(255) | Untuk menyimpan nama kategori*.* |

1. **Normalisasi *Database***

Pada sistem rancang bangun aplikasi pengelolaan dan perawatan laboratorium politeknik negeri indramayu ini terdapat normalisasi *database* dari beberapa tabel.

1. ***Unnormal***

Berikut ini adalah unnormal database pada tabel *user* terdapat pada tabel 3.13 *Unnormal database.*

Tabel 3. 13 *Unnormal database*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **username** | **id\_mhs** | **nim** | **Id** | **nama** | **Id\_kategori\_brg** | **nama\_brg** |
| 1 | jenia | 1 | 1603102 | 1 | Alat | 1 | mouse |
| 1 | Jenia | 1 | 1603102 | 2 | bahan | 2 | Kabel |
| 2 | Shafa | 2 | 1603083 | 1 | Alat | 3 | Keyboard |
| 2 | shafa | 2 | 1603083 | 2 | Bahan | 4 | Rj45 |
| 3 | silvi | 3 | 1603088 | 1 | Alat | 5 | Pc |
| 3 | silvi | 3 | 1603088 | 2 | Bahan | 6 | led |

1. **Tabel Normalisasi Pertama (1NF) pada tabel *User***

Proses selanjutnya terdapat bentuk normalisasi pertama dari tabel *unnormal.* Tabel normalisasi kesatu dapat dilihat pada tabel 3.14 Normalisasi 1NF tabel *user*.

Tabel 3. 14 normalisasi pertama tabel *user*

|  |  |  |
| --- | --- | --- |
| **id\_mhs** | **username** | **nim** |
| 1 | jenia | 1603102 |
| 1 | Jenia | 1603102 |
| 2 | Shafa | 1603083 |
| 2 | shafa | 1603083 |
| 3 | silvi | 1603088 |
| 3 | silvi | 1603088 |

Lalu setelah terbentuk normalisasi pertama pada tabel *user* maka terbentuk hasil 1NF pada normalisasi pertama tabel *user* dapat dilihat pada tabel 3.15 normalisasi 1NF*.*

Tabel 3. 15 normalisasi 1NF

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **nim** | **username** | **Id** | **nama** | **Id\_kategori\_brg** | **nama\_brg** |
| 1603102 | Jenia | 1 | Alat | 1 | Mouse |
| 1603102 | Jenia | 2 | bahan | 2 | Kabel |
| 1603083 | Shafa | 1 | Alat | 3 | Keyboard |
| 1603083 | Shafa | 2 | Bahan | 4 | Rj45 |
| 1603088 | Koriah | 1 | Alat | 5 | PC |
| 1603088 | koriah | 2 | Bahan | 6 | led |

1. **Tabel Normalisasi Kedua (2NF) pada tabel Inventaris Barang**

Dari normalisasi pertama proses selanjutnya normalisasi kedua (2NF). Tabel normalisasi kedua dapat dilihat pada tabel 3.16 normalisasi kedua kategori.

Tabel 3. 16 Normalisasi kedua kategori barang

|  |  |
| --- | --- |
| **Id** | **nama** |
| 1 | Alat |
| 2 | bahan |

Lalu setelah terbentuk normalisasi kedua pada tabel kategori barangmaka terbentuk hasil 2NF pada normalisasi kedua tabel inventaris barang dapat dilihat pada tabel 3.17 normalisasi 2NF*.*

Tabel 3. 17 normalisasi 2NF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **nim** | **username** | **Id\_kategori\_brg** | **nama** | **nama\_brg** |
| 1603102 | Jenia | 1 | Alat | Mouse |
| 1603102 | Jenia | 2 | bahan | Kabel |
| 1603083 | Shafa | 3 | Alat | Keyboard |
| 1603083 | Shafa | 4 | Bahan | Rj45 |
| 1603088 | Koriah | 5 | Alat | PC |
| 1603088 | koriah | 6 | Bahan | led |

**BAB IV**

**HASIL DAN PEMBAHASAN**

1. **Hasil Penelitian**

Pada penelitian ini menghasilkan sebuah aplikasi pengelolaan dan perawatan laboratorium di Politeknik Negeri Indramayu yang dapat digunakan untuk mengelola data pemakaian komputer pada laboratorium secara terpusat yang dilakukan pada saat memakai komputer di ruang laboratorium untuk mahasiswa maupun pengguna dari mana saja, serta mahasiswa dapat melakukan peminjaman barang pada laboratorium *maintenance*. Sedangkan admin melakukan pengelolaan data pemakaian komputer di laboratorium, peminjaman barang, data mahasiswa, data dosen maupun staf, dan data inventaris barang. Penulis dapat menyimpulkan pada rancang bangun aplikasi pengelolaan dan perawatan laboratorium adalah aplikasi yang digunakan untuk mengisi *logbook* komputer dan melakukan kegiatan peminjaman barang.

1. **Cara Kerja Sistem**

Prinsip kerja dari aplikasi web yang dibuat yaitu dapat di akses pada sebuah web browser yang mempunyai local *Server*. Pada aplikasi dekstop dapat diakses pada komputer yang ada di laboratorium.

1. **Pembahasan**

Setelah menganalisis, mengumpulkan data, dan merancang maka penulis mengimplementasikan semuanya untuk terciptanya aplikasi pengelolaan dan perawatan pada laboratorium.

1. **Implementasi Sistem**
2. **Implementasi Antarmuka**

Pada implementasi antarmuka adalah implementasi desain pada aplikasi. Implementasi desain interface terbagi menjadi dua yaitu pada admin dan mahasiswa. Implementasi terdapat pada website dan desktop. Berikut ini adalah implementasi antarmuka(*interface)*.

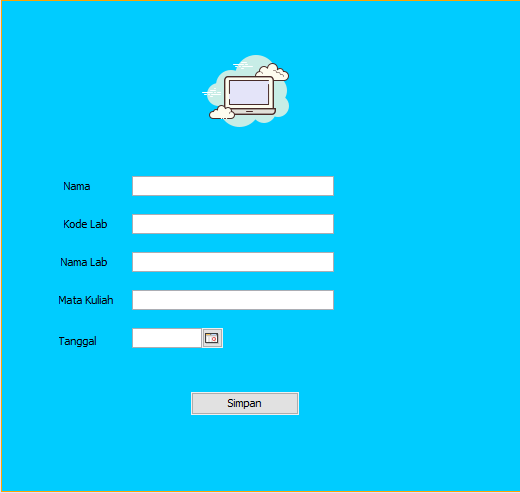
1. ***Login Website***

Halaman *login* adalah halaman yang digunakan *user* untuk *login* ke dalam sistem untuk mendapatkan hak akses *user*. Terdapat pada Gambar 4.1.

Gambar 4. 1 Halaman Dashboard Admin

1. ***Form User* Dekstop**

Pada dekstop untuk *form* *user* sebelum penggunaan komputer di lab maka *user* harus mengisi data terlebih dahulu, lalu setelah disimpan maka proses penggunaan komputer dapat dilakukan. Implementasi form user terdapat pada gambar



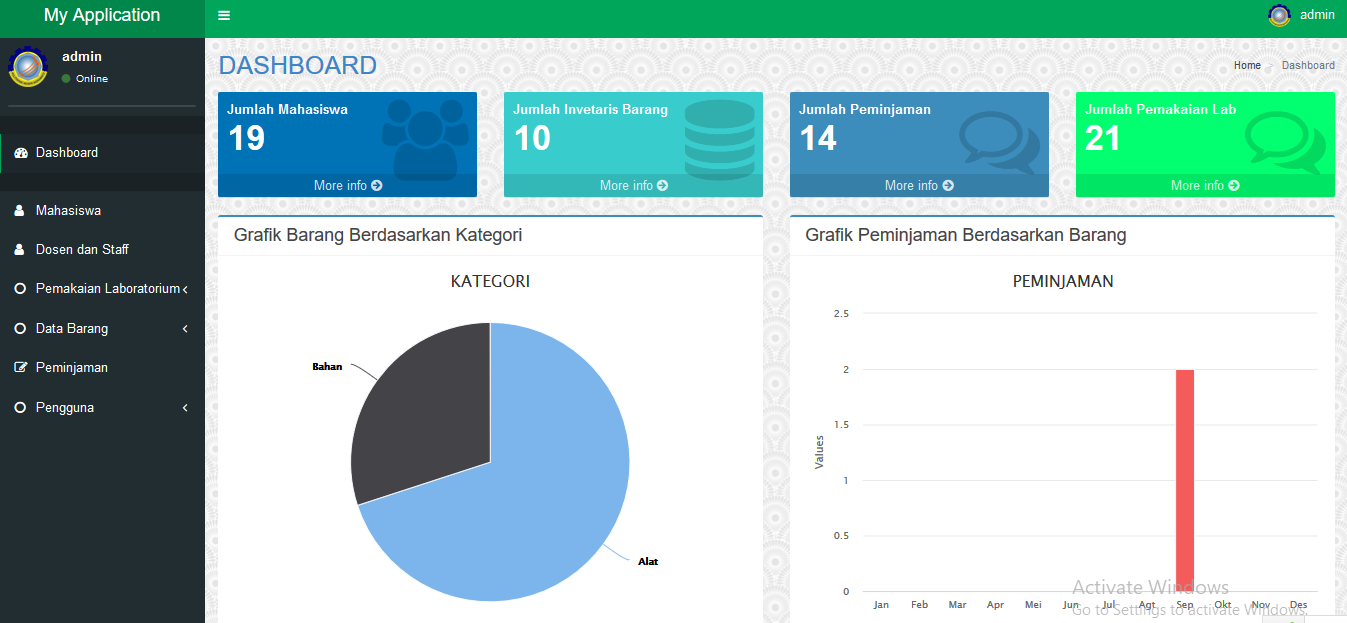
Gambar 4. 2 Halaman Form User Dekstop

1. **Implementasi Antarmuka Admin**

Halaman antarmuka (*interface*) pada admin yaitu halaman yang hanya dapat diakses oleh admin. Admin memiliki satu halaman yaitu hanya di halaman web yang digunakan untuk mengelola kegiatan pada aktivitas di aplikasi.

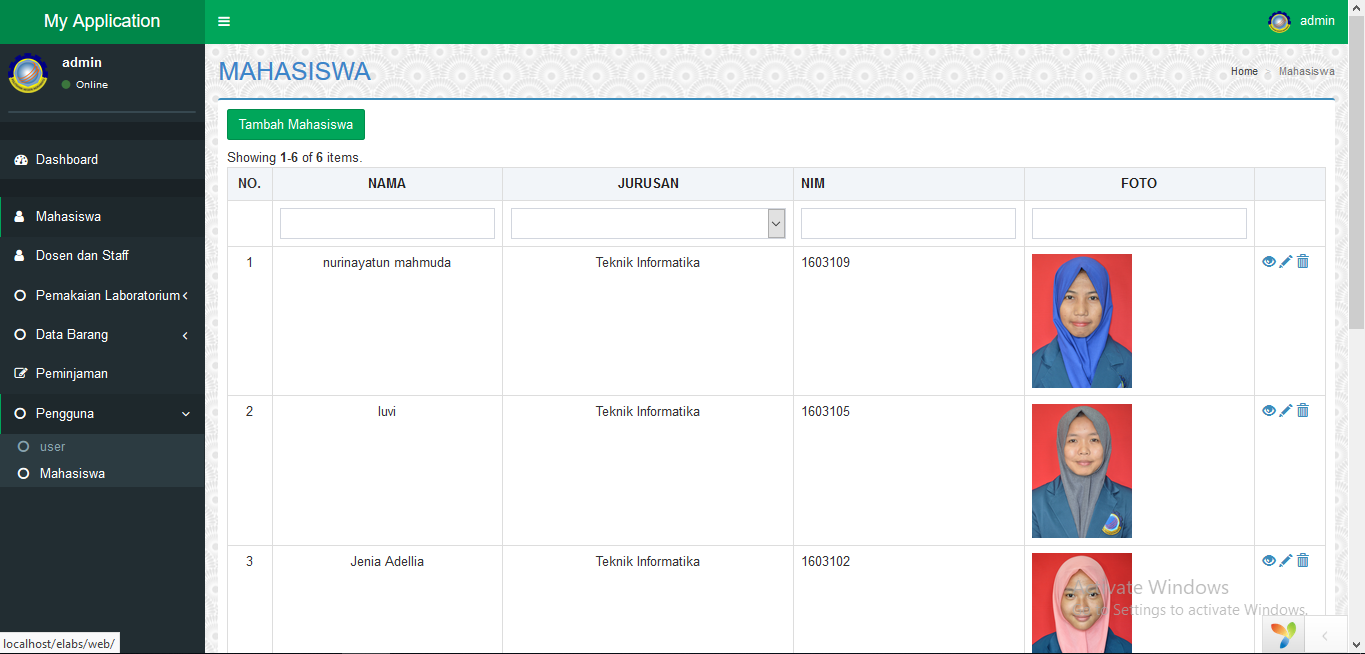
1. ***Dashboard* Admin**

Halaman *dashboard* admin yaitu halaman yang muncul ketika mengakses sebagai admin. Halaman *dashboard* admin berisi tentang grafik barang berdasarkan kategori, dan grafik peminjaman yang dilakukan berdasarkan tanggal bulan dan tahun yang masuk. Serta mahasiswa, inventaris barang, pemakaian lab data jumlah yang tersedia.

Gambar 4. 3 *Dashboard* Admin

1. **Tampilan Halaman Mahasiswa**

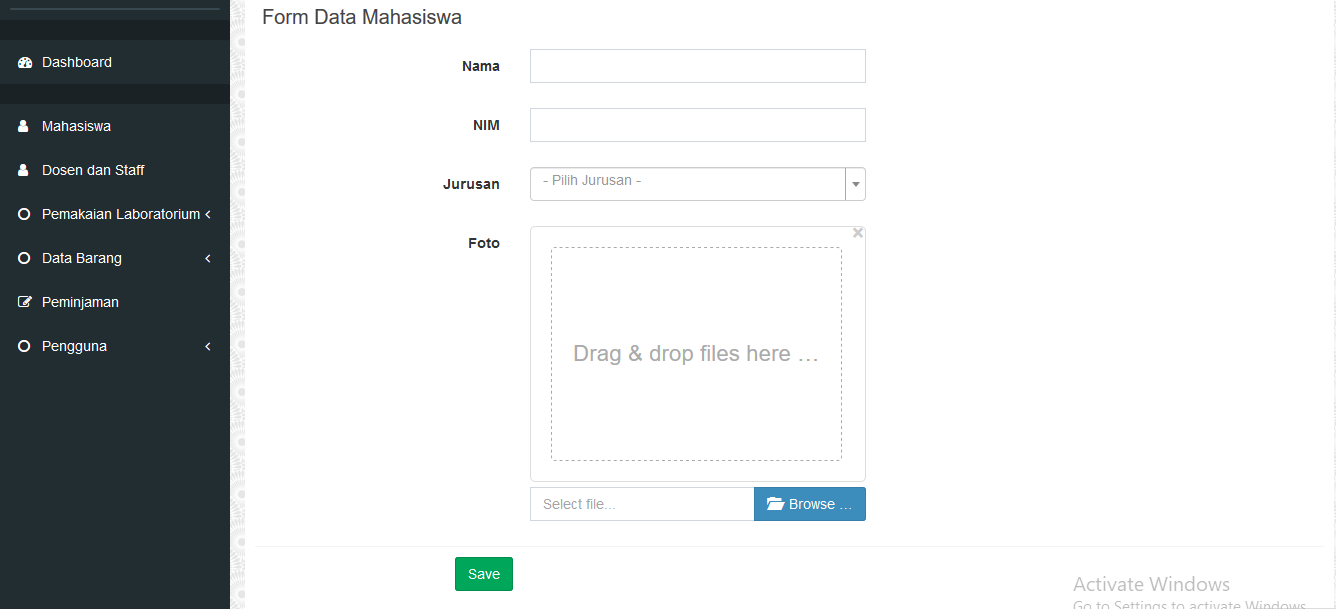
Halaman Mahasiswa yaitu halaman yang menampilkan data keseluruhan mahasiswa. Terdapat pada gambar



Gambar 4. 4 Halaman Mahasiswa

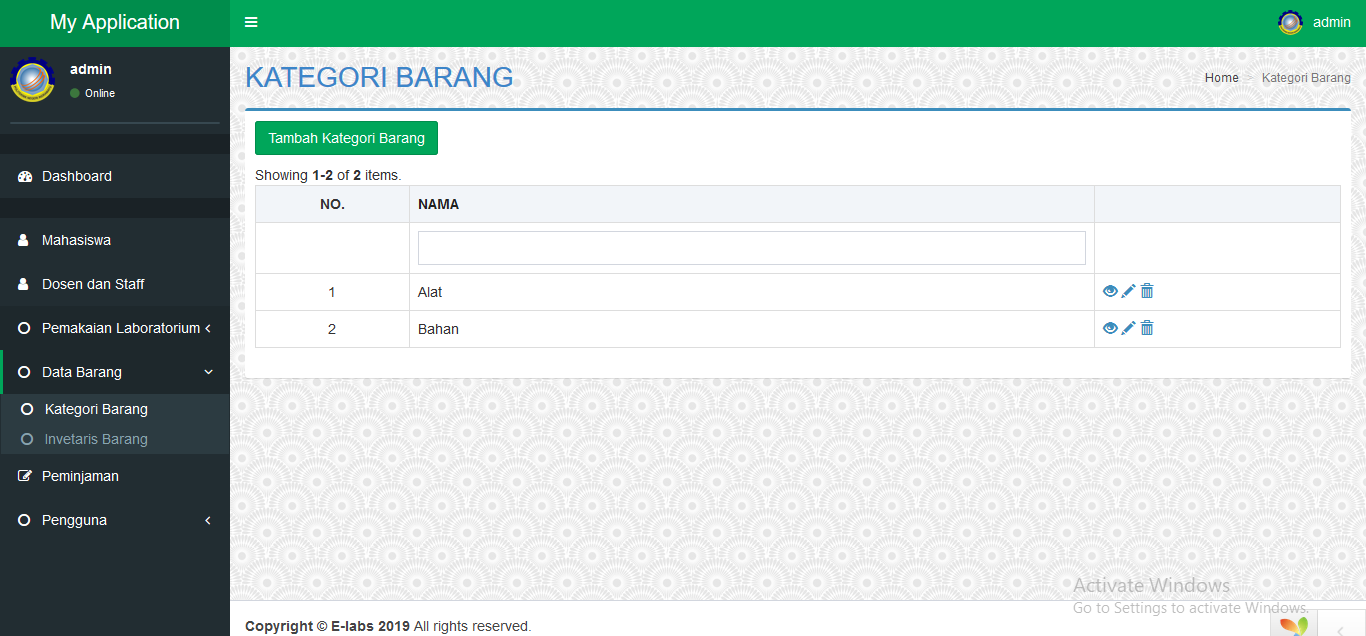
1. **Tampilan Halaman *Form* Data Mahasiswa**

Halaman *form* data mahasiswa yaitu halaman *form* untuk mengisi data mahasiswa untuk menambahkan data mahasiswa. Pada data mahasiswa terdapat nama, nim, jurusan, dan foto mahasiswa.

Gambar 4. 5 Halaman Form Data Mahasiswa

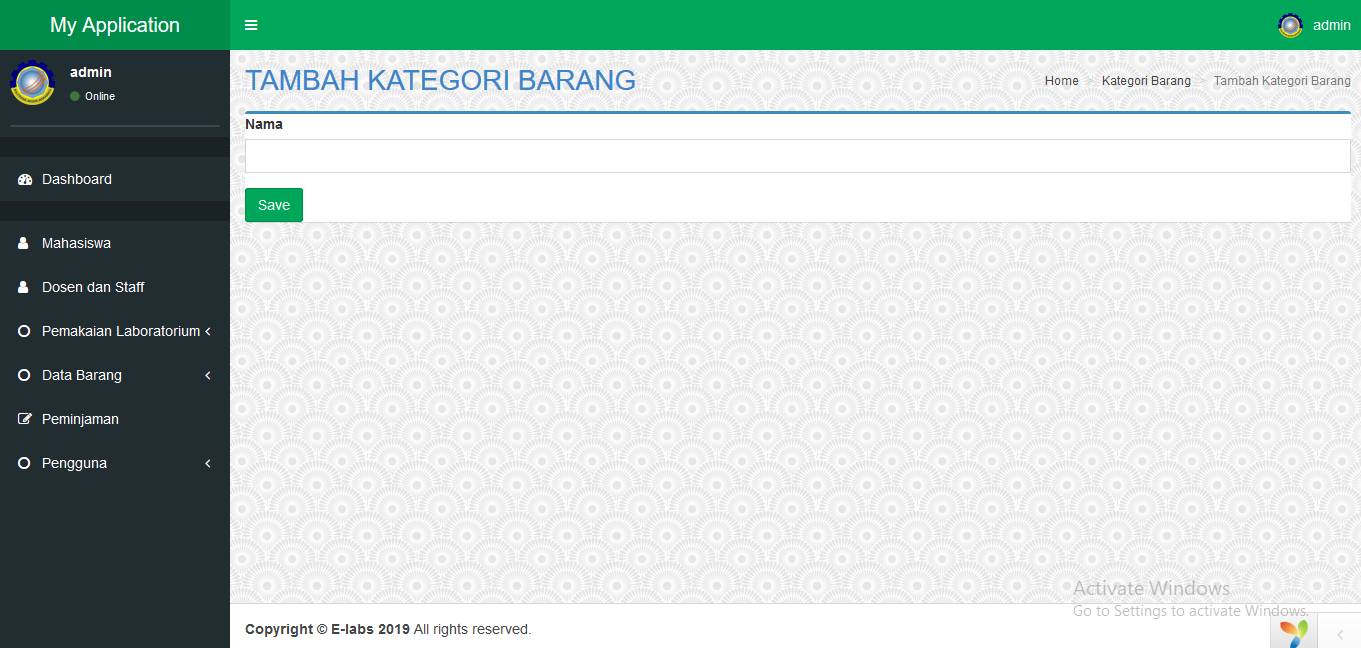
1. **Tampilan Halaman Kategori Barang**

Halaman Kategori Barang yaitu halaman yang menampilkan data kategori barang. Terdapat pada gambar

Gambar 4. 6 Halaman Kategori Barang

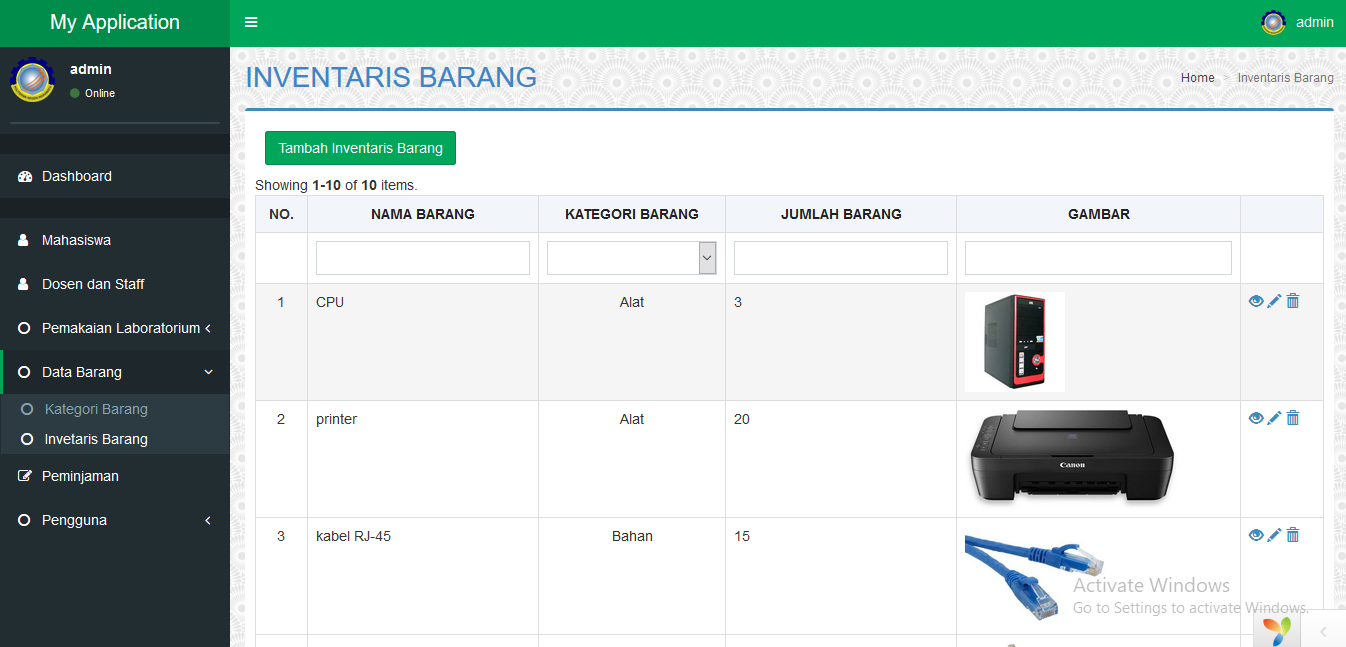
1. Tampilan Halaman Form Kategori Barang

Halaman form data kategori barang yaitu halaman form untuk mengisi data kategori untuk menambahkan data kategori barang dan untuk direlasikan pada tabel barang.

Gambar 4. 7 Halaman Form Kategori Barang

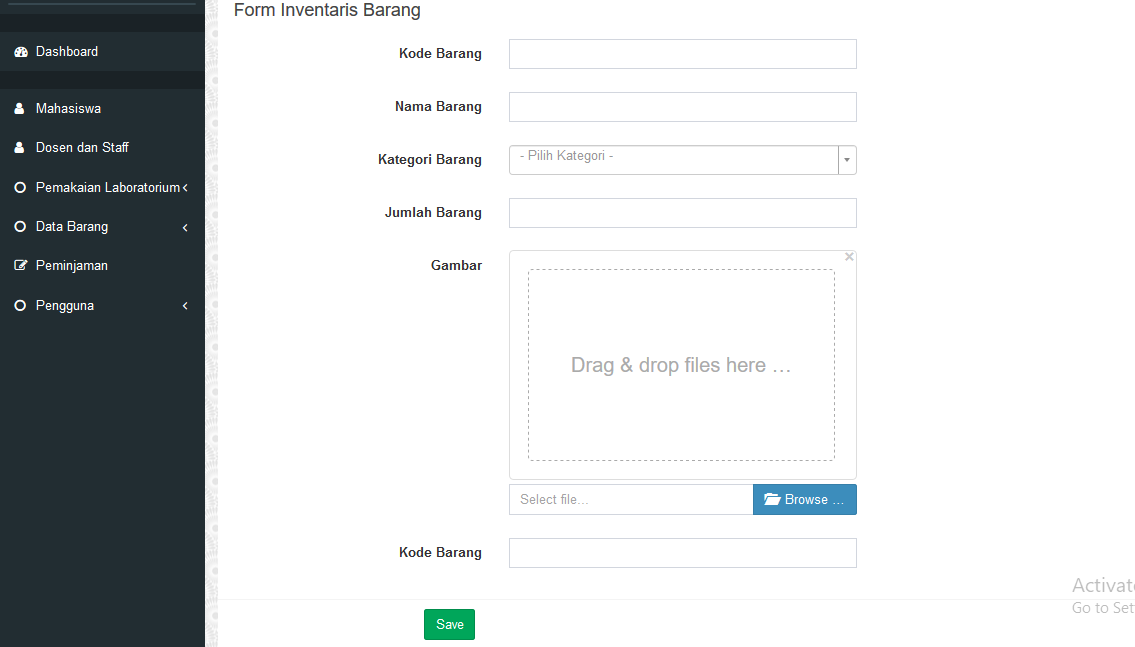
1. Tampilan Halaman Inventaris Barang

Halaman inventaris barang halaman yang menampilkan data inventaris barang secara keseluruhan. Pada field kategori barang terdapat relasi dari tabel kategori barang dengan tabel inventaris barang. Sehingga dapat mencari inventaris barang berdasarkan dari kategori barang.

Gambar 4. 8 Halaman Inventaris Barang

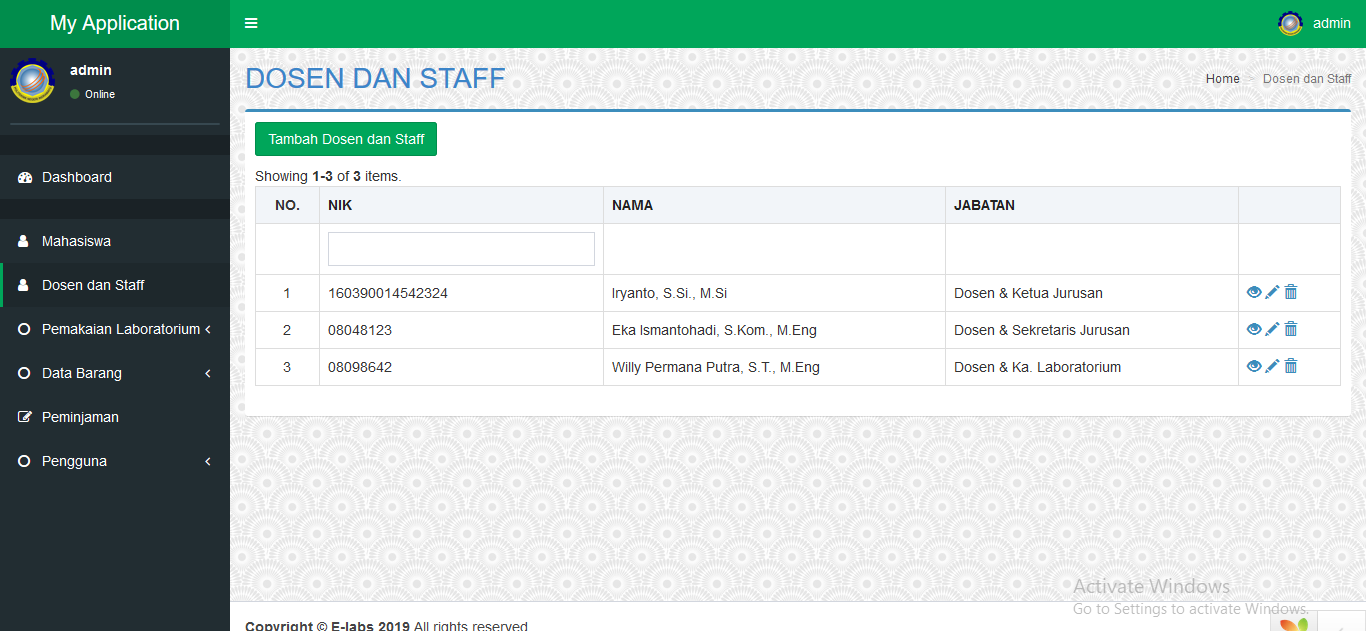
1. Tampilan Halaman Form Inventaris Barang

Halaman *form* inventaris barang yaitu halaman *form* untuk mengisi data barang untuk menambahkan data di inventaris barang. Pada kolom kode barang, kode barang di baca dan dimasukkan menggunakan alat *barcode scan* sehingga tidak perlu untuk mengisi manual pada kolom kode barang.

Gambar 4. 9 Form Inventaris Barang

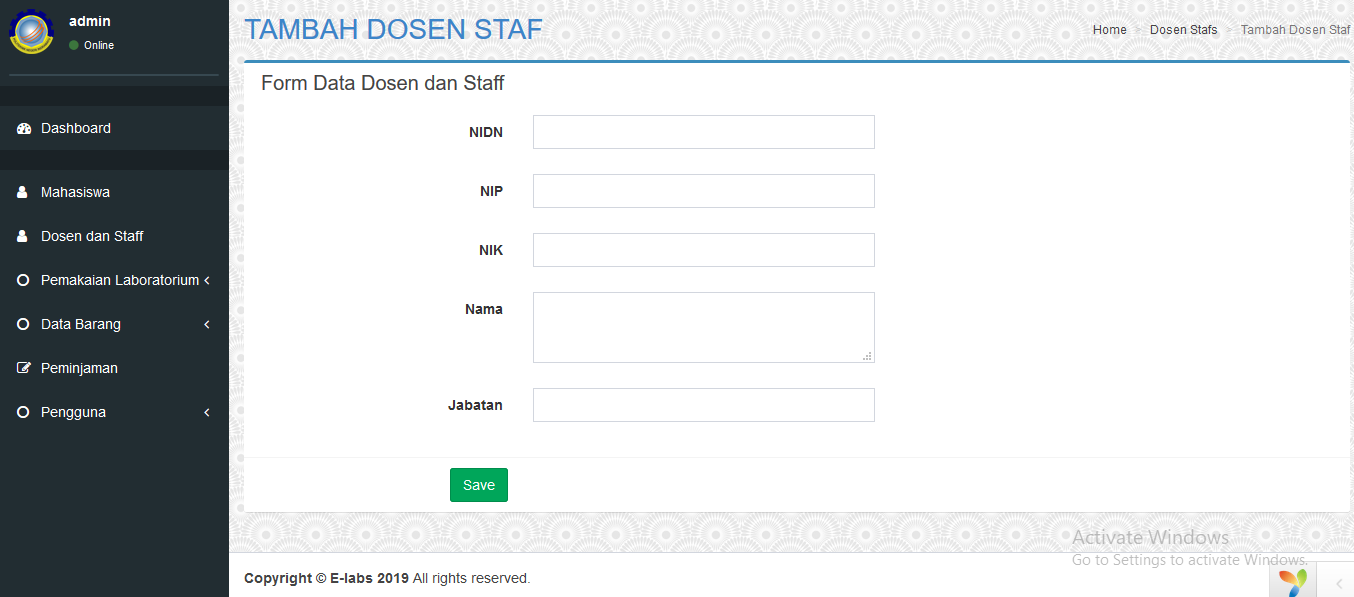
1. **Tampilan Halaman Dosen Staf**

Halaman dosen staf yaitu halaman yang menampilkan data keseluruhan pada tabel dosen dan staf. Terdapat pada gambar

Gambar 4. 10 Halaman Dosen Staf

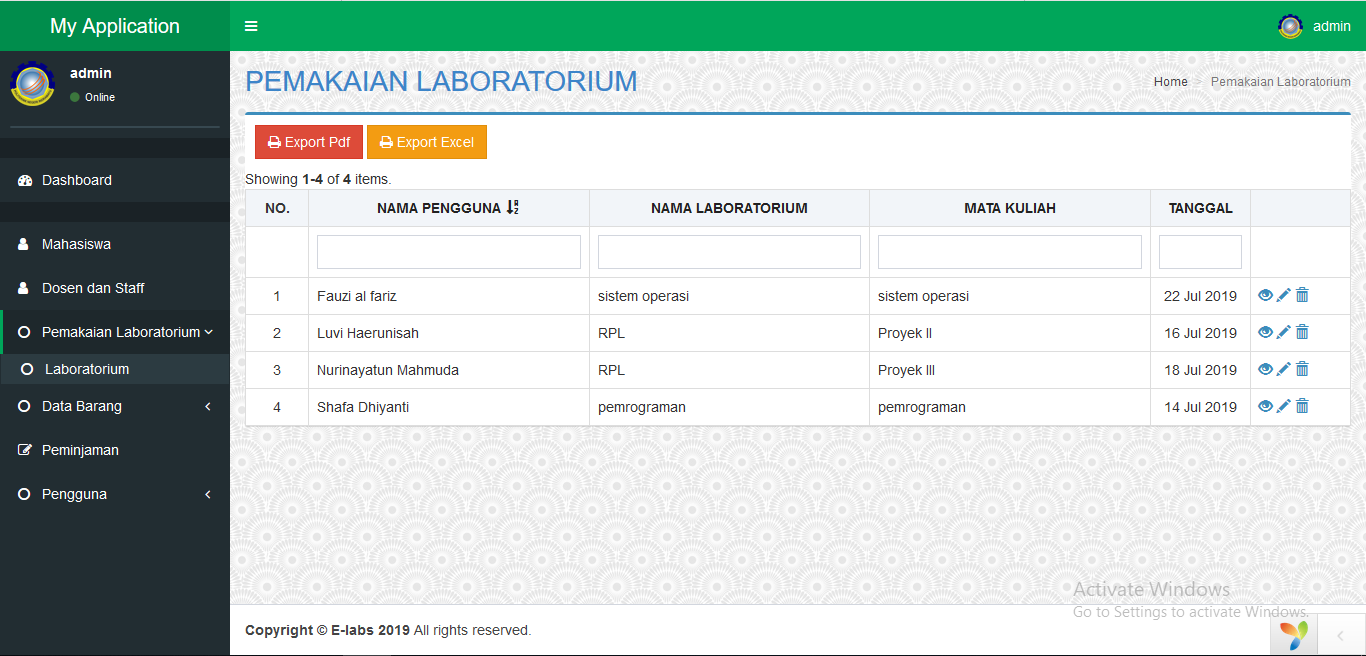
1. **Tampilan Halaman *Form* Data Dosen Staf**

Halaman *form* data dosen stad yaitu halaman *form* untuk mengisi data dosen maupun staf untuk menambahkan di data dosen staf. Pada data dosen staf terdapat NIDN (Nomor Induk Dosen Nasional), NIP (Nomor Induk Pegawai), NIK (Nomor Induk Kependudukan), nama, dan jabatan.

Gambar 4. 11 Form Dosen Staf

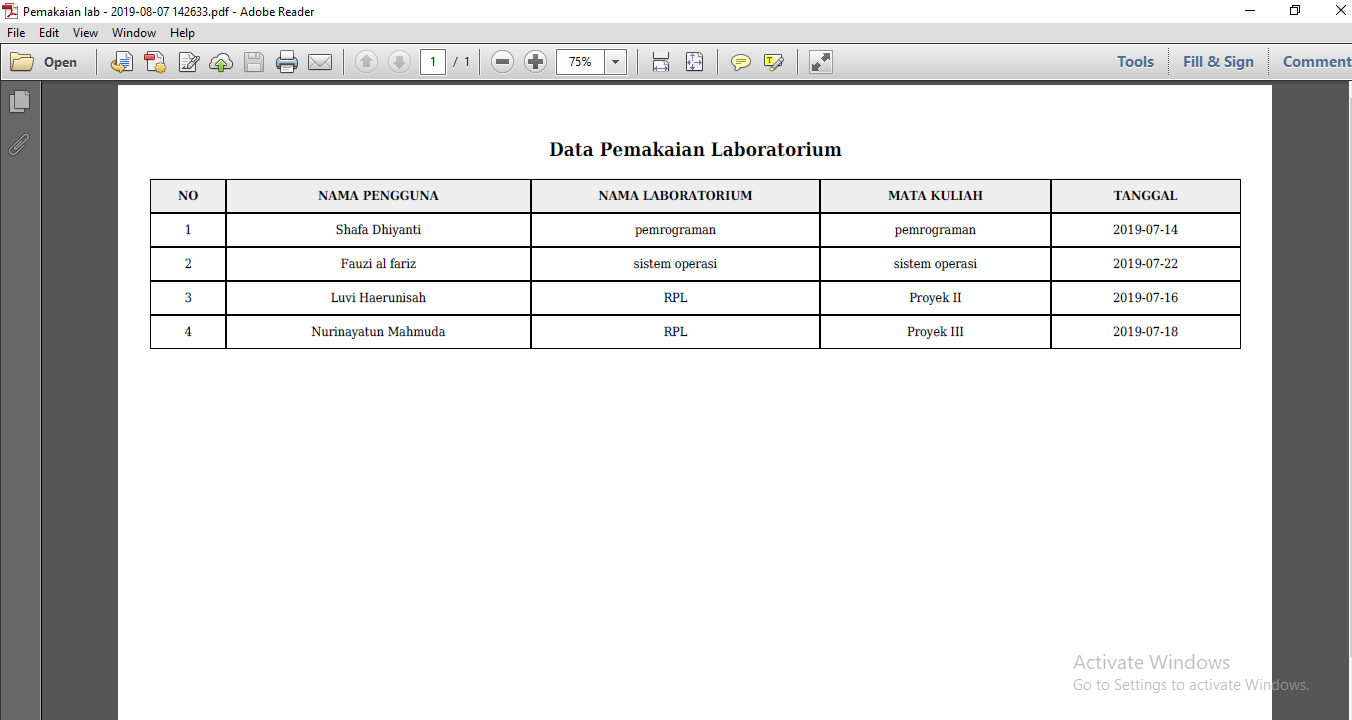
1. **Tampilan Halaman Pemakaian Komputer Laboratorium**

Halaman pemakaian komputer laboratorium merupakan halaman yang ditampilkan untuk admin di *website* yang telah diisi oleh *user* di *form user* pada dekstop. Pemakaian komputer laboratorium digunakan untuk logbook komputer.

Gambar 4. 12 Halaman Pemakaian Komputer Laboratorium

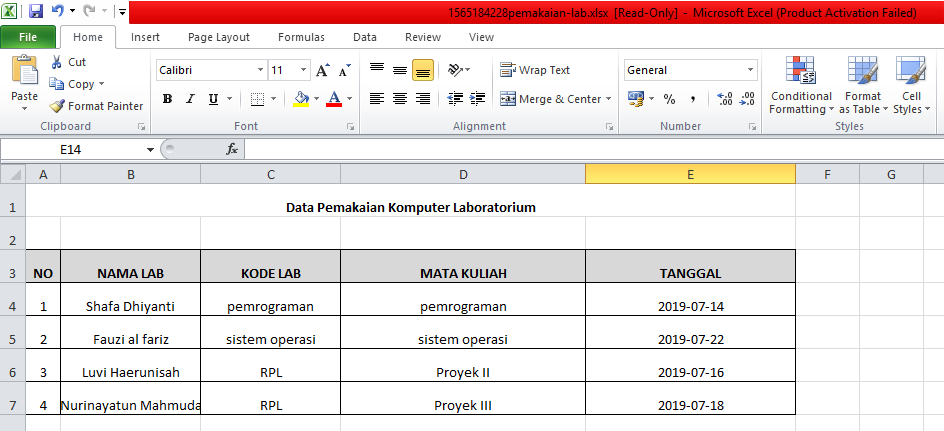
1. **Halaman *Export File* PDF Data Pemakaian Komputer**

Halaman export file PDF data diambil dari data pemakaian komputer secara keseluruhan yang digunakan untuk kebutuhan laporan.



Gambar 4. 13 *Export* PDF pemakaian komputer

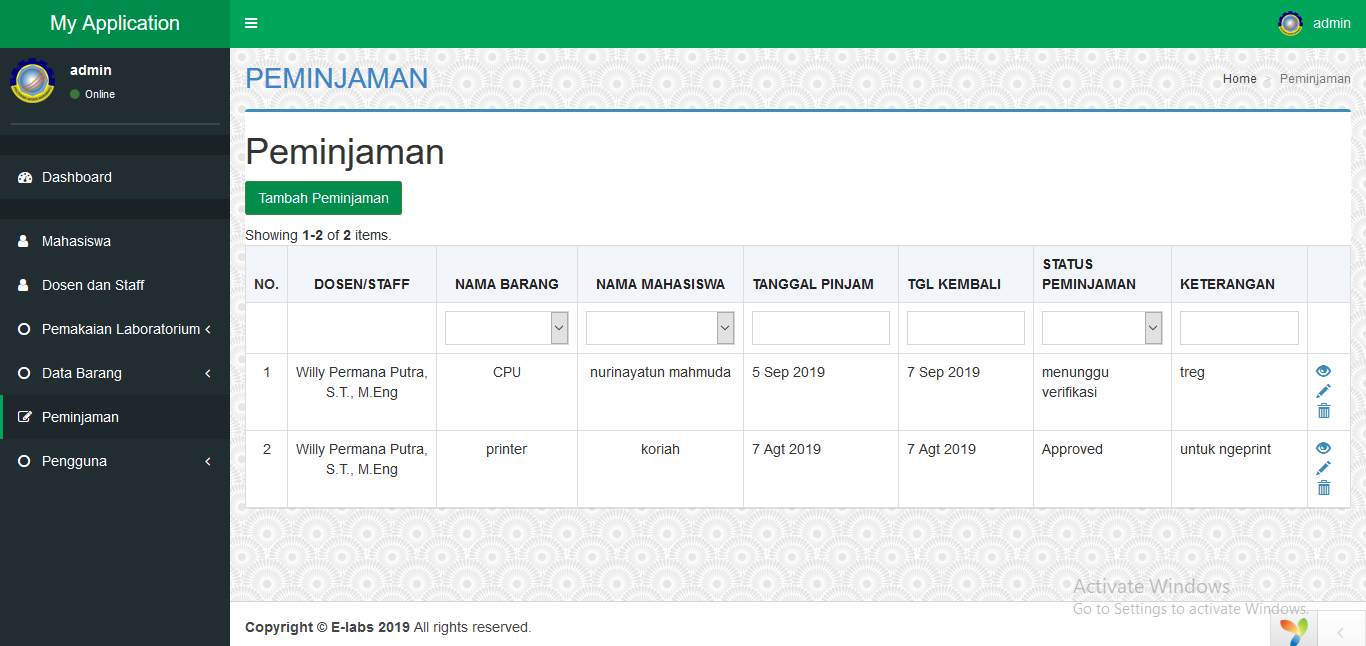
1. **Halaman *Export File* Excel Data Pemakaian Komputer**



Gambar 4. 14 *Export* Excel pemakaian komputer

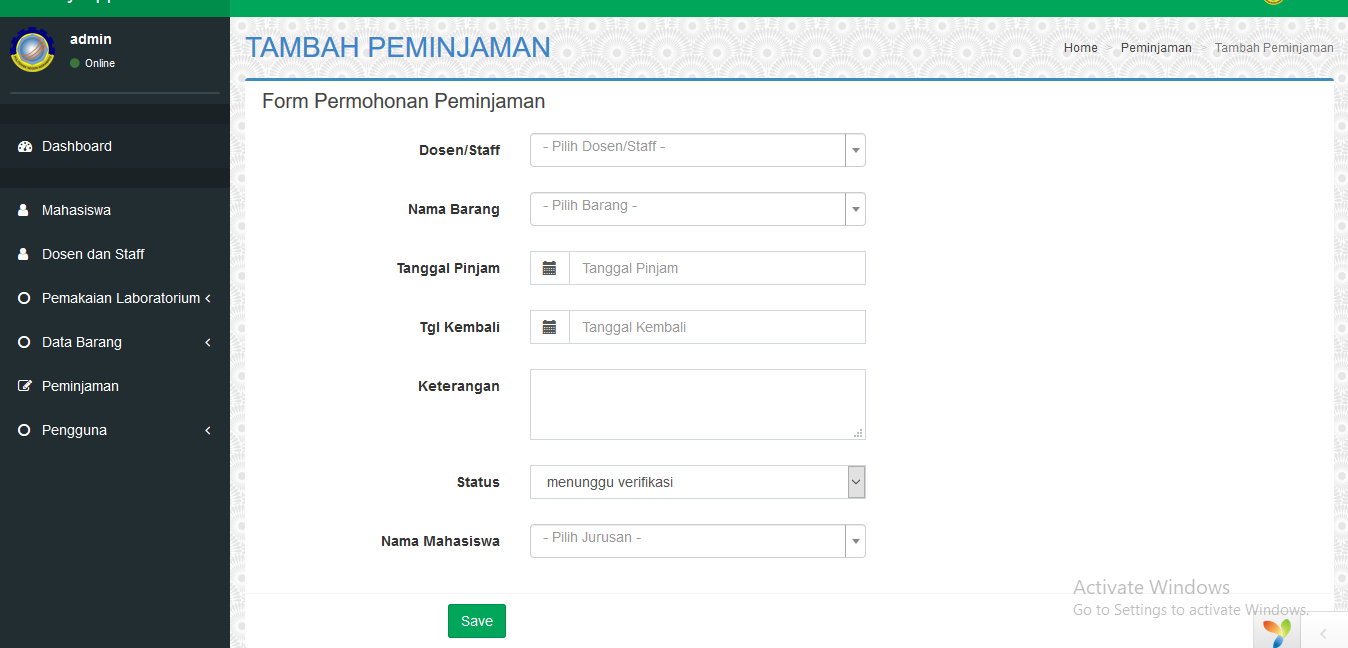
1. **Tampilan Halaman Peminjaman**

Tampilan halaman peminjaman yaitu menampilkan data keseluruhan dari berapa banyak data peminjam. Pada tabel peminjaman admin terdapat yang menangani yaitu dosen staf, barang yang dipilih, siapa yang meminjam, tanggal pinjam dan tanggal kembali, keterangan untuk keperluan peminjaman. Status peminjaman diubah pada admin apabila terdapat peminjam terbaru menunggu verifikasi maka admin dapat mengubah status nya dengan terima atau tolak.

Gambar 4. 15 Halaman Peminjaman Admin

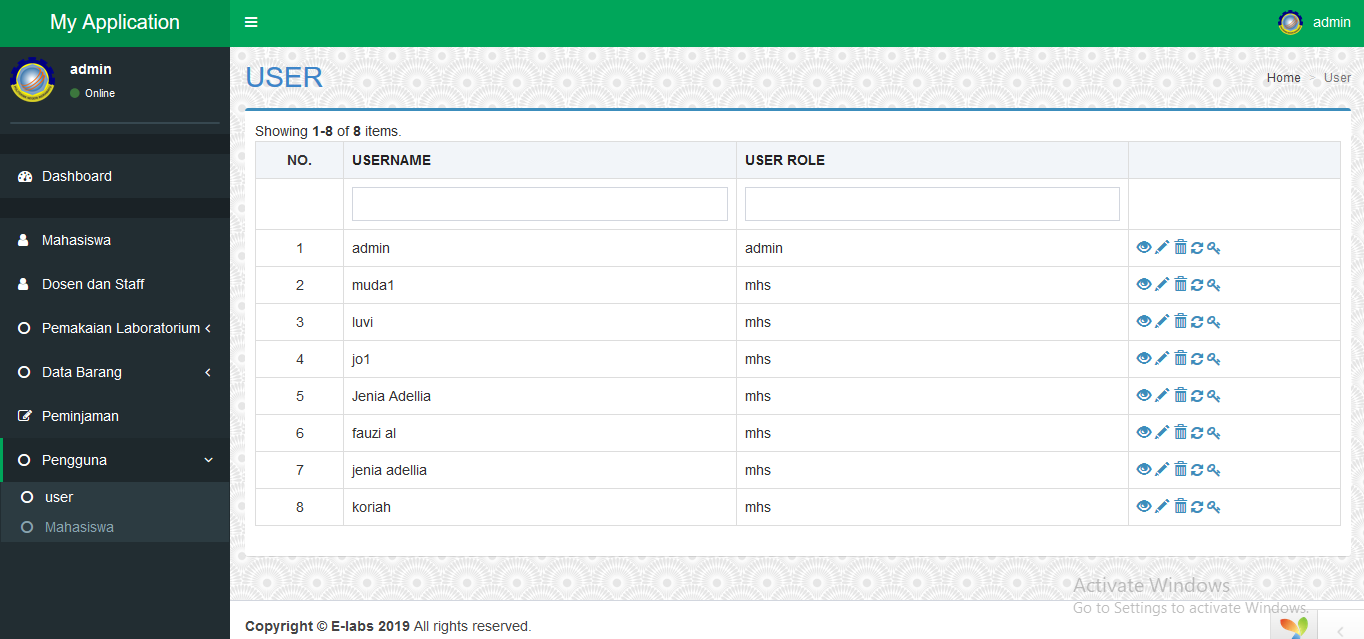
1. **Tampilan *Form* Peminjaman**

Tampilan *form* peminjaman pada admin dilakukan apabila dimasukkan oleh admin sendiri untuk peminjam.

Gambar 4. 16 *Form* Peminjaman Admin

1. **Tampilan Halaman *User***

Halaman user yaitu halaman yang menampilkan data dari tabel user. Penambahan data user terdapat pada penambahan data mahasiswa untuk bisa dapat hak akses untuk masuk ke sistem.

Gambar 4. 17 Halaman *User*

1. **Implementasi Antarmuka Mahasiswa**

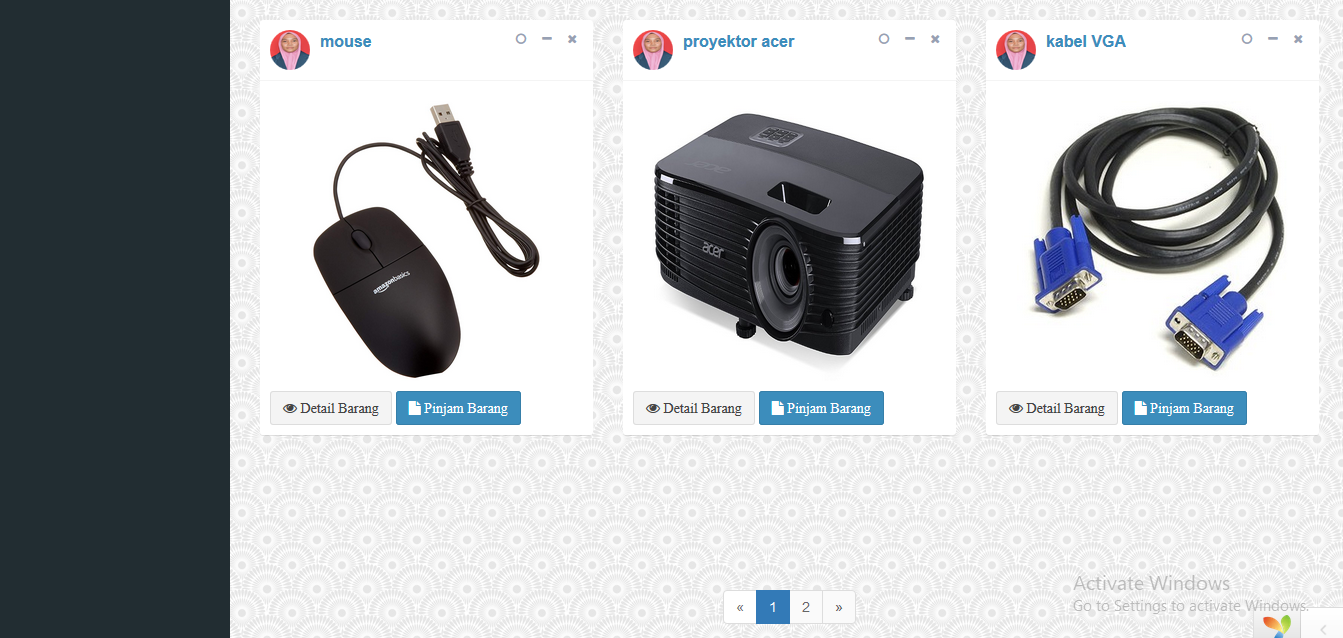
Halaman antarmuka (*interface*) pada mahasiswa yaitu halaman yang terdapat pada *website* dan form di dekstop juga bisa untuk mengisi pemakaian komputer. Mahasiswa di *website* memiliki kegiatan peminjaman barang dan pada dekstop mengisi *form* pemakaian komputer.

1. **Tampilan Halaman *Dashboard* Mahasiswa**

Halaman *Dashboard* Mahasiswa menampilkan pilihan data-data barang.

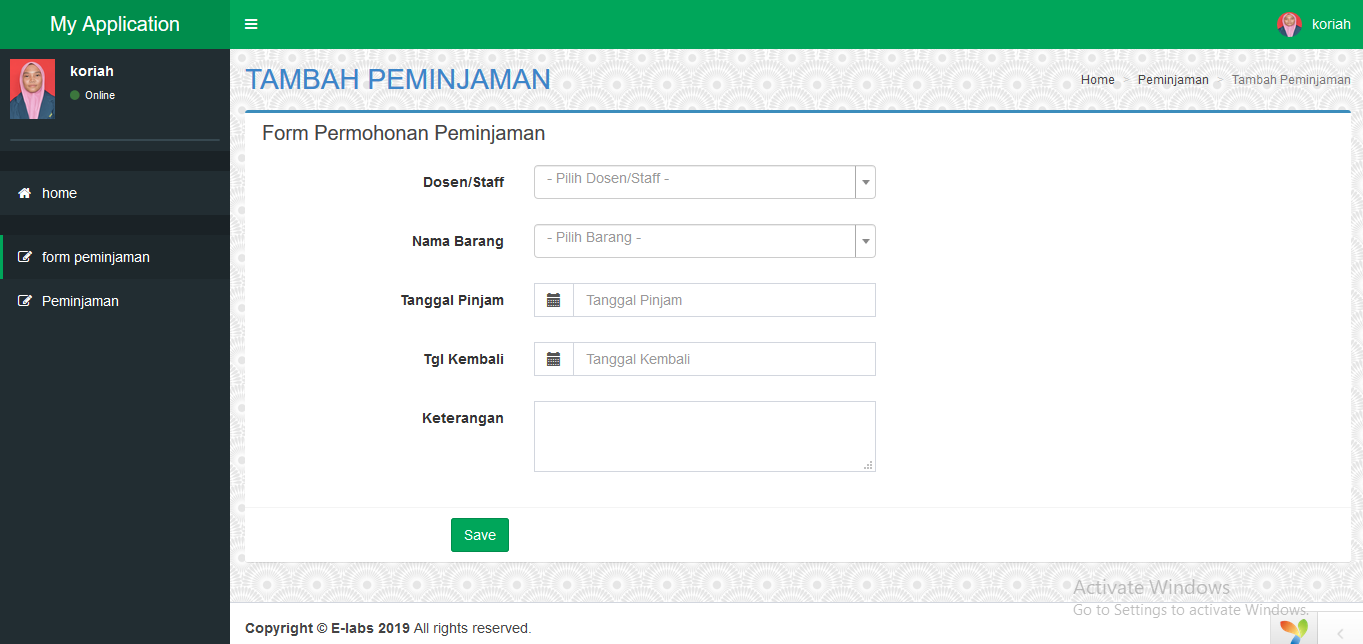
Gambar 4. 18 Halaman *Dashboard* Mahasiswa

Pada halaman *dashboard* mahasiswa, mahasiswa memilih barang yang ingin dipinjam, atau bisa cari dengan pencarian barang dengan memasukkan nama barang atau melalui pembaca kode barang.

Gambar 4. 19 *Dashboard* Mahasiswa (Lanjutan)

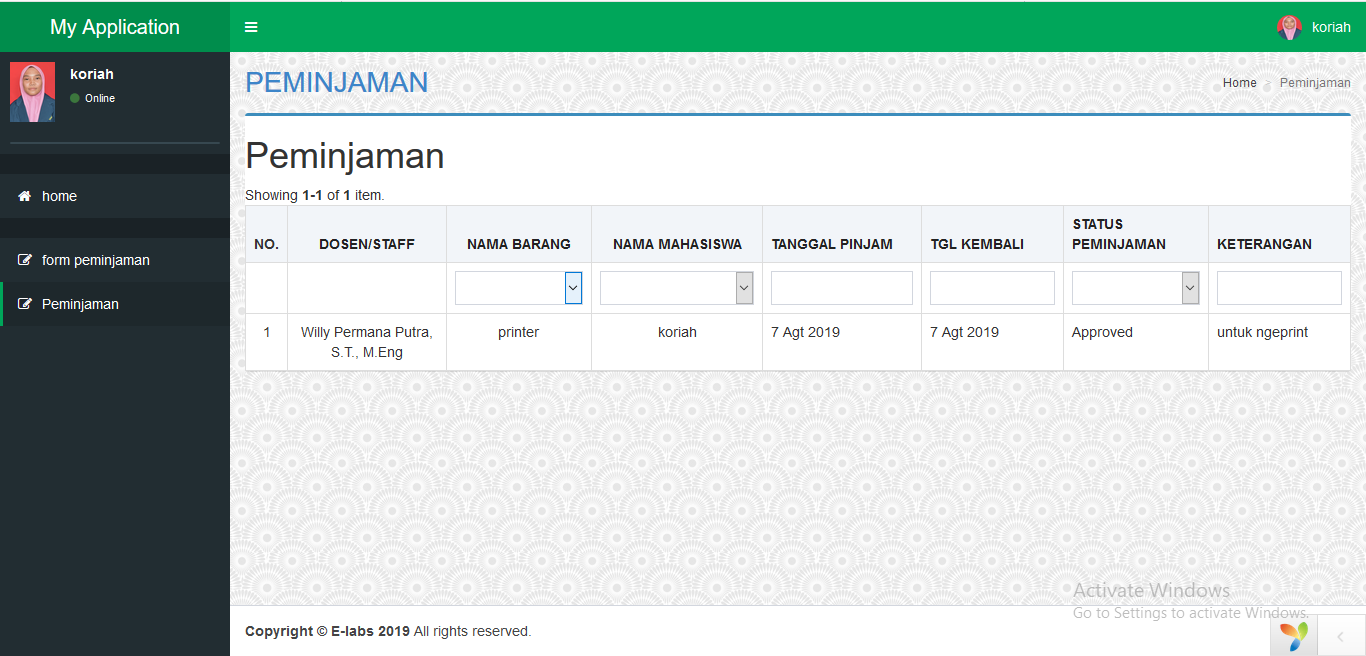
1. **Tampilan Halaman form Pinjam Barang**

Halaman form pinjam barang mahasiswa mengisi datau penanggung jawab dosen atau staf, barang yang dipilih, tanggal kapan meminjam dan mengembalikan, keterangan untuk peminjaman keperluan. Status menunggu verifikasi otomatis muncul pada sistem apabila mahasiswa sudah mengajukan permohonan peminjaman.

Gambar 4. 20 *Form* Peminjaman Barang Mahasiswa

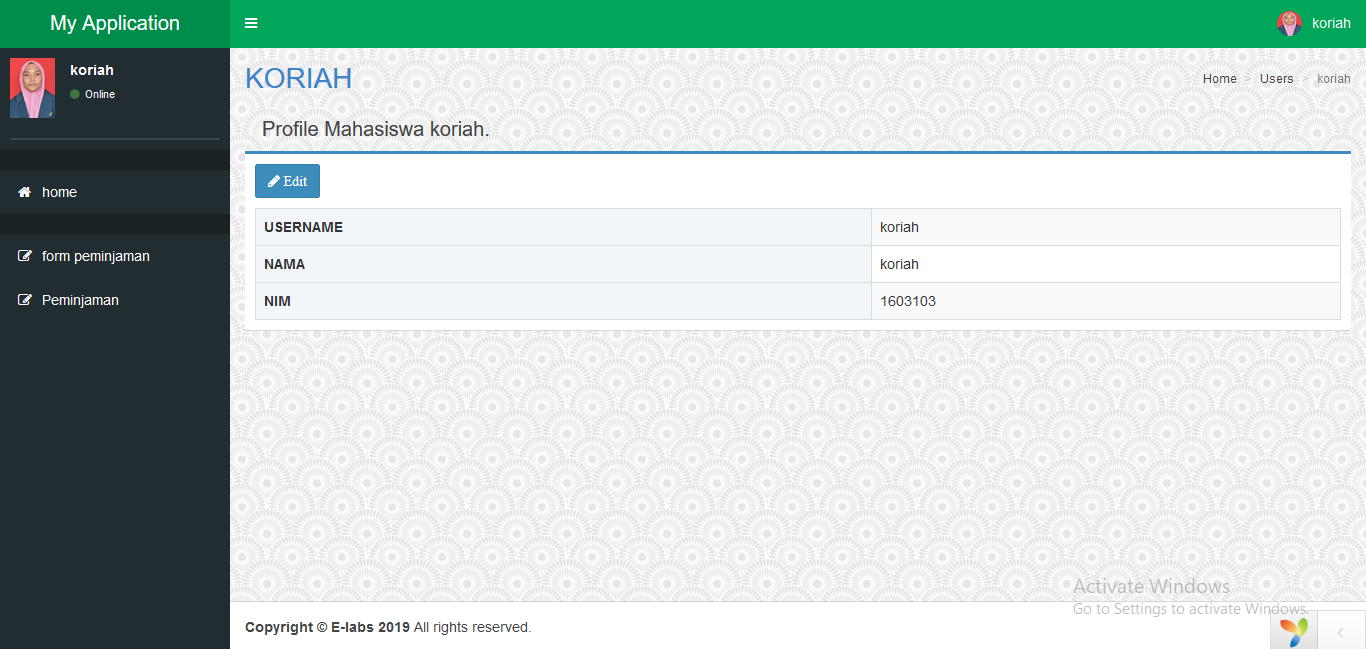
1. ***History* Peminjaman**

*History* peminjaman adalah data peminjaman yang telah dilakukan pada masing-masing mahasiswa namun mahasiswa hanya bisa melihat data peminjamannya saja tidak mempunyai akses untuk mengubah atau menghapus data pinjam yang sudah dilakukan.



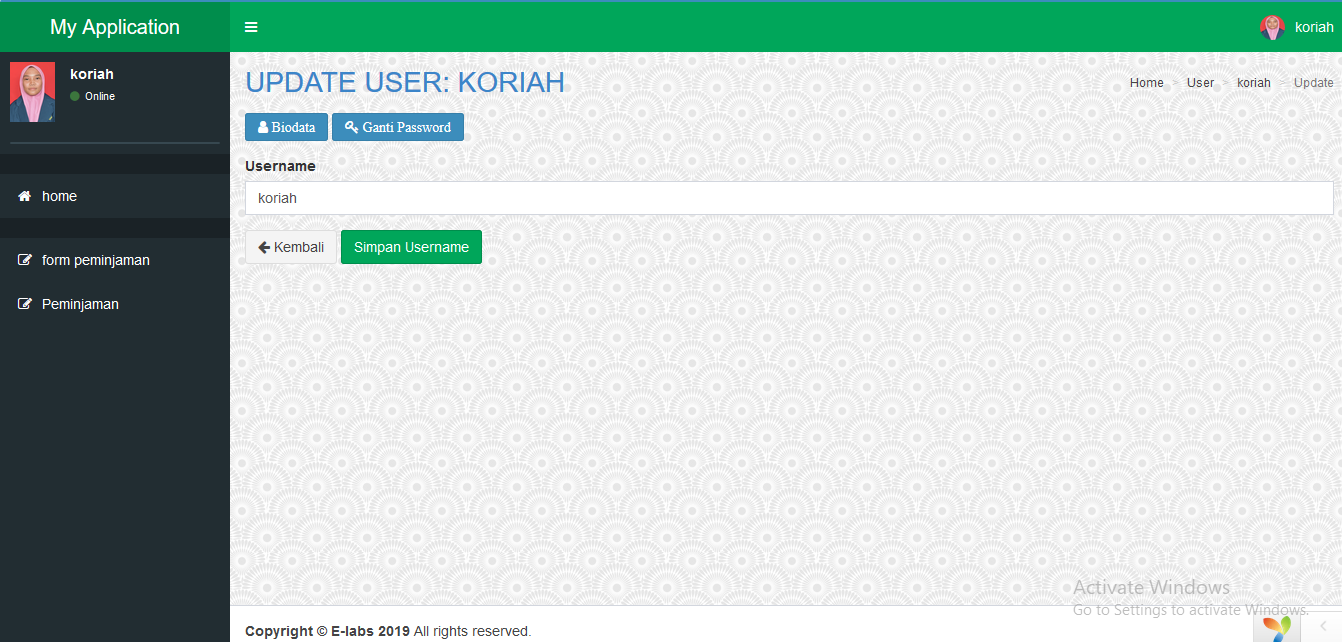
1. **Profil Mahasiswa**

Profil mahasiswa yaitu profil mahasiwa yang terdaftar pada sistem. Profil mahasiswa menampilkan username, nama mahasiswa, dan nim mahasiswa yang dapat di ubah pada halaman *edit* profil mahasiswa.



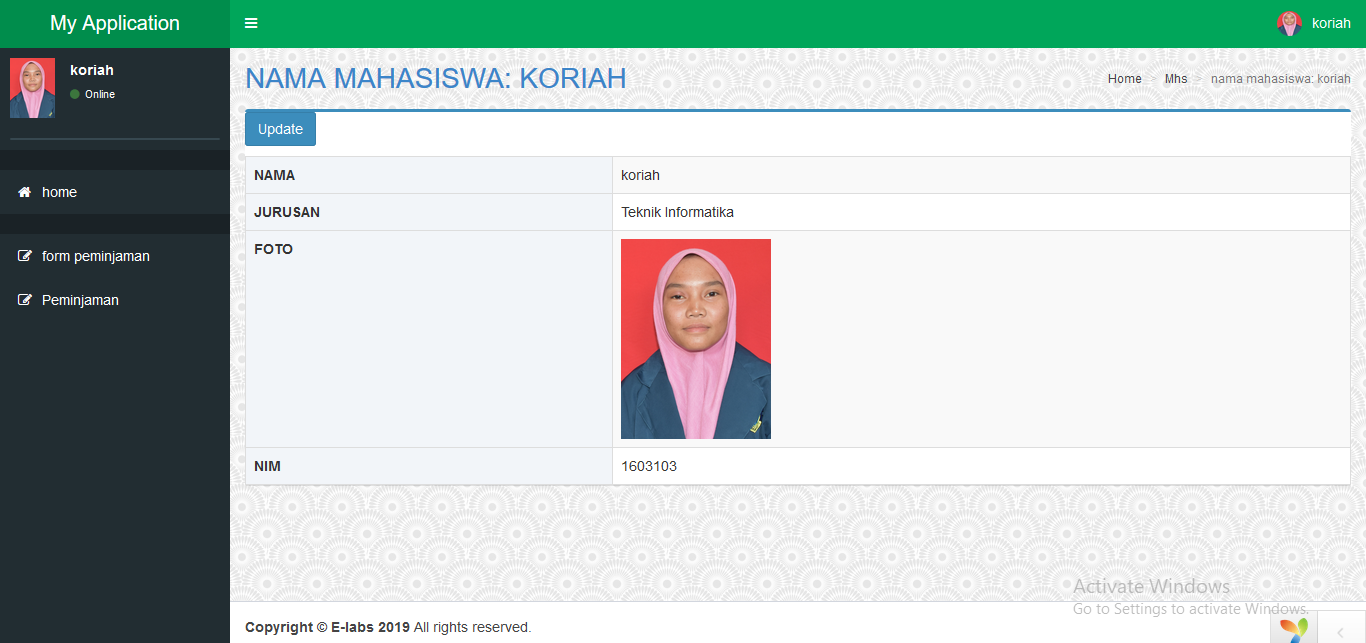
1. **Ubah *username* mahasiswa**

*Username* digunakan untuk masuk ke sistem dengan cara *login*, *username* mahasiswa dapat diubah pada kolom rubah *username* yang terdapat pada gambar dibawah ini.

Gambar 4. 21 *Update username*

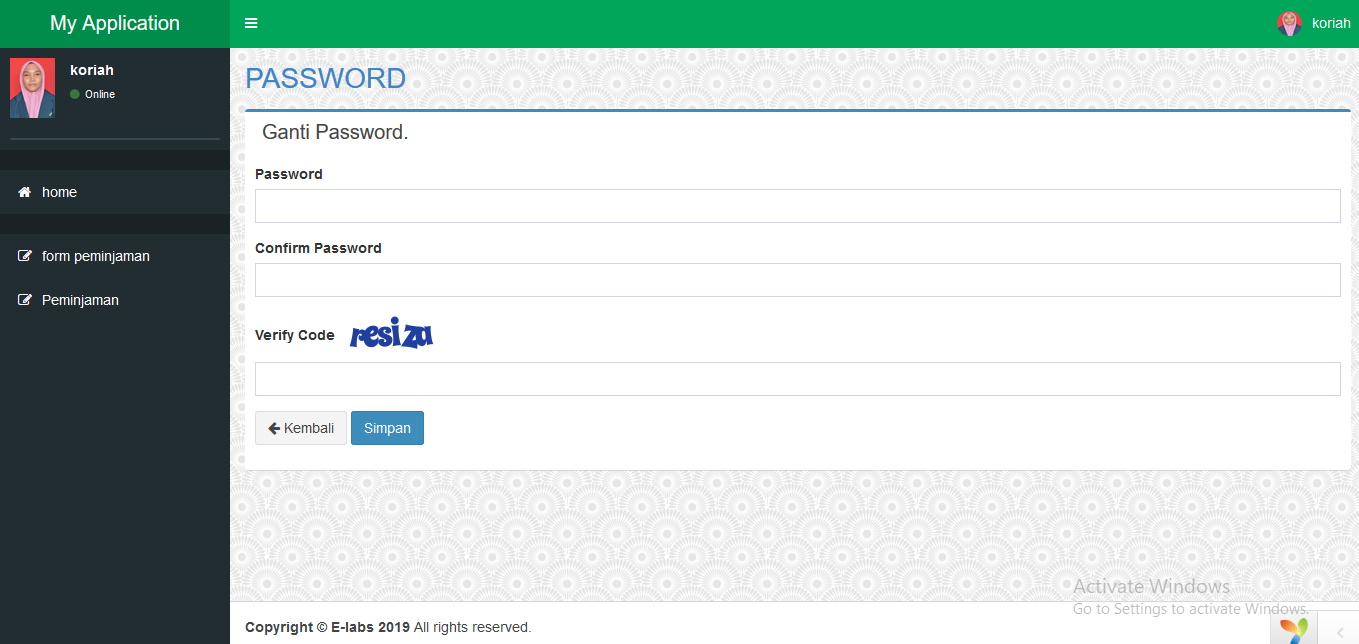
1. **Ubah Data Mahasiswa**

Ubah data mahasiswa dilakukan apabila terdapat kesalahan pada data mahasiswa, maka mahasiswa itu sendiri dapat merubah nya pada *button update.* Mahasiswa dapat merubah nama, jurusan, nim, dan foto profil.



1. **Ubah Password**

*Password* salah satu digunakan untuk masuk ke sistem dengan cara *login*, *password user* mahasiswa dapat diubah pada kolom ubah *password* yang kolom nya berisi *new password*, *confirm password* yang baru dan kode verifikasi untuk keamanan merubah passwordyang terdapat pada gambar dibawah ini.



1. **Implementasi *Database***
2. **Tabel Inventaris Barang**

Tabel Inventaris Barang berfungsi untuk menyimpan data barang dan mengambil data kategori barang yang direlasikan pada tabel inventaris barang. Berikut ini query untuk membuat tabel inventaris barang dan hasil dari pembuatan tabel inventaris\_brg terdapat pada Gambar.

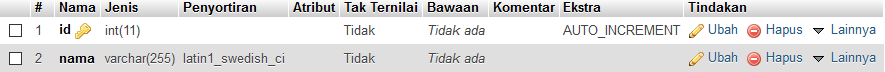
|  |
| --- |
| CREATE TABLE `inventaris\_brg` (  `id` int(11) NOT NULL,  `kode\_brg` varchar (255) NOT NULL,  `nama\_brg` varchar (255) NOT NULL,  `id\_kategori\_brg` int(11) NOT NULL,  `jumlah\_brg` varchar (255) NOT NULL,  `foto` varchar (255) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |



1. **Tabel Kategori Barang**

Tabel Kategori Barang berfungsi untuk menyimpan data kategori barang yang direlasikan pada tabel inventaris barang. Berikut ini query untuk membuat tabel kategori barang dan hasil dari pembuatan tabel kategori\_brg terdapat pada Gambar.

|  |
| --- |
| CREATE TABLE `kategori\_brg` (  `id` int(11) NOT NULL,  `nama` varchar (255) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |



1. **Tabel Pemakaian Laboratorium**

Tabel Pemakaian Laboratorium digunakan untuk menyimpan data pemakaian komputer di laboratorium. Berikut ini query untuk membuat tabel pemakaian laboratorium dan hasil dari pembuatan tabel pemakaian\_lab terdapat pada Gambar.

|  |
| --- |
| CREATE TABLE `pemakaian\_lab` (  `id` int(11) NOT NULL,  `kode\_lab` varchar (4) NOT NULL,  `nama\_lab` varchar (255) NOT NULL,  `mata\_kuliah` varchar (255) NOT NULL,  `date` DATE  ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |



1. **Tabel Dosen Staf**

Tabel Dosen Staf digunakan untuk menyimpan data dosen dan staf. Tabel dosen staf digunakan untuk membuat relasi pada tabel peminjaman. Berikut ini query untuk membuat tabel dosen staf dan hasil dari pembuatan tabel dosen\_staf terdapat pada Gambar.

|  |
| --- |
| CREATE TABLE `dosen\_staf` (  `id` int(11) NOT NULL,  `nidn` varchar (100) NOT NULL,  `nip` varchar (100) NOT NULL,  `nik` varchar (100) NOT NULL,  `nama` varchar (255) NOT NULL,  `jabatan` varchar (255) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |



1. **Tabel Mahasiswa**

Tabel Mahasiswa digunakan untuk menyimpan data mahasiswa. Tabel mahasiswa menjadi data relasi pada tabel user dan peminjaman. Berikut ini query untuk membuat tabel mahasiswa dan hasil dari pembuatan tabel mhs terdapat pada Gambar.

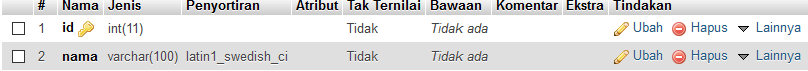
|  |
| --- |
| CREATE TABLE `mhs` (  `id` int(11) NOT NULL,  `nama` varchar (255) NOT NULL,  `id\_jurusan` int (11) NOT NULL,  `foto` varchar (255) NOT NULL,  `nim` varchar (20) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |



1. **Tabel Jurusan**

Tabel Kategori Barang berfungsi untuk menyimpan data jurusan yang direlasikan pada tabel mahasiswa. Berikut ini query untuk membuat tabel jurusan dan hasil dari pembuatan tabel jurusan terdapat pada Gambar.

|  |
| --- |
| CREATE TABLE `jurusan` (  `id` int(11) NOT NULL,  `nama` varchar (255) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |



1. **Tabel Peminjaman**

Tabel Peminjaman berfungsi untuk menyimpan data peminjaman yang dilakukan oleh mahasiswa yang direlasikan pada tabel mahasiswa, dosen staf, dan inventaris barang. Berikut ini query untuk membuat tabel peminjaman dan hasil dari pembuatan tabel peminjaman terdapat pada Gambar.

|  |
| --- |
| CREATE TABLE `peminjaman` (  `id` int(11) NOT NULL,  `id\_dosen\_staf` int(11) NOT NULL,  `id\_inventaris\_brg` int(11) NOT NULL,  `id\_mhs` int(11) NOT NULL,  `tgl\_pinjam` DATE,  `tgl\_kembali` DATE,  `keterangan` varchar (255) NOT NULL,  `status` int(11) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |



1. **Tabel *User***

Tabel *User* digunakan untuk menyimpan data *user* dan membuat hak akses untuk pengguna, pada tabel *user* terdapat relasi dari tabel mahasiswa dan tabel *user role*. Dari tabel mahasiswa sendiri untuk mengetahui data mahasiswa yang masuk sebagai user dan *user role* digunakan untuk mengetahui sebagai user admin atau mahasiswa. Berikut ini query untuk membuat tabel *user* dan hasil dari pembuatan tabel user terdapat pada Gambar.

|  |
| --- |
| CREATE TABLE `user` (  `id` int(11) NOT NULL,  `username` varchar(255) NOT NULL,  `password` varchar(255) NOT NULL,  `id\_mhs` int(11) NOT NULL,  `id\_user\_role` int(11) NOT NULL,  `token` varchar(255) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |



1. **Tabel *User Role***

Tabel *User Role* digunakan untuk menyimpan data *user role* dan mendeklarasi suatu level yang direlasikan tabel user. Berikut ini query untuk membuat tabel *user role* dan hasil dari pembuatan tabel user\_role terdapat pada Gambar.

|  |
| --- |
| CREATE TABLE `user\_role` (  `id` int(11) NOT NULL,  `nama` varchar(255) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |



1. **Pengujian Aplikasi**

Pengujian yang dilakukan untuk menguji Rancang Bangun Aplikasi Pengelolaan dan Perawatan Laboratorium Politeknik Negeri Indramayu ini memakai *black-box testing*. Berikut tabel pengujian *black-box testing* dapat dilihat pada tabel.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Deskripsi** | **Prosedur Pengujian** | **Keluaran yang diharapkan** | **Hasil yang didapat** | **Kesimpulan** |
| **Menu Admin** | | | | | |
| 1. | Admin login | * Masukkan url * Masukkan username dan password | Admin dapat melakukan login | Login berhasil | Berhasil |
| 2. | Admin ingin menambahmahasiswa | * pilih menu Data Mahasiswa * Pilih tambah mahasiswa * Isi data diri mahasiswa * Klik simpan | Admin dapat melakukan penambahan mahasiswa | Mahasiswa bertambah | Berhasil |
| 3 | Admin mengedit status peminjaman | * Pilih peminjaman * Cari peminjaman terbaru yang belum di cek. * Ubah status * Klik simpan | Admin dapat mengubah status peminjaman | Status peminjaman dapat berubah | Berhasil |
| 4 | Admin ingin mengunduh file laporan pemakaian komputer laboratorium | * Pilih pemakaian laboratorium * Pilih salah satu sub pemakaian laboratorium * Pilih button export Pdf | Admin dapat mengunduh file pdf pemakaian komputer laboratorium | File pdf terunduh | Berhasil |
| 5 | Admin ingin mengunduh file laporan pemakaian komputer laboratorium | * Pilih pemakaian laboratorium * Pilih salah satu sub pemakaian laboratorium * Pilih button export Excel | Admin dapat mengunduh file excel pemakaian komputer laboratorium | File excel  terunduh | Berhasil |
| **No** | **Deskripsi** | **Prosedur Pengujian** | **Keluaran yang diharapkan** | **Hasil yang didapat** | **Kesimpulan** |
| 6 | Admin ingin melihat hasil perhitungan UKT | * Pilih menu perhitungan * Pilih sub menu lihat UKT | Admin dapat melihat hasil perhitungan UKT | Hasil perhitungan UKT dapat dilihat admin | Berhasil |
| 7 | Admin ingin mencetak laporan UKT keseluruhan | * Pilih menu perhitungan * Pilih sub menu lihat UKT * Klik tombol Print PDF | Admin dapat mencetak laporan UKT keseluruhan | Laporan UKT keseluruhan tercetak | Berhasil |
| 8 | Admin ingin mencetak laporan UKT per periode | * Pilih menu perhitungan * Pilih sub menu Lihat UKT * Searh by periode * Klik tombol Print PDF | Admin dapat mencetak laporan UKT per periode | Laporan UKT per periode tercetak | Berhasil |
| 9 | Admin ingin mengganti akun admin dan Mahasiswa | * Pilih menu user * Pilih akun user * Tekan icon edit * Ganti isi | Admin dapat mengganti akun Admin dan Mahasiswa | Akun admin dan mahasiswa teredit | Berhasil |
| 10 | Admin ingin menghapus akun Mahasiswa | * Pilih menu user * Pilih akun user * Tekan icon hapus | Admin dapat Menghapus akun Mahasiswa | Akun mahasiswa berhasil dihapus | Berhasil |
| **Menu Mahasiswa** | | | | | |
| 11 | Mahasiswa ingin melakukan login | * Masukkan url * Masukkan username dan password | Mahasiswa dapat melakukan login | Login berhasil | Berhasil |
| 12 | Mahasiswa ingin mendownload formulir UKT | * Lakukan Login * Masuk menu Dashboard * Tekan tulisan DISINI yang ada pada halaman dashboard | Mahasiswa dapat melakukan proses download form UKT | Formulir terdownload | Berhasil |
| **No** | **Deskripsi** | **Prosedur Pengujian** | **Keluaran yang diharapkan** | **Hasil yang didapat** | **Kesimpulan** |
| 13 | Mahasiswa ingin mengupload formulir UKT yang sudah diisi dan persyaratan UKT yang sudah di scan | * Pilih menu Upload berkas * Isi data * Klik tombol simpan | Mahasiswa dapat melakukan Upload Form UKT dan persyaratan UKT | Form UKT dan persyaratan terupload | Berhasil |