

## Pembuatan *Website Administrator* Peminjaman Ruangan Laboratorium Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta

Jethro Reyhan Naldo, Pascal Aldwin Hernando, Ing. Artambo Benjamin Pangaribuan, BSc (Hons)  
Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran  
Jakarta

Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia  
jethron@gmail.com, pascal@upnvj.ac.id, artambo@upnvj.ac.id

**Abstrak.** Aplikasi *Website Administrator* Peminjaman Ruangan Laboratorium Fakultas Ilmu Komputer merupakan sebuah aplikasi berbasis web yang dibuat dengan tujuan untuk mempermudah dan membantu pihak administrasi dalam proses pengecekan peminjaman ruangan laboratorium yang terhubung pada aplikasi *android* “Room Meet” pada Fakultas Ilmu Komputer. Aplikasi ini digunakan untuk menerima atau menolak permintaan peminjaman ruangan laboratorium; mendaftarkan jadwal kelas pada satu semester; penerimaan pengguna untuk menggunakan aplikasi *client* “Room Meet”; dan melihat riwayat peminjaman yang telah terlaksana. Aplikasi ini dibangun menggunakan *framework Vue.JS* untuk bagian *front-end* nya, dan menggunakan *Firebase* sebagai *database* nya. Didapatkan hasil berupa aplikasi berbasis *website* dengan fitur login, penerimaan atau penolakan peminjaman, pendaftaran pengguna, pendaftaran kelas, dan penyimpanan riwayat peminjaman ruangan yang terintegrasi menggunakan *Firebase*

**Kata kunci:** *Website, Vue.JS, Firebase.*

### 1 Pendahuluan

Pada era ini, perkembangan teknologi yang pesat telah membantu pengguna-pengguna umum dalam mempermudah dan mengefisienkan suatu proses pekerjaan. Kemajuan teknologi yang pesat ini juga mengakibatkan penyerapan penggunaan teknologi yang sangat tinggi di masyarakat umum. Penggunaan *smartphone*, terutama *android*, sangat banyak digunakan di kalangan umum. Hal ini dapat dimanfaatkan untuk membantu dalam menyelesaikan suatu proses, contohnya seperti proses peminjaman ruangan melalui aplikasi *smartphone*. Untuk menunjang proses seleksi dari peminjaman tersebut, pihak administrasi membutuhkan pula aplikasi yang dapat menggabungkan menjadi satu seluruh proses penerimaan atau penolakan permintaan peminjaman ruangan ini. Aplikasi ini harus dapat mudah digunakan pada berbagai perangkat agar proses administrasi dapat dilaksanakan dimana saja dengan mudah, oleh karena itu digunakanlah tipe aplikasi berupa situs web untuk meningkatkan ketersediaan dalam penggunaan aplikasi ini.

Dalam proses pembuatan aplikasi ini, digunakan proses *modelling Waterfall*. Proses *Waterfall* merupakan sekumpulan tahapan pekerjaan yang dilakukan perancang dan *programmer* suatu aplikasi dimana proses pengerjaannya dilakukan secara bertahap, dimulai dari perencanaan konsep, pemodelan, implementasi, pengujian, dan pemeliharaan. Setiap proses ini harus selesai secara maksimal karena proses *Waterfall* tidak akan mengulang proses yang telah dilakukan sebelumnya. (Rizky, 2019).

Aplikasi *administrator* berbasis *website* ini memiliki fungsi untuk pekerja laboratorium Fakultas Ilmu Komputer agar dapat dengan mudah melihat siapa yang ingin meminjam ruangan, mengesahkan peminjaman, mengesahkan *client* yang akan meminjam ruangan, menambahkan jadwal perkuliahan pada ruangan laboratorium, dan juga melihat dan mengunduh *log* peminjaman ruangan yang telah disetujui.

## 2 Landasan Teori

### 2.1 Website

*Website* merupakan kumpulan halaman web yang digunakan untuk menampilkan informasi berupa teks, gambar dan sebagainya. Dan merupakan sebuah domain yang terdapat di internet. *Website* biasanya diakses melalui *HTTP* yang merupakan *protocol* yang digunakan untuk menyampaikan informasi dari *server website*. Dimana *website* memiliki dua jenis yaitu statis dan dinamis. (Domantas., 2019)

### 2.2 HTML

*HTML* atau biasa dikenal dengan *Hyper Text Markup Language* merupakan kode program yang digunakan untuk menyusun dan menampilkan halaman *website*. Penyusunan menggunakan *HTML* melibatkan *tag-tag* untuk setiap format dalam *HTML*. Dimana hampir sebagian besar teknologi website menggunakan *HTML* dalam pembuatan *website*. (Domantas., 2019)

### 2.3 CSS

*CSS* atau *Cascading Style Sheet* digunakan untuk peningkatan tampilan pada website. Dimana *CSS* biasa digunakan bersama dengan *HTML*. Dalam penulisannya *CSS* disimpan dalam *file* berekstensi *.css* dan disimpan terpisah dengan *file HTML*. Sehingga *CSS* berguna dalam melakukan perubahan tampilan agar menjadi lebih menarik lagi. (Mozilla, 2021)

### 2.4 Javascript

*Javascript* merupakan kode program yang digunakan dalam meningkatkan kedinamisan pada halaman *website* dan membuat tampilan lebih menarik lagi. (Mozilla, 2021)

### 2.5 Vue JS

*Vue JS* merupakan *framework* progresif yang digunakan untuk membangun antarmuka pengguna (UI). *Vue* didesain agar dapat digunakan secara bertahap. Inti dari *library Vue* terfokuskan pada lapisan *view*, dan juga mudah untuk digunakan dan diintegrasikan dengan *library* lain atau projek yang sudah ada. (Vue.js, n.d.)

### 2.6 Firebase

*Firebase* adalah suatu layanan dari google untuk mempermudah para pengembang-pengembang aplikasi untuk mengembangkan aplikasinya. *Firebase* (BaaS 'Backend as a Service') ini merupakan solusi yang ditawarkan oleh *Google* untuk mempermudah pekerjaan *Developer*. Dengan adanya *Firebase*, pengembang aplikasi bisa fokus mengembangkan aplikasi tanpa harus memberikan *effort* yang besar untuk urusan *backend*. (Stevenson, 2018)

### 2.7 Software Development Life Cycle Waterfall

Proses *Waterfall* merupakan sekumpulan tahapan pekerjaan yang dilakukan perancang dan *programmer* suatu aplikasi dimana proses pengerjaannya dilakukan secara bertahap, dimulai dari perencanaan konsep, pemodelan, implementasi, pengujian, dan pemeliharaan. Setiap proses ini harus selesai secara maksimal karena proses *Waterfall* tidak akan mengulang proses yang telah dilakukan sebelumnya. (Rizky, 2019).

Teknik ini terdiri dari berbagai bagian, yakni:

#### 2.7.1. Requirement Analysis

Bagian ini membahas ketentuan, kebutuhan, dan fitur yang dibutuhkan dalam aplikasi pada hasil akhir produknya setelah selesai. Informasi ini didapatkan menggunakan proses wawancara, survei, atau diskusi.

- 2.7.2. Desain Sistem**  
 Dalam proses desain sistem, pihak pengembang aplikasi menerjemahkan kebutuhan yang telah didapatkan pada proses *requirement analysis* dengan membahas mengenai struktur data, arsitektur perangkat lunak, tampilan aplikasi, dan algoritma yang digunakan dalam aplikasi.
- 2.7.3. Implementasi**  
 Tahap implementasi ini berfungsi untuk menerjemahkan desain yang telah dilaksanakan pada proses desain sistem, lalu diterjemahkan menjadi kode bahasa pemrograman. Pada tahap ini pembuat aplikasi membuat modul-modul kecil yang nantinya akan diintegrasikan pada proses selanjutnya.
- 2.7.4. Integrasi dan Pengetesan**  
 Proses integrasi dan pengetesan berfungsi untuk menggabungkan modul dan asset yang telah dibuat pada proses implementasi, lalu digabungkan menjadi aplikasi yang telah dirancang. Setelah digabungkan, dibutuhkan juga pengetesan hasil dari penggabungan tersebut untuk mengetahui akan kesalahan-kesalahan yang dihasilkan.
- 2.7.5. Operasi dan Pemeliharaan**  
 Aplikasi yang telah selesai dikembangkan dilakukan proses pemeliharaan oleh pengembangnya. Proses ini berfungsi untuk memperbaiki kesalahan yang tidak ditemukan atau disadari pada proses sebelumnya. Perbaikan dari kesalahan tersebut dikirimkan dengan cara peningkatan versi aplikasi.

### 3 Hasil dan Pembahasan

#### 3.1. Perancangan *Front End Website Administrator* “Room Meet”

Perancangan *front-end website Administrator* “Room Meet” dibuat dengan menggunakan HTML, serta menggunakan CSS pada websitenya. Dengan menggabungkan *HTML* dan *CSS*, akan menghasilkan *interface* yang menarik dengan penempatan posisi yang baik. Struktur utama dari *interface* dari *website Administrator* “Room Meet” meliputi halaman *login*, halaman *dashboard*, halaman kelas, halaman *user profile*, halaman, halaman verifikasi user, halaman *history*, dan halaman *about*.

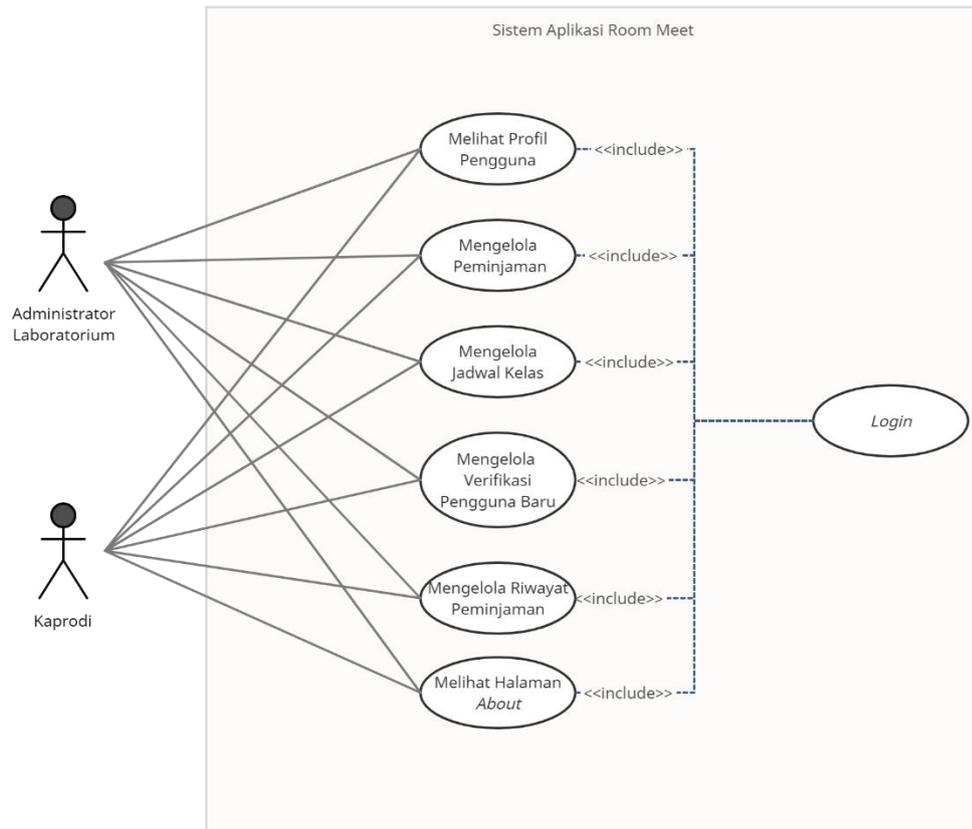
**Tabel. 1.** Tabel Penjelasan Rancangan Aplikasi Peminjaman Ruang Laboratorium FIK

| No | Bagian                           | Keterangan   |
|----|----------------------------------|--|
| 1  | Halaman <i>login</i>             | Halaman pertama yang muncul dari website. Terdapat logo LAB FIK serta tombol login menggunakan akun <i>google</i>  |
| 2  | Halaman <i>dashboard</i>         | Merupakan halaman utama setelah melakukan <i>login</i> , dimana halaman ini memuat data-data calon peminjaman ruangan lab yang akan di konfirmasi/tolak oleh admin.  |
| 3  | Halaman Kelas                    | Halaman yang berisikan jadwal kelas regular yang dijadwalkan pada semester sekarang serta melakukan <i>input</i> kelas regular.  |
| 4  | Halaman <i>User Profile</i>      | Menunjukkan profil user admin yang sedang <i>login</i> , berupa tampilan nama dan email <i>Administrator</i> .   |
| 5  | Halaman <i>User Verification</i> | Halaman yang berisikan calon <i>client user</i> yang mengajukan konfirmasi, menunjukkan kelengkapan data yang diperlukan untuk dikonfirmasi oleh admin agar <i>client user</i> dapat melakukan peminjaman. Apabila ditolak, maka data yang terinput tersebut akan di <i>clean</i> dari sistem. |

- |   |                        |   |
|---|------------------------|---|
| 6 | Halaman <i>History</i> | Halaman yang berisikan riwayat peminjaman lab yang diterima serta melakukan <i>export</i> kedalam bentuk pdf. |
| 7 | Halaman <i>About</i>   | Halaman pengenalan website secara umum.   |

### 3.2. Pemodelan Fitur

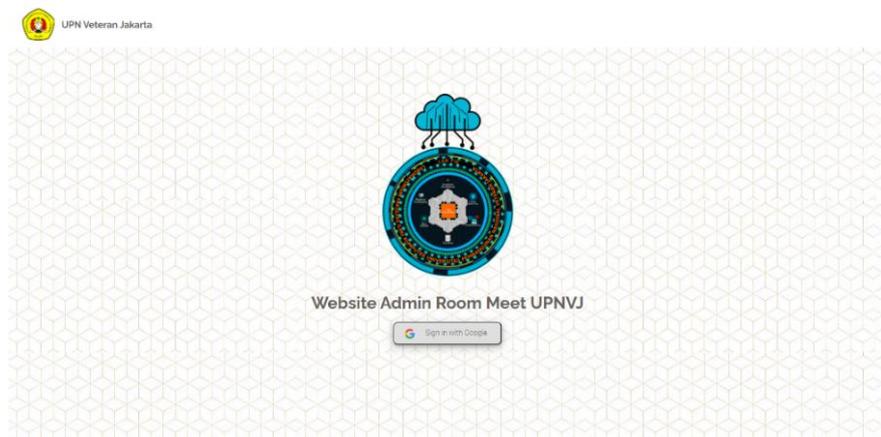
Pada pemodelan fitur ini berfungsi untuk merancang model yang akan digunakan pada aplikasi *website* administrasi “Room Meet” berdasarkan kebutuhan dan hak akses dari pengguna. Perancangan model ini akan digambarkan melalui model *Use Case Diagram*.



**Gambar. 1.** Use Case Diagram

### 3.3. Tampilan Hasil Dari Pembuatan Aplikasi

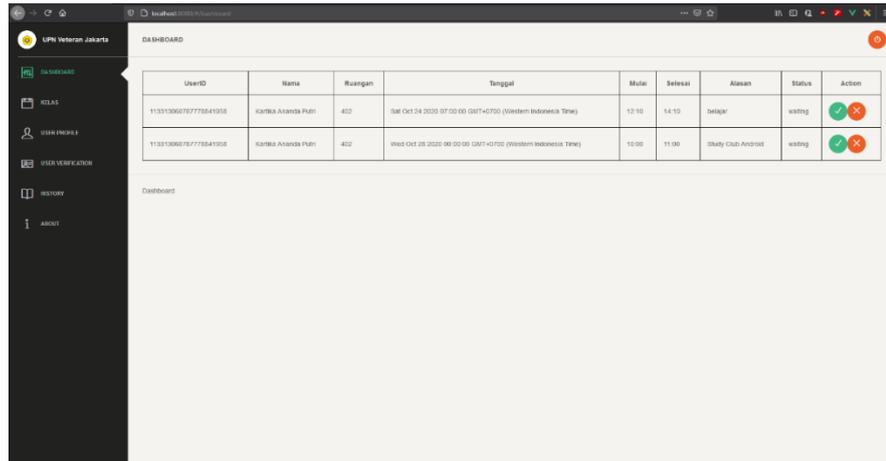
#### 3.3.1. Halaman *Login*



**Gambar. 2.** Tampilan Halaman *Login*

Gambar 2 merupakan halaman pertama yang akan ditemui pada saat pertama kali membuka aplikasi ini. Halaman menunjukkan tampilan dari halaman *login*. Pada halaman ini terdapat logo LAB FIK serta tombol login menggunakan akun *google*. Semua akun dapat masuk ke *website* ini, dengan perbedaan otorisasi. Akun yang tidak terotorisasi hanya dapat melihat halaman "*User Profile*", dimana didalamnya berisikan nama lengkap dan nama *email* yang digunakan untuk *login*.

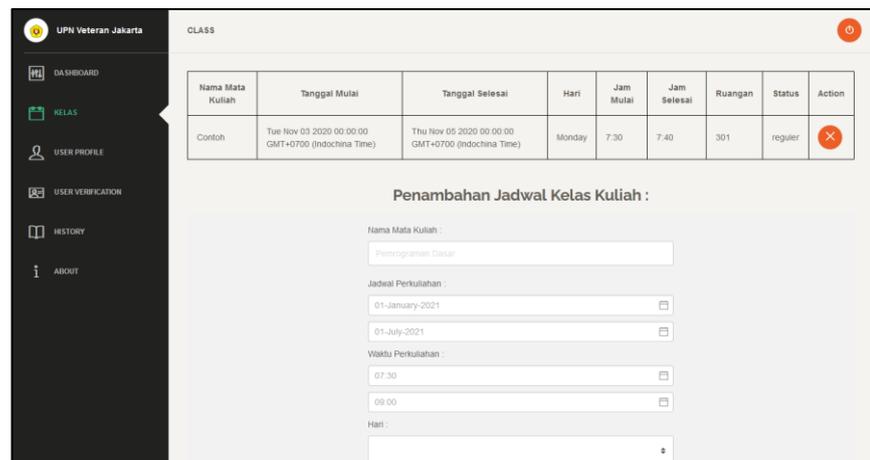
### 3.3.2. Halaman *Dashboard*



**Gambar. 3.** Tampilan Halaman *Dashboard*

Gambar 3 menunjukkan tampilan halaman *dashboard* yang dapat diakses setelah mendapatkan otorisasi untuk masuk kedalam aplikasi. Pada halaman *dashboard* memuat *data-data* calon peminjaman ruangan lab yang didapatkan dari input aplikasi *Android* “*Room Meet*”. Didalam halaman ini *Administrator* dapat menerima atau menolak permintaan peminjaman ruangan.

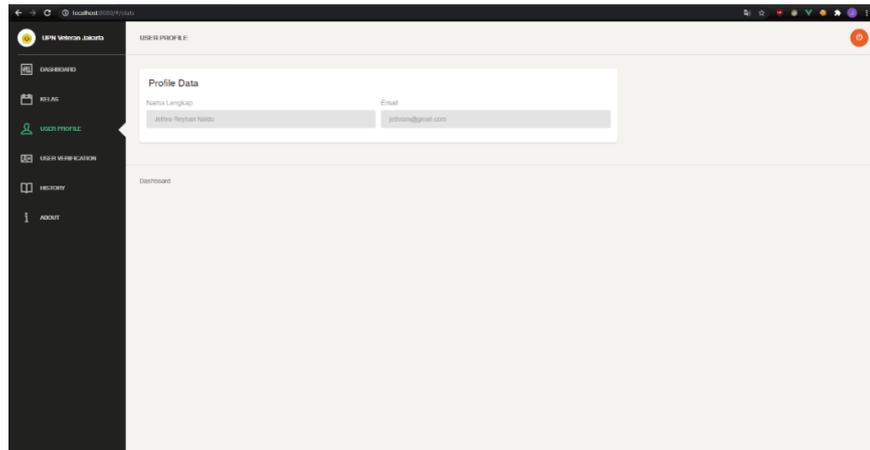
### 3.3.3. Halaman *Kelas*



**Gambar. 4.** Tampilan Halaman *Kelas*

Gambar 4 merupakan halaman kelas yang berfungsi untuk memperlihatkan kelas yang sudah terdaftar pada *database*, menambahkan jadwal kelas pada kegiatan belajar mengajar (KBM) dalam satu semester, dan menghapus jadwal kelas jika ada kesalahan. Jadwal kelas ini akan menjadi eksepsi pada aplikasi nantinya, dimana pengguna tidak dapat meminjam ruangan yang waktunya berbenturan dengan jadwal kelas yang telah terdaftar.

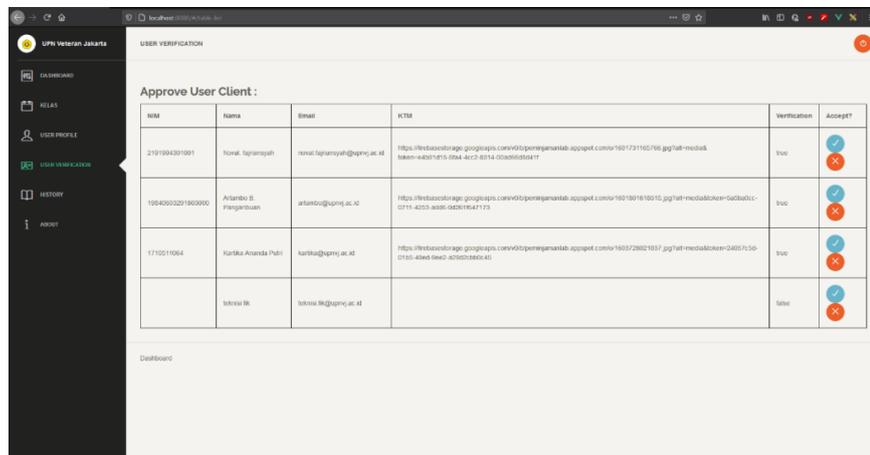
### 3.3.4. Halaman *User Profile*



**Gambar. 5.** Tampilan Halaman *User Profile*

Gambar 5 menunjukkan tampilan pada halaman *User Profile* yang berfungsi untuk menunjukkan profil dari user yang sedang *login*. Didalam halaman ini terdapat bagian nama lengkap yang menunjukkan nama lengkap dari pengguna *website* sesuai yang terdaftar pada *database*, dan *email* yang sedang digunakan untuk masuk kedalam *website* ini. Jikalau pengguna menggunakan email yang belum terotorisasi oleh *Administrator*, maka pengguna tidak terotorisasi tersebut hanya dapat melihat halaman ini saat masuk kedalam *website*.

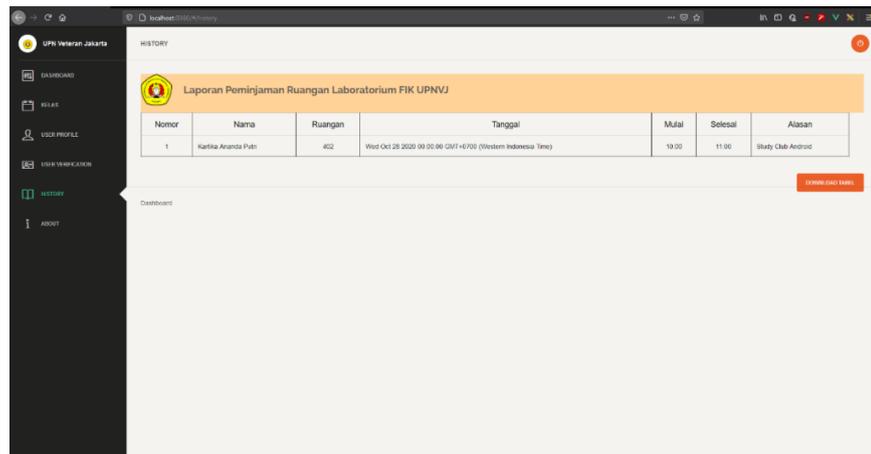
### 3.3.5. Halaman *User Verification*



**Gambar. 6.** Tampilan Halaman *User Verification*

Gambar 6 menunjukkan tampilan dari halaman *User Verification* yang berisikan calon *client user* yang mengajukan pendaftaran untuk memiliki akun pada aplikasi *android "Room Meet"*, melihat akun mana saja yang telah terverifikasi, dan untuk memberikan verifikasi atau menolak permintaan pendaftaran pengguna tersebut. Pada tabelnya berisikan nomor induk mahasiswa (NIM), nama lengkap pendaftar, email yang digunakan untuk mendaftar, dan juga link untuk melihat kartu tanda mahasiswa (KTM) untuk melakukan verifikasi keaslian pendaftar.

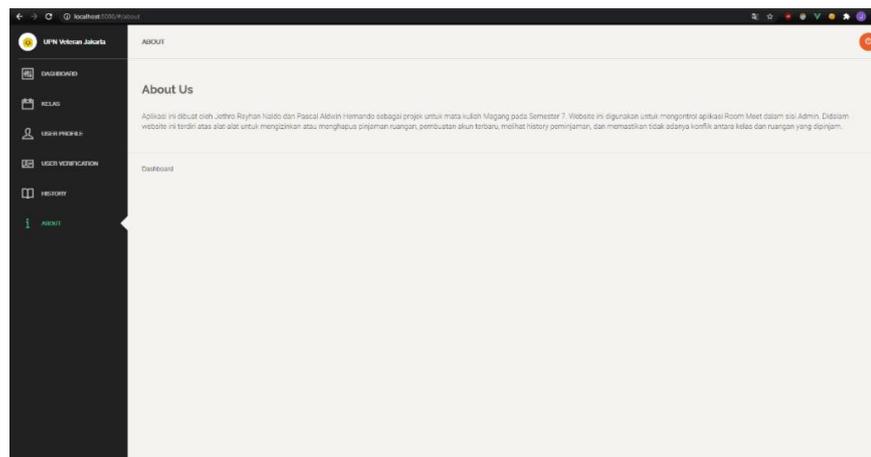
### 3.3.6. Halaman *History*



**Gambar. 7.** Tampilan Halaman *History*

Gambar 7 menunjukkan tampilan dari halaman *history* yang berfungsi untuk melihat riwayat peminjaman yang telah diterima oleh administrasi. Pada halaman ini juga dapat mengambil informasi riwayat menjadi file *Portable Document Format* (PDF).

### 3.3.7. Halaman *About*



**Gambar. 8.** Tampilan Halaman *About*

Gambar 8s menampilkan tampilan dari halaman *about* yang berisikan informasi mengenai pembuat aplikasi dan latar belakang dari pembuatan aplikasi ini.

## 4 Kesimpulan dan Saran

Pengerjaan *website Administrator* peminjaman lab “Room Meet” terbagi atas dua bagian, yakni bagian *front-end* dan *back-end*. *Front-end website* dirancang dengan menggunakan kombinasi dari HTML, dan CSS yang dilengkapi dengan tampilan-tampilan yang sederhana sehingga mudah untuk digunakan oleh orang umum. Sementara itu, untuk bagian *back-end* menggunakan bahasa pemrograman *javascript*, dengan menyatukan pemrograman logika dan layanan *database* dari *google*, yaitu *firebase*. Situs ini terdiri atas halaman *login*, halaman *dashboard*, halaman *kelas*, halaman *user profile*, halaman *user verification*, halaman *history*, dan halaman *about*.

Berdasarkan hasil pengerjaan, maka disarankan dalam tahap perancangan *website* bisa lebih memberikan tampilan *interface* yang mudah dipahami. Penggunaan warna, *logo icon* yang tepat dapat membantu dalam mempermudah *user* awam memahami sistem yang berjalan dalam *website* tersebut. Selain itu juga, pemrograman pada bagian *backend* dibuat dengan lebih efisien lagi. Dengan sistem *loop* yang banyak membuat kompleksitas waktu yang cenderung memberatkan sumber daya sistem.

## 5 Referensi

- [1] Domantas., G. (2019, November 25). *What is HTML? The Basics of Hypertext Markup Language Explained*. Diambil kembali dari hostinger.com: <https://www.hostinger.com/tutorials/what-is-html>
- [2] Mozilla. (2021, Januari 22). *Learn to style HTML using CSS*. Retrieved from MDN Web Docs: <https://developer.mozilla.org/en-US/docs/Learn/CSS>
- [3] Mozilla. (2021, Februari 20). *What is JavaScript?* Retrieved from MDN Web Docs: [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)
- [4] Stevenson, D. (2018, September 25). *What is Firebase? The complete story, abridged*. Retrieved from Medium.com: <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>
- [5] Vue.js. (n.d.). Retrieved from vuejs.org: <https://vuejs.org/v2/guide/>
- [6] Rizky, D. (2019, Januari 19). *Apa itu SDLC Waterfall?* Diambil kembali dari medium.com: <https://medium.com/dot-intern/sdlc-metode-waterfall-5ae2071f161d>