

## LAMPIRAN

### Lampiran 1 Surat Kesediaan Pembimbing

---

#### SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : R. Rekyan Maulana Setyawan  
NIM : 20090024  
Program Studi : D4 Teknik Informatika

Pihak Kedua

Nama : Sharfina Febbi Handayani, S.Kom., M.Kom.  
Status : Dosen Tetap  
NIDN : 0617029201  
Jabatan Fungsional : Asisten Ahli  
Pangkat/Golongan : Penata Muda Tk I/IIIb

Pada hari ini Jumat tanggal 15 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat bersedia bimbingan secara rutin untuk menyelesaikan Skripsi minimal 3 kali dalam sebulan, apabila dalam waktu 40 hari pertama setelah kesepakatan ini dibuat pihak pertama tidak konsisten untuk menyanggupi maka pihak kedua berhak membatalkan kesepakatan ini dan tidak melanjutkan untuk membimbing pihak pertama dengan tidak akan memberikan rekomendasi untuk mengikuti ujian skripsi. Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 15 Maret 2024

Pihak Pertama



R. Rekyan Maulana Setyawan

Pihak Kedua



Sharfina Febbi Handayani, S.Kom., M.Kom.

Mengetahui  
Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Apriliani, S.T., M.Kom.  
NIPY. 09.015.225

## SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : R. Rekyan Maulana Setyawan  
NIM : 20090024  
Program Studi : D4 Teknik Informatika

Pihak Kedua

Nama : Ardi Susanto, S.Kom., M.Cs.  
Status : Dosen Tetap  
NIDN : 0629109301  
Jabatan Fungsional : Asisten Ahli  
Pangkat/Golongan : Penata Muda Tk. I/IIIB

Pada hari ini Senin tanggal 18 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak. Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 18 Maret 2024

Pihak Pertama



R. Rekyan Maulana Setyawan

Pihak Kedua



Ardi Susanto, S.Kom., M.Cs.

Mengetahui  
Ketua Program Studi Sajana Temban Teknik Informatika



Dyah Apriliani, S.T., M.Kom.  
NIPY. 09.015.225

## Lampiran 2 Surat Pernyataan Pengajuan HKI

### SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : R. Rekyan Maulana Setyawan  
Kewarganegaraan : Indonesia  
Alamat : Desa Randusari, RT.01/RW.01, Kecamatan Pagerbarang,  
Kabupaten Tegal, Jawa Tengah 52462
2. Nama : Sharfina Febbi Handayani, S.Kom., M.Kom  
Kewarganegaraan : Indonesia  
Alamat : Jalan Brayon 1 No 46 Desa Grobog Kulon RT 02 RW 06  
Kecamatan Pangkah Kabupaten Tegal
3. Nama : Ardi Susanto, S.Kom., M.Cs.  
Kewarganegaraan : Indonesia  
Alamat : Margadana, Tegal

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:  
Berupa : Program Komputer  
Berjudul : Penerapan Model LSTM dan Cosine Similarity dalam Rekomendasi  
Freelancer untuk Meningkatkan Layanan Jasa Berbasis Website
  - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyektif kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
  - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
  - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
  - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
  - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
  - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.

4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
- a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
  - b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
  - c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, Juli 2024



R. Rekyan Maulana Setyawan  
Pemegang Hak Cipta \*

Sharfina Febbi Handayani, S.Kom., M.Kom  
Pemegang Hak Cipta \*

Ardi Susanto, S.Kom., M.Cs.  
Pemegang Hak Cipta \*

\* Semua pemegang hak cipta agar menandatangani di atas materai.

## Lampiran 3 Surat Pengalihan HKI

### SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

1. Nama : R. Rekyan Maulana Setyawan  
Kewarganegaraan : Indonesia  
Alamat : Desa Randusari, RT.01/RW.01, Kecamatan Pagerbarang, Kabupaten Tegal, Jawa Tengah 52462
2. Nama : Sharfina Febbi Handayani, S.Kom., M.Kom  
Kewarganegaraan : Indonesia  
Alamat : Jalan Brayon 1 No 46 Desa Grobog Kulon RT 02 RW 06 Kecamatan Pangkah Kabupaten Tegal
3. Nama : Ardi Susanto, S.Kom., M.Cs.  
Kewarganegaraan : Indonesia  
Alamat : Margadana, Tegal

Adalah Pihak I selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

- Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M)  
Politeknik Harapan Bersama  
Alamat : Jl. Mataran No.9 Pasurungan Lor Kota Tegal

Adalah Pihak II selaku Pemegang Hak Cipta berupa Program Komputer dengan judul Penerapan Model LSTM dan Cosine Similarity dalam Rekomendasi Freelancer untuk Meningkatkan Layanan Jasa Berbasis Website. untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Tegal, Juli 2024

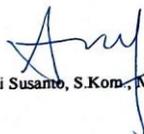


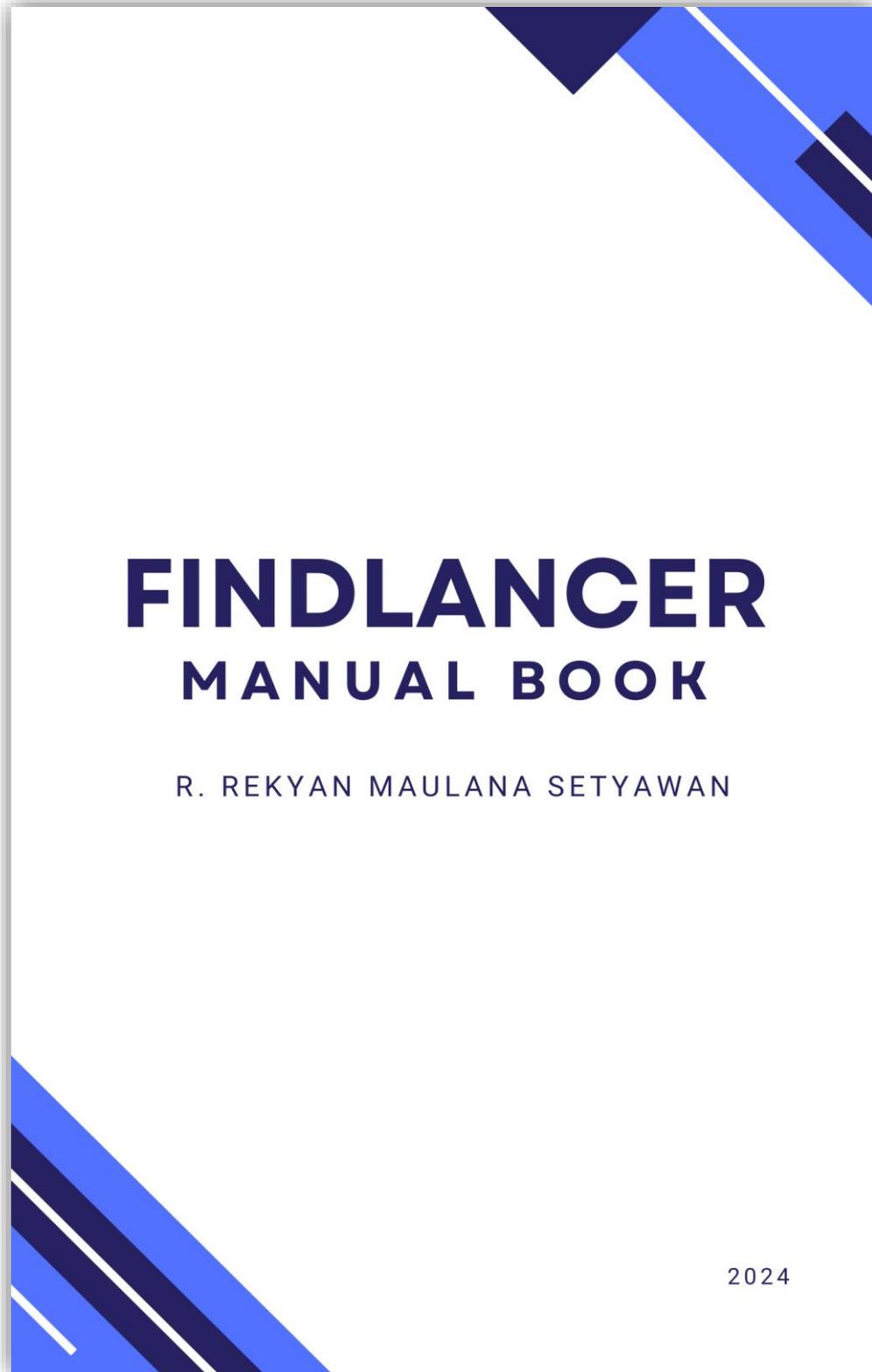
(Dr. Aldi Budi Riyanta, S.Si, M.T.)



(R. Rekyan Maulana Setyawan)

  
(Sharfina Febbi Handayani, S.Kom., M.Kom)

  
(Ardi Susanto, S.Kom., M.Cs.)



## 1. Pendahuluan

FindLancer adalah *website* yang membantu pengguna menemukan jasa dan *freelancer* yang sesuai dengan kebutuhan mereka. Menggunakan teknologi canggih, FindLancer menganalisis deskripsi pekerjaan yang dikirimkan oleh pengguna dan menggabungkannya dengan profil *freelance* yang ada di sistem. Sistem ini menggunakan *machine learning* untuk memberikan rekomendasi yang akurat dan relevan. FindLancer bertujuan untuk mempermudah dan mempercepat pencarian *freelancer* sehingga pengguna dapat dengan mudah menemukan profesional yang tepat untuk proyek mereka.

## 2. Role User

Pada *website* FindLancer terdapat 3 *role user*, yaitu pengguna, *freelancer* dan admin. Masing-masing *role* ini mempunyai wewenang yang berbeda. Berikut merupakan perbedaan *user* dengan *role* pengguna, *freelancer* dan admin:

### 1) Admin

- Dapat mengelola profil pribadi
- Dapat melihat jumlah *users* dan jumlah *freelancers*
- Dapat mengelola data *users* seperti menghapus data *user*
- Dapat mengelola data *freelancers* seperti menghapus data *freelancer*.

### 2) Pengguna

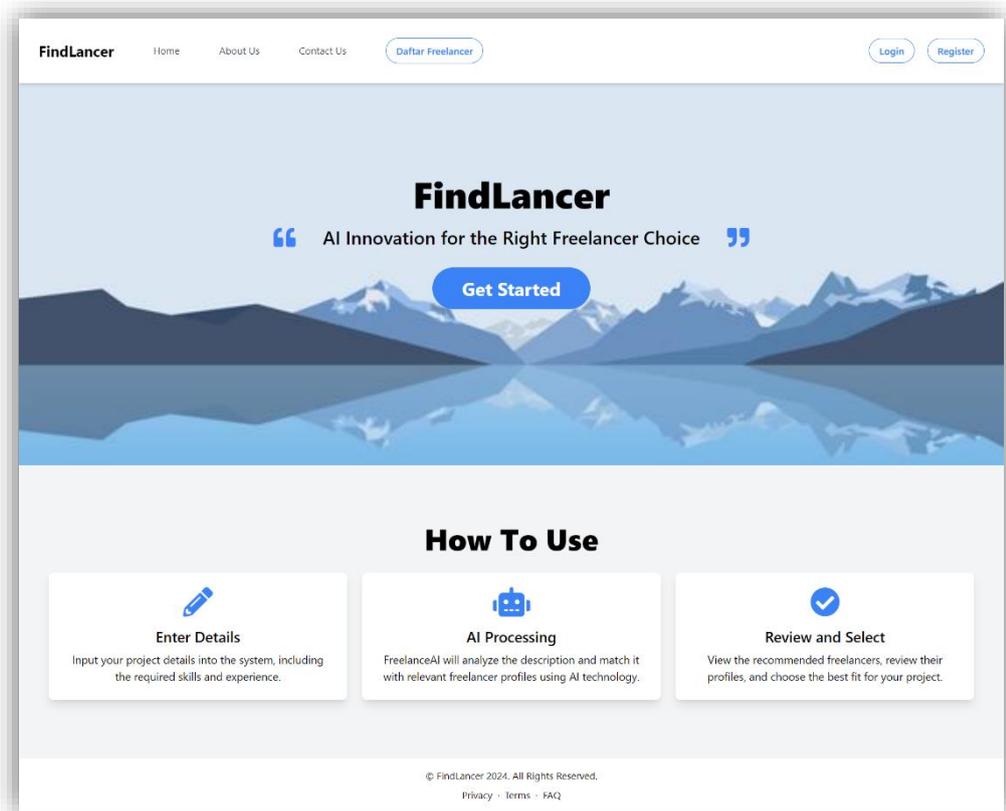
- Dapat mengelola profil pribadi
- Dapat melakukan pencarian rekomendasi *freelancer*
- Dapat menyimpan data *freelancer* ke dalam daftar *bookmark*
- Dapat mengunjungi halaman *freelancer*
- Dapat menghubungi *freelancer*
- Dapat merekrut *freelancer*
- Dapat melihat *project* individu dan *team*

### 3) Freelancer

- Dapat mengelola profil pribadi
- Dapat mengelola profil *freelancer*
- Dapat melihat *project* individu dan *team*

### 3. Halaman Index

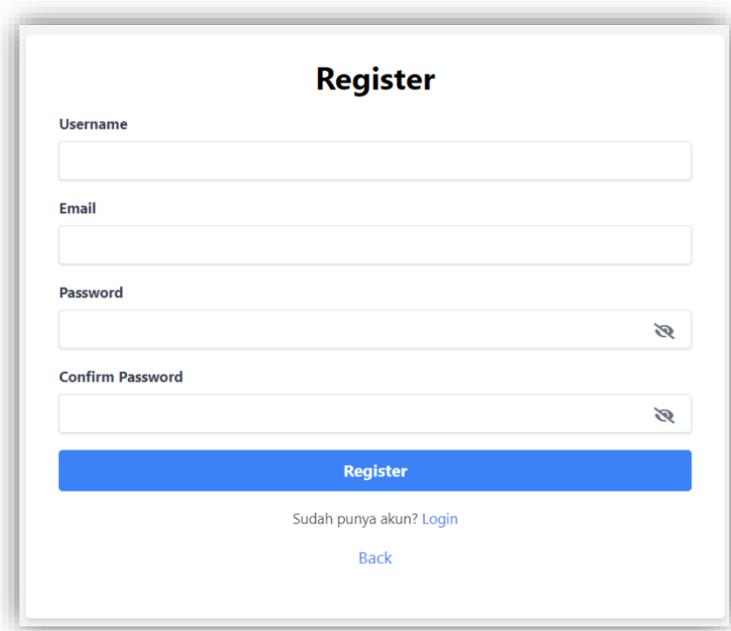
Pada halaman ini terdapat 3 bagian, yaitu top bar, konten dan *footer*. Pada bagian top bar terdapat nama *website* yaitu FindLancer. Terdapat 3 menu utama yaitu menu *home*, *about us* dan *contact us*. Terdapat *button* daftar *freelancer* dan terdapat juga 2 *button* yaitu *button login* dan *register*. Pada bagian *content* sendiri terdapat 2 bagian juga, bagian pertama konten berupa *button get started* yang digunakan untuk memulai ke halaman rekomendasi. Pada bagian kedua konten terdapat *informasi* berupa cara menggunakan sistem rekomendasi ini. Pada bagian *footer* berupa *footer website* pada umumnya. Halaman Index ini dapat dilihat pada gambar 3.1 di bawah ini.



Gambar 3.1 Halaman Index

#### 4. Halaman *Register*

Jika *user* ingin melakukan *login*, tetapi *user* belum mendaftar atau belum mempunyai akun, maka *user* tidak akan bisa *login*. Karena itu, *user* yang belum mempunyai akun diharuskan untuk mendaftar akun terlebih dahulu di halaman *register*. Pada halaman ini, terdapat *form register* yang dimana *user* harus memasukkan *username*, alamat email dan *password*. Jika *user* sudah berhasil mendaftar akun, *user* akan diarahkan ke halaman *login*. Tampilan *Form Register* dapat dilihat pada gambar 4.1 di bawah ini.



The image shows a web registration form titled "Register". It contains four input fields: "Username", "Email", "Password", and "Confirm Password". The "Password" and "Confirm Password" fields have a small icon of a crossed-out key, indicating they are password fields. Below the input fields is a blue button labeled "Register". At the bottom of the form, there is a link "Sudah punya akun? Login" and a "Back" link.

Gambar 4.1 *Form Register*

#### 5. Halaman *Login*

Pada halaman ini, terdapat *form login* Dimana *user* harus memasukkan alamat email dan *password* yang sesuai Ketika mendaftar akun. Jika *user* masuk dengan *role* admin, maka akan diarahkan ke halaman *dashboard* admin. Dan jika *user* masuk dengan *role* pengguna, maka akan diarahkan ke halaman *home*. Tampilan *Form Login* dapat dilihat pada gambar 5.1 di bawah ini.

**Login**

Email

Password

**Login**

[Belum punya akun? Daftar](#)

[Lupa Password?](#)

[Back](#)

Gambar 5.1 *Form Login*

## 6. Admin

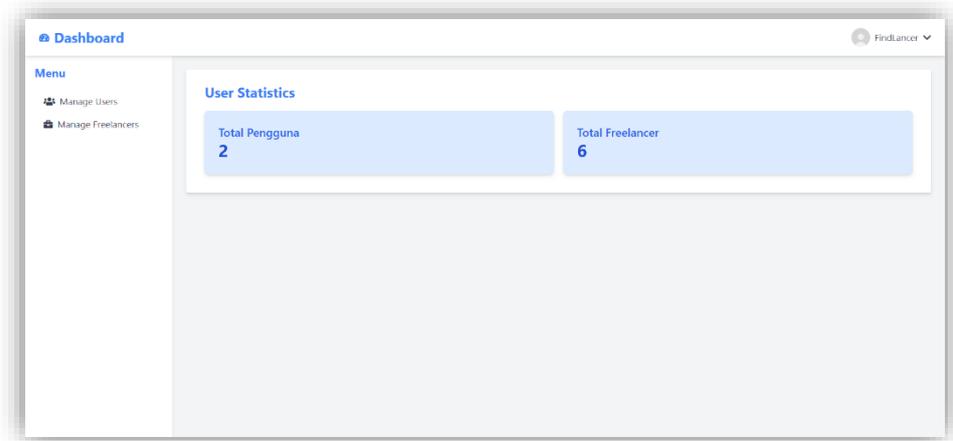
Jika masuk sebagai admin, maka akan diarahkan di halaman *dashboard* admin, berikut beberapa halaman yang terdapat pada admin.

### 1) Halaman *Dashboard* Admin

Pada halaman ini, admin akan disajikan dengan jumlah data berupa jumlah data *user* dan jumlah *freelancer*. Pada bagian top bar terdapat nama system dan *username* admin. Pada bagian *side bar* terdapat menu *manage users* dan menu *manage freelancers*.

- Menu *Manage Users* akan mengarah ke halaman *manage users*
- Menu *Manage Freelancers* akan mengarah ke halaman *manage freelancers*

Halaman *Dashboard* Admin dapat dilihat pada gambar 6.1 di bawah ini.

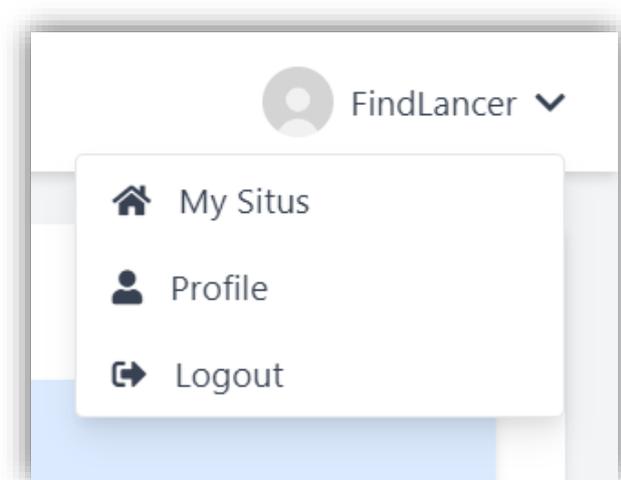


Gambar 6.1 Halaman *Dashboard* Admin

Pada *username* admin terdapat menu *dropdown* berupa :

- Menu *MySitus* yang akan mengarah ke halaman *home*,
- Menu *Profile* yang akan mengarahkan ke halaman profil,
- Menu *Logout* untuk keluar akun

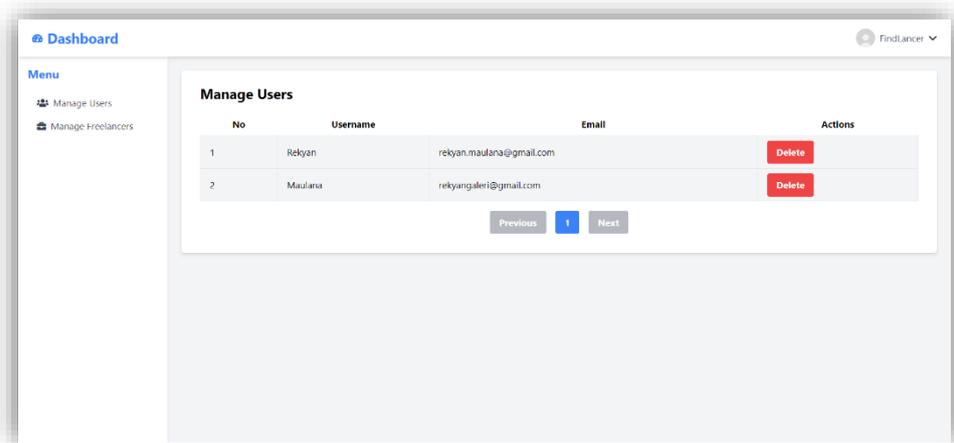
Menu *dropdown* dibagian top bar pada halaman *dashboard* admin dapat dilihat pada gambar 6.2 di bawah ini.



Gambar 6.2 Menu *Dropdown Dashboard* Admin

## 2) Halaman *Manage User*

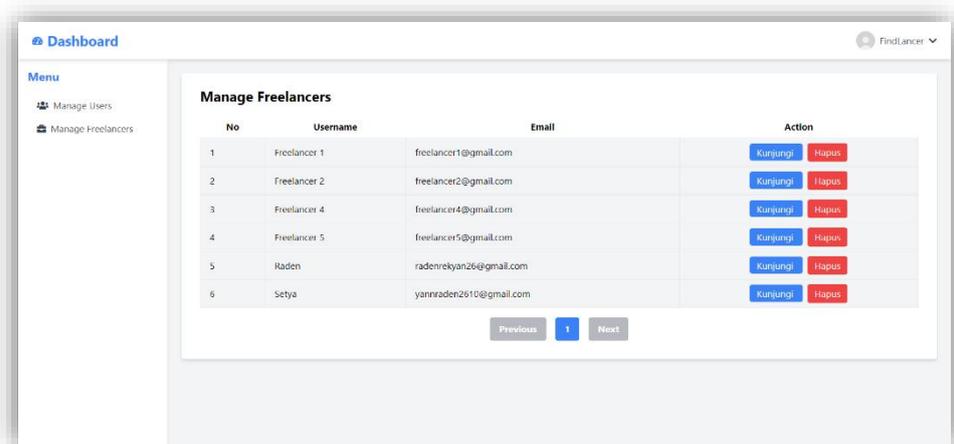
Pada halaman ini, terdapat table *user* berupa nama *user*, email dan *button delete*. *Button delete* digunakan untuk menghapus data *user*. Halaman *Manage User* dapat dilihat pada gambar 6.3 di bawah ini.



Gambar 6.3 Halaman *Manage Users*

## 3) Halaman *Manage Freelancers*

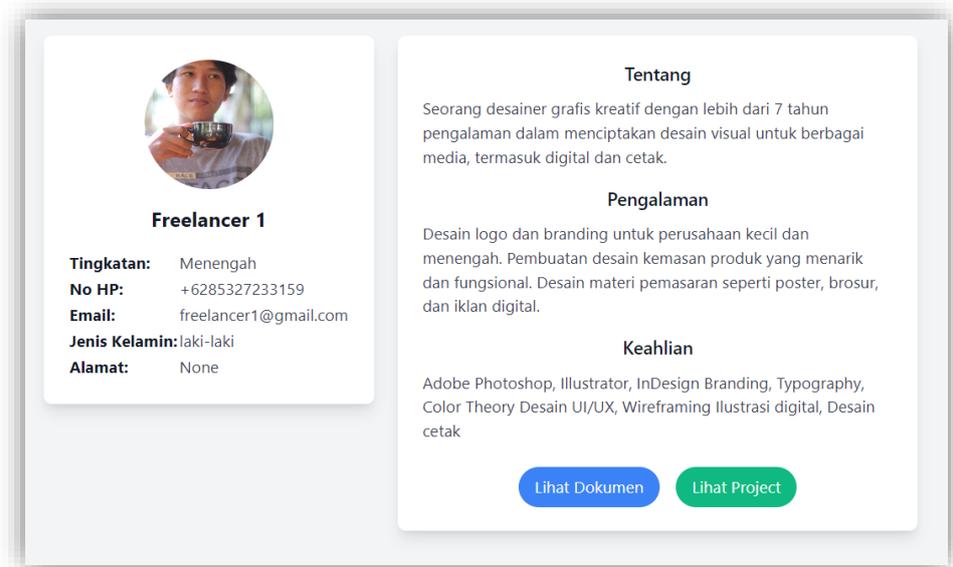
Pada halaman ini, terdapat table *freelancers* berupa *username*, email, *button kunjungi* dan *button delete*. *Button kunjungi* akan mengarahkan ke halaman *freelancer* dan *button delete* digunakan untuk menghapus data *freelancer*. Halaman *Manage freelancer* dapat dilihat pada gambar 6.4 di bawah ini.



Gambar 6.4 Halaman *Manage Freelancers*

#### 4) Halaman *Freelancer*

Pada halaman ini terdapat informasi seputar *freelancer*. Halaman *freelancer* dapat dilihat pada gambar 6.5 dibawah ini.



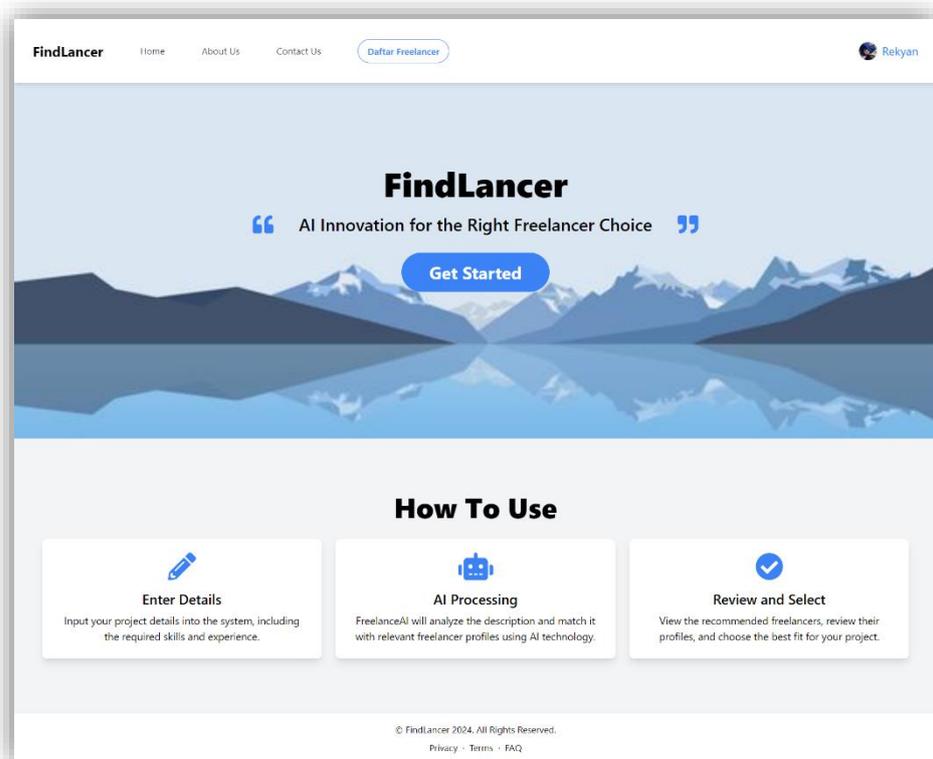
Gambar 6.5 Halaman *Freelancer*

#### 7. Pengguna

Jika masuk sebagai pengguna, maka akan diarahkan ke halaman *home*, berikut beberapa halaman yang terdapat pada pengguna.

##### 1) Halaman *Home*

Pada Halaman *Home* ini sama halnya dengan halaman *index*, yang membedakan adalah pada bagian *top bar*, *button login* dan *register* berganti dengan nama pengguna. Halaman *Home* dapat dilihat pada gambar 7.1 di bawah ini.

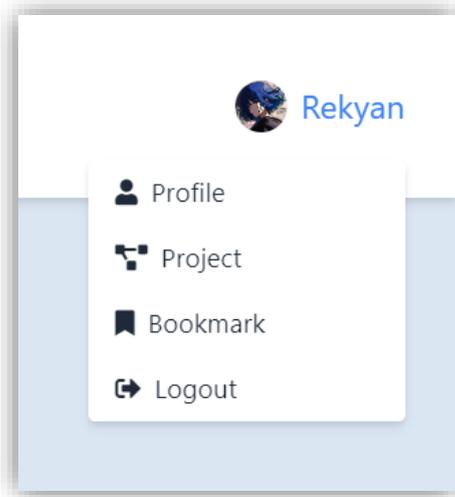


Gambar 7.1 Halaman *Home*

Pada bagian nama pengguna terdapat menu dropdown berupa :

- Menu *Profile* yang akan mengarah ke halaman *profile*
- Menu *Bookmark* yang akan mengarah ke halaman *bookmark*
- Menu *Project* yang akan mengarah ke halaman *project*
- Menu *Logout* untuk keluar akun

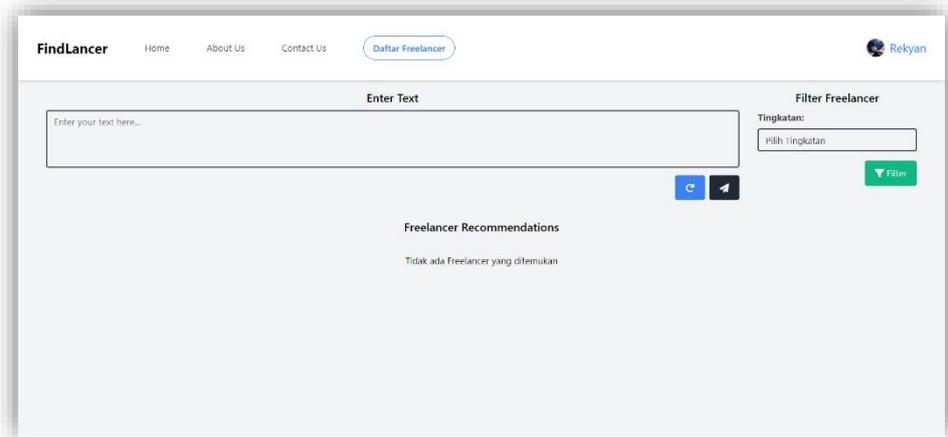
Menu *Dropdown* di bagian top bar pada halaman *home* dapat dilihat pada gambar 7.2 di bawah ini.



Gambar 7.2 Menu *Dropdown* Halaman *Home*

## 2) Halaman Rekomendasi

Pada Halaman Rekomendasi terdapat beberapa bagian diantaranya bagian *input* teks, bagian filter tingkatan, dan bagian rekomendasi *freelancer*. Halaman Rekomendasi dapat dilihat pada gambar 7.3 di bawah ini.



Gambar 7.3 Halaman Rekomendasi

## 3) *Input* Teks

Pada bagian ini, pengguna akan memasukkan teks berupa deskripsi dari *freelancer* yang dibutuhkan. Bagian ini juga terdapat *button* submit dan *refresh*. *Button* submit digunakan untuk melakukan pencarian hasil

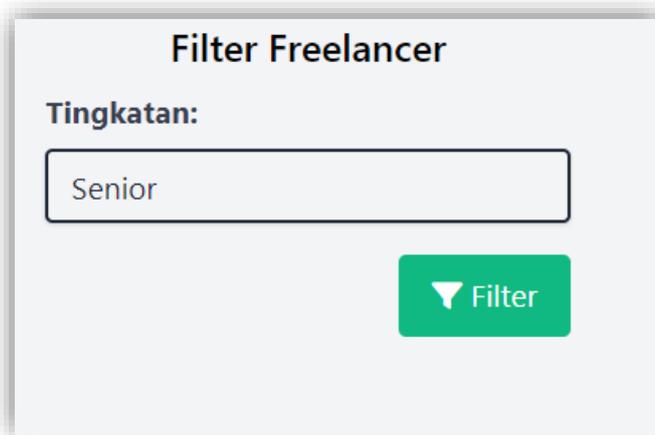
rekomendasi, sedangkan *button refresh* digunakan untuk memuat ulang bagian *input* dan hasil. Tampilan *Form Input* Teks dapat dilihat pada gambar 7.4 di bawah ini.

A screenshot of a web form titled "Enter Text". It features a large text input field with a light gray border and a light gray background. Inside the field, there is a line of placeholder text: "Web developer yang dapat menggunakan HTML, CSS, JavaScript, dan berbagai framework seperti React dan Angular." To the right of the input field, there are two small icons: a blue circular refresh icon and a dark gray square icon with a white arrow pointing right.

Gambar 7.4 *Form Input* Teks

#### 4) Filter *Freelancer*

Pada bagian ini, pengguna memasukkan tingkatan yang akan dilakukan *filtering* untuk data *freelancer*. Filter *Freelancer* dapat dilihat pada gambar 7.5 di bawah ini.

A screenshot of a form titled "Filter Freelancer". The form has a light gray background. At the top, the title "Filter Freelancer" is centered. Below the title, the label "Tingkatan:" is positioned to the left of a text input field. The input field contains the text "Senior". Below the input field, there is a green rectangular button with a white downward-pointing triangle icon and the text "Filter" in white.

Gambar 7.5 Filter *Freelancer*

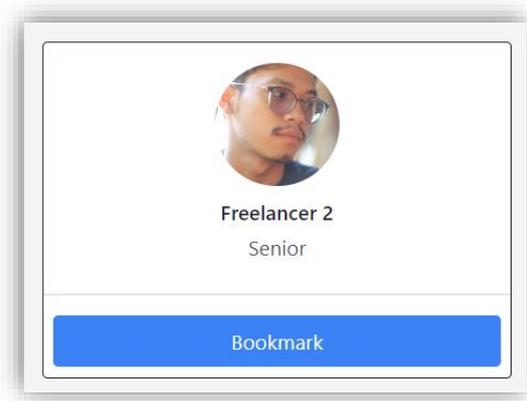
#### 5) Hasil Rekomendasi

Pada bagian Hasil Rekomendasi terdapat hasil *freelancer* berupa *card*. Dimana dalam *card* tersebut berupa:

- *Image Profile Freelancer*
- *Username*
- Tingkatan

- *Button Bookmark*

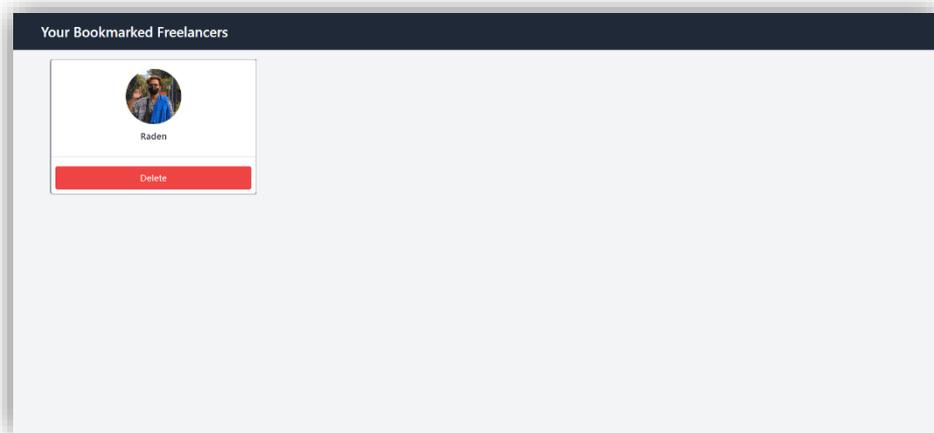
*Button bookmark* digunakan untuk menyimpan data *freelancer* tersebut ke dalam halaman *bookmark*. Bagian Hasil Rekomendasi dapat dilihat pada gambar 7.6 di bawah ini.



Gambar 7.6 Hasil Rekomendasi

#### 6) Halaman *Bookmark*

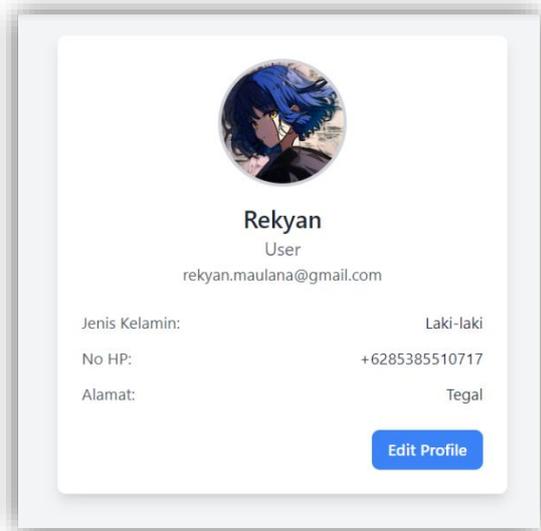
Pada Halaman *Bookmark* terdapat data *freelancer* yang telah disimpan oleh pengguna. Data *freelancer* ini berupa card yang sama dengan yang ada dibagian hasil rekomendasi. Halaman *Bookmark* ini dapat dilihat pada gambar 7.7 di bawah ini.



Gambar 7.7 Halaman *Bookmark*

### 7) Halaman *Profile* Pribadi

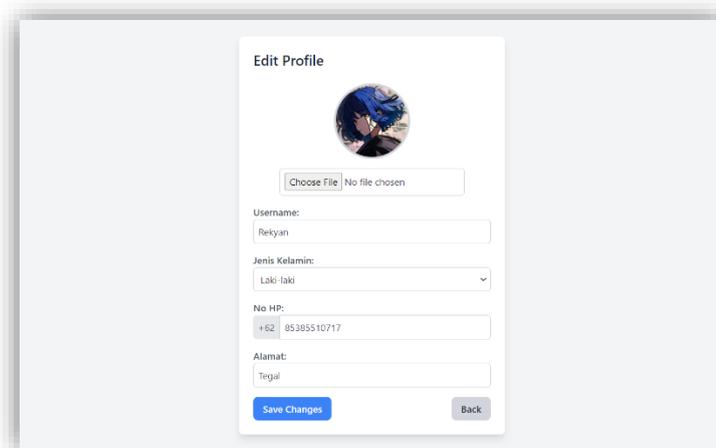
Pada Halaman *Profile* Pribadi ini terdapat informasi dari *user* seperti *username*, *role*, email, jenis kelamin, No. hp, alamat bahkan *image profile*. Halaman *Profile* Pribadi ini dapat dilihat pada gambar 7.8 di bawah ini.



Gambar 7.8 Halaman *Profile*

### 8) Halaman *Edit Profile* Pribadi

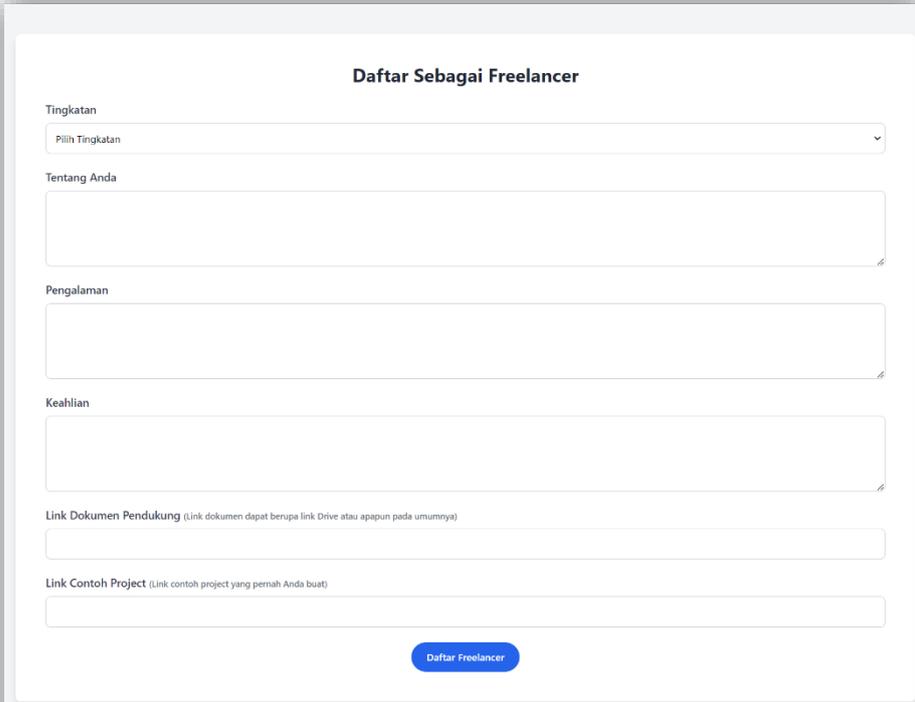
Pada halaman ini hanya terdapat *form edit profile* pribadi untuk *user*. *Form Edit Profile* Pribadi dapat dilihat pada gambar 7.9 di bawah ini.



Gambar 7.9 *Form Edit Profile*

### 9) Halaman Daftar *Freelancer*

Pada Halaman Daftar *Freelancer* ini terdapat informasi dari *freelancer* seperti tingkatan, tentang, pengalaman, keahlian bahkan *link* dokumen dan *project* yang perlu diisi sehingga pengguna dapat menjadi *freelancer*. Halaman Daftar *Freelancer* ini dapat dilihat pada gambar 7.10 di bawah ini.



The image shows a registration form titled "Daftar Sebagai Freelancer". The form contains the following fields:

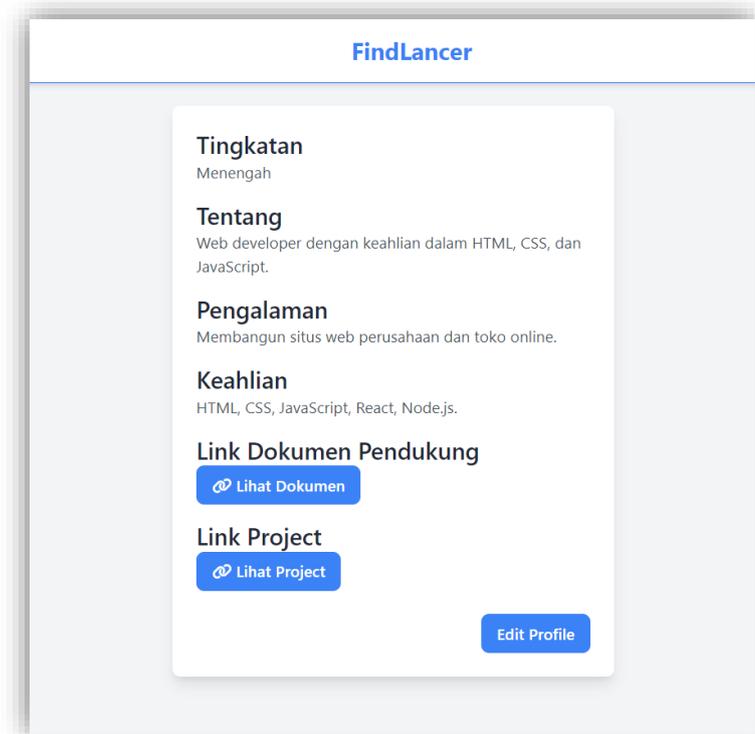
- Tingkatan**: A dropdown menu with the placeholder text "Pilih Tingkatan".
- Tentang Anda**: A large text input field.
- Pengalaman**: A large text input field.
- Keahlian**: A large text input field.
- Link Dokumen Pendukung**: A text input field with a sub-label "(Link dokumen dapat berupa link Drive atau apapun pada umumnya)".
- Link Contoh Project**: A text input field with a sub-label "(Link contoh project yang pernah Anda buat)".

At the bottom of the form is a blue button labeled "Daftar Freelancer".

Gambar 7.10 Halaman Daftar *Freelancer*

## 10) Halaman *Profile Freelancer*

Pada Halaman *Profile Freelancer* ini terdapat informasi dari *freelancer* seperti tingkatan, tentang, pengalaman, keahlian bahkan *link* dokumen dan *project*. Halaman *Profile Freelancer* ini dapat dilihat pada gambar 7.11 di bawah ini.



Gambar 7.11 Halaman *Profile Freelancer*

### 11) Halaman *Edit Profile Freelancer*

Pada halaman ini hanya terdapat *form edit profile freelancer* untuk *freelancer*. *Form Edit Profile Freelancer* dapat dilihat pada gambar 7.12 di bawah ini.

**Edit Freelancer Profile**

Tingkatan  
Menengah

Tentang  
Web developer dengan keahlian dalam HTML, CSS, dan JavaScript.

Pengalaman  
Membangun situs web perusahaan dan toko online.

Keahlian  
HTML, CSS, JavaScript, React, Node.js.

Link Dokumen Pendukung (Link dokumen dapat berupa link Drive atau lainnya)  
<https://drive.google.com/drive/folders/1M2E1sZzcj55v>

Link Contoh Project (Link contoh project yang pernah Anda buat)  
<https://drive.google.com/drive/folders/1M2E1sZzcj55v>

Save Changes Back

Gambar 7.12 *Form Edit Profile Freelancer*

## 12) Halaman *Freelancer Profile*

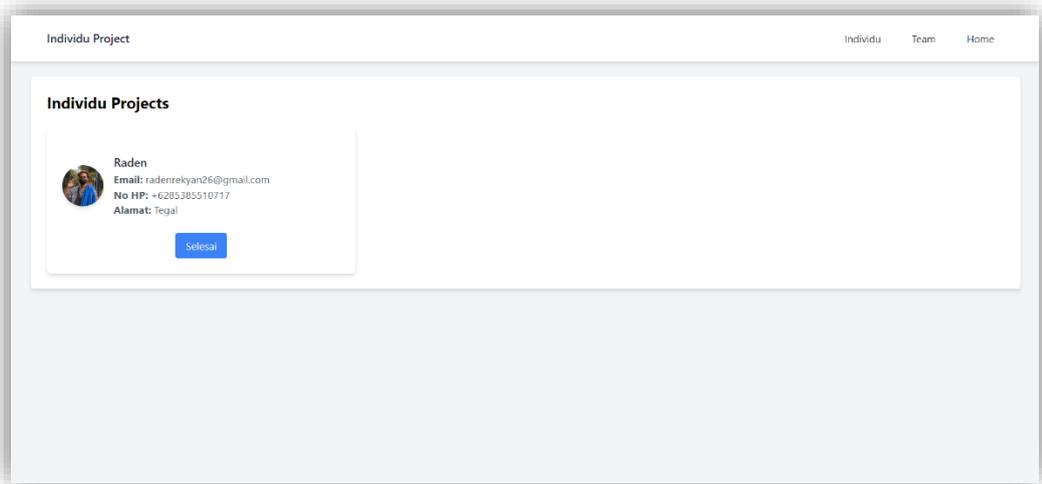
Pada halaman ini hanya terdapat informasi dari *freelancer* dan terdapat *button* hubungi yang digunakan untuk menghubungi *freelancer* serta *button* rekrut untuk merekrut *freelancer*. Halaman *Freelancer Profile* dapat dilihat pada gambar 7.13 di bawah ini.



Gambar 7.13 Halaman *Freelancer Profile*

### 13) Halaman *Project Individu*

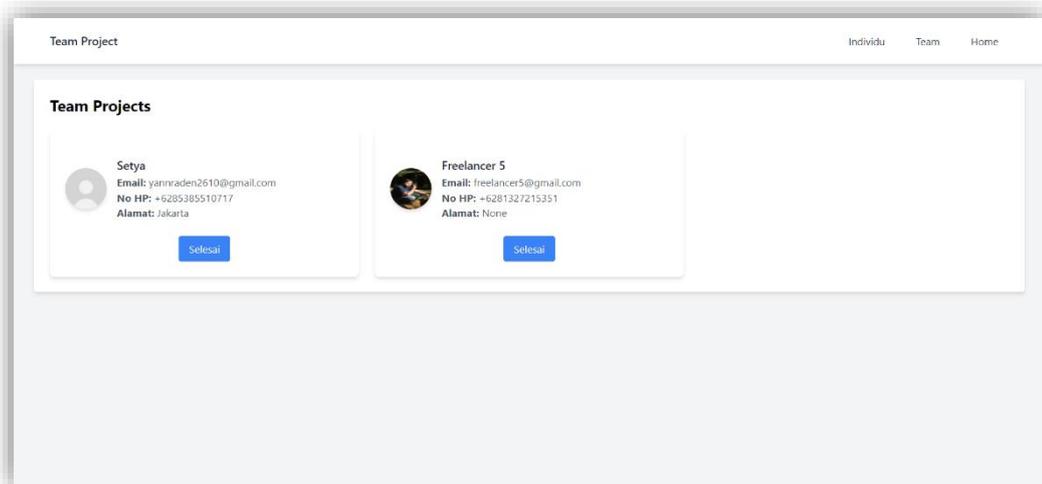
Pada halaman ini hanya terdapat *freelancer* yang direkrut oleh *user* secara individu. Halaman *Project Individu* dapat dilihat pada gambar 7.14 di bawah ini.



Gambar 7.14 Halaman *Project Individu*

### 14) Halaman *Project Team*

Pada halaman ini hanya terdapat *freelancer* yang direkrut oleh *user* secara *team*. Halaman *Project Team* dapat dilihat pada gambar 7.15 di bawah ini.



Gambar 7.14 Halaman *Project Individu*



# **FINDLANCER**

## **TECHNICAL BOOK**

R. REKYAN MAULANA SETYAWAN

2024

## A. Pendahuluan

FindLancer adalah sebuah aplikasi web yang didesain untuk merekomendasikan *freelancer* dan layanan sesuai dengan kebutuhan pengguna. Aplikasi ini menggunakan teknologi *machine learning* untuk mencocokkan pengguna dengan layanan *freelance* yang paling cocok. Dibangun menggunakan *framework* Flask untuk *backend* dan HTML serta *Tailwind CSS* untuk *frontend*, aplikasi ini mencakup peran pengguna seperti admin dan pengguna biasa, masing-masing dengan fungsionalitas yang spesifik.

## B. Spesifikasi

### 1. Spesifikasi *Hardware*

Untuk menjalankan aplikasi web FindLancer dengan lancar, berikut adalah spesifikasi *hardware* yang disarankan:

- RAM: Minimal 4 GB (disarankan 8 GB)
- *Storage*: SSD 128 GB atau lebih
- *Network*: Koneksi internet yang stabil

### 2. Spesifikasi *Software*

Untuk pengembangan dan *deployment* aplikasi FindLancer, berikut adalah spesifikasi *software* yang diperlukan:

- a) Sistem Operasi
  - Windows: Windows 10 atau lebih baru
- b) Bahasa Pemrograman dan *Framework*
  - Python: Versi 3.7 atau lebih tinggi
  - Flask: Versi 2.0 atau lebih tinggi
- c) *Database*
  - MySQL: Versi 5.7 atau lebih tinggi
- d) *Library*
  - Flask Extensions:
  - Flask-Bcrypt: Untuk *hashing password*
  - Flask-SQLAlchemy: Untuk ORM

e) *Machine Learning*

- tensorflow: Untuk implementasi model LSTM
- scikit-learn: Untuk *preprocessing* dan evaluasi model

f) *Frontend*

- HTML: Untuk struktur halaman web
- *Tailwind* CSS: Untuk *styling* halaman web
- JavaScript: Untuk interaktivitas halaman web

g) *Alat*

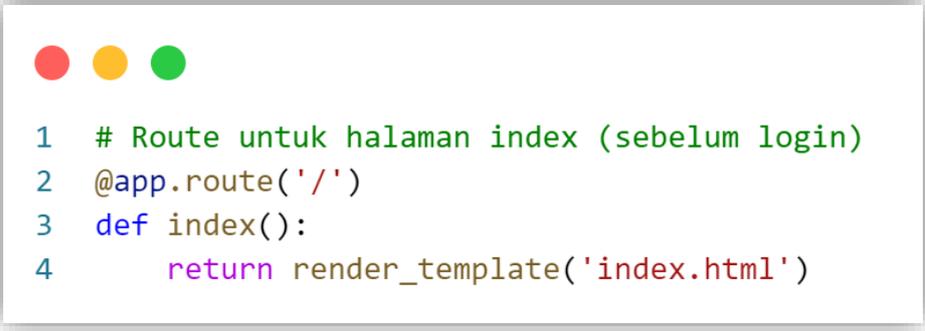
- *Visual Studio Code*: Editor untuk pengembangan kode
- *Google Colaboratory*: Untuk pengembangan model *machine learning*

### C. *Source Code*

Berikut penjelasan mengenai *source code website* FindLancer, yang meliputi komponen dan fitur utama:

#### 1. Halaman Index dan *Home*

Untuk halaman index dibagi menjadi dua *route* yaitu *route* ketika *user* belum *login* dan *route* ketika *user* sudah *login*. Berikut penjelasan *code route* untuk halaman index dan *home*. Pada *route* '/' akan menampilkan *index.html* pada fungsi *index* yang dimana fungsi ini untuk *user* yang belum *login*. *Code route* index dapat dilihat pada gambar 1.1 di bawah.



```
1 # Route untuk halaman index (sebelum login)
2 @app.route('/')
3 def index():
4     return render_template('index.html')
```

Gambar 1.1 *Route* Halaman Index

Pada *route* `"/home"` akan menampilkan `index.html` pada fungsi *home*, tetapi fungsi ini dipanggil untuk *user* yang sudah *login*. *Code route home* dapat dilihat pada gambar 1.2 di bawah ini.

A screenshot of a code editor window with a white background and a grey border. At the top left, there are three colored circles: red, yellow, and green. Below them, there is a list of four lines of code. Line 1 is a comment: '# Route untuk halaman home (setelah login)'. Line 2 is '@app.route('/home')'. Line 3 is 'def home():'. Line 4 is 'return render\_template('index.html')'. The code is color-coded: '#' is green, '@app' is blue, 'route' is red, '/' is black, 'home' is black, 'def' is blue, 'home' is black, '():' is black, 'return' is purple, 'render\_template' is red, and 'index.html' is black.

```
1 # Route untuk halaman home (setelah login)
2 @app.route('/home')
3 def home():
4     return render_template('index.html')
```

Gambar 1.2 *Route Halaman Home*

Perbedaan sebelum *login* dan sesudah *login* terletak pada bagian topbar di halaman `index` dan `home`. *Source code* `topbar.html` dapat dilihat pada gambar 1.3 di bawah ini.

```

1 <!-- topbar.html -->
2 <div class="bg-white shadow-md py-8 flex justify-between items-center w-full px-4 md:px-8 fixed top-0 z-10">
3   <div class="flex items-center space-x-16">
4     <a href="{{ url_for('index')}}">
5       <h1 class="text-2xl font-bold">FindLancer</h1>
6     </a>
7     <div class="hidden md:flex space-x-16">
8       <a href="{{ url_for('home')}}" class="text-gray-800 hover:text-gray-600 py-2">Home</a>
9       <a href="{{ url_for('about')}}" class="text-gray-800 hover:text-gray-600 py-2">About Us</a>
10      <a href="{{ url_for('contact')}}" class="text-gray-800 hover:text-gray-600 py-2">Contact Us</a>
11      {% if session.get('role') != 'freelancer' %}
12      <a href="{{ url_for('daftar_freelancer')}}"
13      class="text-blue-500 hover:text-white font-semibold py-2 px-4 border border-blue-500 hover:bg-blue-500 rounded-full">
14      Daftar Freelancer</a>
15      {% endif %}
16    </div>
17  </div>
18  <div class="relative">
19    {% if session.logged_in %}
20      <button id="dropdownButton" class="flex items-center text-xl py-2 text-blue-500">
21        
23        {{ session['username'] }}
24      </button>
25      <div id="dropdownMenu" class="hidden bg-white shadow-md rounded mt-2 absolute right-0 w-48">
26        {% if session.role == 'admin' %}
27          <a href="{{ url_for('dashboard_admin')}}" class="flex items-center px-4 py-2 text-gray-800 hover:bg-gray-200">
28            <i class="fas fa-tachometer-alt mr-2"></i> Dashboard Admin
29          </a>
30          {% endif %}
31          <a href="{{ url_for('profile')}}" class="flex items-center px-4 py-2 text-gray-800 hover:bg-gray-200">
32            <i class="fas fa-user mr-2"></i> Profile
33          </a>
34          <a href="{{ url_for('project_individu')}}" class="flex items-center px-4 py-2 text-gray-800 hover:bg-gray-200">
35            <i class="fas fa-project-diagram mr-2"></i> Project
36          </a>
37          <a href="{{ url_for('bookmark')}}" class="flex items-center px-4 py-2 text-gray-800 hover:bg-gray-200">
38            <i class="fas fa-bookmark mr-2"></i> Bookmark
39          </a>
40          <a href="{{ url_for('logout')}}" class="flex items-center px-4 py-2 text-gray-800 hover:bg-gray-200">
41            <i class="fas fa-sign-out-alt mr-2"></i> Logout
42          </a>
43        </div>
44      {% else %}
45        <a href="{{ url_for('login')}}"
46        class="text-blue-500 hover:text-white font-semibold py-2 px-4 border border-blue-500 hover:bg-blue-500 rounded-full">Login</a>
47        <a href="{{ url_for('register')}}"
48        class="text-blue-500 hover:text-white font-semibold py-2 px-4 border border-blue-500 hover:bg-blue-500 rounded-full ml-4">Register</a>
49      {% endif %}
50    </div>
51    <button class="md:hidden text-gray-800 hover:text-gray-600" id="mobileMenuButton">
52      <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke="currentColor" class="w-6 h-6">
53        <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M4 6h16M4 12h16M4 18h16" />
54      </svg>
55    </button>
56  </div>
57
58 <!-- Mobile Menu -->
59 <div id="mobileMenu" class="hidden bg-white shadow-md p-4 fixed top-20 w-full z-10">
60   <a href="{{ url_for('home')}}" class="block px-4 py-2 text-gray-800 hover:bg-gray-200">Home</a>
61   <a href="{{ url_for('about')}}" class="block px-4 py-2 text-gray-800 hover:bg-gray-200">About Us</a>
62   <a href="{{ url_for('contact')}}" class="block px-4 py-2 text-gray-800 hover:bg-gray-200">Contact Us</a>
63   {% if session.get('role') != 'freelancer' %}
64   <a href="{{ url_for('daftar_freelancer')}}"
65   class="block px-4 py-2 text-blue-500 hover:text-white font-semibold border border-blue-500 hover:bg-blue-500 rounded-full">Daftar Freelancer</a>
66   {% endif %}
67 </div>

```

Gambar 1.3 Source Code Topbar Html

## 2. Halaman *Register*

*Source code* ini merupakan *route* untuk halaman *register*. Pada *route* ini dijelaskan bahwa email yang didaftarkan oleh *user* adalah email yang belum didaftarkan sebelumnya dan password akan dilakukan hashing. *Source code route register* dapat dilihat pada gambar 2.1 di bawah ini.

```
1 @app.route('/register', methods=['GET', 'POST'])
2 def register():
3     form = RegistrationForm()
4     if form.validate_on_submit():
5         existing_user = User.query.filter_by(email=form.email.data).first()
6
7         if existing_user:
8             flash('Email already registered. Please use a different email.', 'danger')
9         else:
10            hashed_password = bcrypt.generate_password_hash(form.password.data).decode('utf-8')
11            new_user = User(
12                username=form.username.data,
13                email=form.email.data,
14                password=hashed_password,
15                confirmed=False
16            )
17            db.session.add(new_user)
18            db.session.commit()
19
20            # Generate confirmation token and send email
21            token = generate_confirmation_token(new_user.email)
22            send_confirmation_email(new_user.email, token)
23
24            flash('A confirmation email has been sent. Please check your email.', 'success')
25            return redirect(url_for('login'))
26
27            return render_template('auth/register.html', form=form)
```

Gambar 2.1 *Route* Halaman *Register*

*Source* ini merupakan *code* html untuk halaman *register*. *Source code* halaman *register.html* dapat dilihat pada gambar 2.2 di bawah ini.

```

1 <div class="w-full max-w-xs sm:max-w-sm md:max-w-md lg:max-w-lg xl:max-w-xl 2xl:max-w-2xl">
2   <form class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4" method="POST" action="{{ url_for('register') }}">
3     {{ form.hidden_tag() }}
4     <h1 class="text-3xl font-bold mb-4 text-center">Register</h1>
5
6     <!-- Flash messages for errors -->
7     {% with messages = get_flashed_messages(with_categories=true) %}
8       {% if messages %}
9         {% for category, message in messages %}
10          <div class="bg-red-100 border border-red-400 text-red-700 px-4 py-3 rounded mb-4" role="alert">
11            <span class="block sm:inline">{{ message }}</span>
12          </div>
13        {% endfor %}
14      {% endif %}
15    {% endwith %}
16
17    <div class="mb-4">
18      <label for="username" class="block text-gray-700 text-sm font-bold mb-2">Username</label>
19      {{ form.username(class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight
20        focus:outline-none focus:shadow-outline") }}
21      {% for error in form.username.errors %}
22        <span class="text-red-500 text-xs italic">{{ error }}</span>
23      {% endfor %}
24    </div>
25    <div class="mb-4">
26      <label for="email" class="block text-gray-700 text-sm font-bold mb-2">Email</label>
27      {{ form.email(class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight
28        focus:outline-none focus:shadow-outline") }}
29      {% for error in form.email.errors %}
30        <span class="text-red-500 text-xs italic">{{ error }}</span>
31      {% endfor %}
32    </div>
33    <div class="mb-4">
34      <label for="password" class="block text-gray-700 text-sm font-bold mb-2">Password</label>
35      <div class="relative">
36        {{ form.password(id="password", class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight
37          focus:outline-none focus:shadow-outline") }}
38        <span id="toggle-password" class="absolute right-0 top-0 mt-2 mr-4 cursor-pointer text-gray-500 hover:text-gray-700"
39          onclick="togglePassword('password')">
40          <i id="icon-password" class="fa fa-eye-slash"></i>
41        </span>
42      </div>
43      {% for error in form.password.errors %}
44        <span class="text-red-500 text-xs italic">{{ error }}</span>
45      {% endfor %}
46    </div>
47    <div class="mb-4">
48      <label for="confirm_password" class="block text-gray-700 text-sm font-bold mb-2">Confirm Password</label>
49      <div class="relative">
50        {{ form.confirm_password(id="confirm_password", class="shadow appearance-none border rounded
51          w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline") }}
52        <span id="toggle-confirm_password" class="absolute right-0 top-0 mt-2 mr-4
53          cursor-pointer text-gray-500 hover:text-gray-700" onclick="togglePassword('confirm_password')">
54          <i id="icon-confirm_password" class="fa fa-eye-slash"></i>
55        </span>
56      </div>
57      {% for error in form.confirm_password.errors %}
58        <span class="text-red-500 text-xs italic">{{ error }}</span>
59      {% endfor %}
60    </div>
61    <div class="mb-4">
62      <button type="submit" class="bg-blue-500 hover:bg-blue-700
63        text-white font-bold py-2 px-4 rounded focus:outline-none
64        focus:shadow-outline w-full">Register</button>
65    </div>
66
67    <div class="text-center mb-4">
68      <p class="text-gray-600 text-sm">Sudah punya akun? <a href="{{ url_for('login') }}"
69        class="text-blue-500 hover:underline">Login</a></p>
70    </div>
71  </form>
72 </div>

```

Gambar 2.2 Source Code Register Html

### 3. Halaman Login

*Source code* ini merupakan *route* untuk halaman *login*. Pada *route* ini berisi logika untuk *login user*. Ketika *user login* dengan *role* admin maka akan diarahkan ke halaman dashboard admin, selain itu akan diarahkan ke halaman *home*. *Source code route login* dapat dilihat pada gambar 3.1 di bawah ini.

```
1 @app.route('/login', methods=['GET', 'POST'])
2 def login():
3     form = LoginForm()
4     if form.validate_on_submit():
5         user = User.query.filter_by(email=form.email.data).first()
6
7         if user:
8             if not user.confirmed:
9                 flash('Your email is not confirmed. Please confirm your email before logging in.', 'warning')
10                return redirect(url_for('unconfirmed'))
11
12            if bcrypt.check_password_hash(user.password, form.password.data):
13                session['logged_in'] = True
14                session['username'] = user.username
15                session['role'] = user.role
16                session['user_id'] = user.id
17                session['email'] = user.email
18                print(f"Email saved in session: {session['email']}") # Verifikasi email di terminal
19
20            # Cek apakah profil sudah ada
21            profile = Profile.query.filter_by(id_user=user.id).first()
22            if not profile:
23                # Jika profil tidak ada, buat profil baru dengan img_profil 'default.jpg'
24                profile = Profile(id_user=user.id, img_profil='default.jpg')
25                db.session.add(profile)
26                db.session.commit()
27                session['img_profil'] = 'default.jpg'
28            else:
29                session['img_profil'] = profile.img_profil
30
31            if user.role == 'admin':
32                return redirect(url_for('dashboard_admin', login_success=True))
33            else:
34                return redirect(url_for('home', login_success=True))
35        else:
36            flash('Login Unsuccessful. Please check email and password', 'danger')
37    else:
38        flash('Login Unsuccessful. Please check email and password', 'danger')
39
40    return render_template('auth/login.html', form=form)
```

Gambar 3.1 *Route* Halaman *Login*

*Source code* ini merupakan *code* html untuk halaman *login*. *Source code* halaman *login.html* dapat dilihat pada gambar 3.2 di bawah ini.

```

1 <div class="w-full max-w-xs sm:max-w-sm md:max-w-md lg:max-w-lg xl:max-w-xl 2xl:max-w-2xl">
2 <form class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4" method="POST" action="{{ url_for('login') }}">
3   {{ form.hidden_tag() }}
4   <h1 class="text-3xl font-bold mb-4 text-center">Login</h1>
5
6   <!-- Flash messages for errors -->
7   {% with messages = get_flashed_messages(with_categories=true) %}
8     {% if messages %}
9       {% for category, message in messages %}
10        <div class="bg-red-100 border border-red-400 text-red-700 px-4 py-3 rounded mb-4" role="alert">
11          <span class="block sm:inline">{{ message }}</span>
12        </div>
13      {% endfor %}
14    {% endif %}
15  {% endwith %}
16
17  <div class="mb-4">
18    <label for="email" class="block text-sm font-medium text-gray-700">Email</label>
19    {{ form.email(class="form-input mt-1 block w-full px-3 py-2 border border-gray-300 rounded-md shadow-sm
20    focus:outline-none focus:border-blue-500 focus:ring focus:ring-blue-500 focus:ring-opacity-50") }}
21    {% for error in form.email.errors %}
22      <span class="text-red-500 text-xs italic">{{ error }}</span>
23    {% endfor %}
24  </div>
25  <div class="mb-4">
26    <label for="password" class="block text-sm font-medium text-gray-700">Password</label>
27    <div class="relative">
28      {{ form.password(id="password", class="form-input mt-1 block w-full px-3 py-2 border border-gray-300 rounded-md
29      shadow-sm focus:outline-none focus:border-blue-500 focus:ring focus:ring-blue-500 focus:ring-opacity-50") }}
30      <span id="toggle-password" class="absolute right-0 top-0 mt-2 mr-4 cursor-pointer text-gray-500 hover:text-gray-700"
31      onclick="togglePassword('password')">
32        <i id="icon-password" class="fa fa-eye-slash"></i>
33      </span>
34    </div>
35    {% for error in form.password.errors %}
36      <span class="text-red-500 text-xs italic">{{ error }}</span>
37    {% endfor %}
38  </div>
39  <div class="mb-4">
40    <button type="submit" class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none
41    focus:shadow-outline w-full">Login</button>
42  </div>
43
44  <div class="text-center mb-4">
45    <p class="text-sm text-gray-600">Belum punya akun? <a href="{{ url_for('register') }}"
46    class="text-blue-500 hover:underline">Daftar</a></p>
47  </div>
48 </form>
49 </div>

```

Gambar 3.2 Source Code Login Html

#### 4. Dashboard Admin

Source code ini merupakan route untuk halaman dashboard admin. Pada route ini terdapat variable total users untuk menghitung jumlah keseluruhan role pengguna dan variable total freelancers untuk menghitung jumlah jumlah freelancers. Source code route halaman dashboard admin dapat dilihat pada gambar 4.1 di bawah ini.

```

1 @app.route('/dashboard_admin')
2 def dashboard_admin():
3     if session.get('role') != 'admin':
4         return redirect(url_for('login'))
5
6     username = session.get('username')
7     total_users = User.query.filter_by(role='pengguna').count()
8     total_freelancers = User.query.filter_by(role='freelancer').count()
9
10    return render_template('admin/dashboard_admin.html',
11                          username=username, total_users=total_users, total_freelancers=total_freelancers)

```

Gambar 4.1 *Route* Halaman Dashboard Admin

*Source code* ini merupakan *code* html untuk halaman dashboard admin yang menampilkan jumlah total *users* dan total *freelancers*. *Source code* halaman `dashboard_admin.html` dapat dilihat pada gambar 4.2 di bawah ini.

```

1 <!-- Content -->
2 <div id="content" class="flex-1 p-4 lg:p-6 lg:ml-64 overflow-auto">
3   <div class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">
4     <h2 class="text-2xl font-bold mb-4 text-blue-500">User Statistics</h2>
5     <div class="grid grid-cols-1 md:grid-cols-2 gap-4">
6       <div class="bg-blue-100 p-6 rounded-lg shadow-md">
7         <h3 class="text-xl font-semibold text-blue-600">Total Pengguna</h3>
8         <p class="text-3xl font-bold text-blue-700">{{ total_users }}</p>
9       </div>
10      <div class="bg-blue-100 p-6 rounded-lg shadow-md">
11        <h3 class="text-xl font-semibold text-blue-600">Total Freelancer</h3>
12        <p class="text-3xl font-bold text-blue-700">{{ total_freelancers }}</p>
13      </div>
14    </div>
15  </div>
16 </div>

```

Gambar 4.2 *Source Code* Dashboard Admin Html

## 5. Manage Users

*Source code* ini merupakan *route* untuk halaman *manage users*. Pada *route* ini terdapat *variable* `list` pengguna untuk menampilkan semua *user* dengan *role*. *Source code route* halaman *manage users* dapat dilihat pada gambar 5.1 di bawah ini.

```

1 @app.route('/manage_users')
2 def manage_users():
3     if session.get('role') != 'admin':
4         return redirect(url_for('login'))
5
6     username = session.get('username')
7     page = request.args.get('page', 1, type=int)
8     pengguna_list = User.query.filter_by(role='pengguna').paginate(page=page, per_page=10, error_out=False)
9
10    return render_template('admin/manage_users.html', pengguna_list=pengguna_list, username=username)

```

Gambar 5.1 *Route Halaman Manage Users*

*Source code* ini merupakan *code* html untuk halaman *manage users* yang menampilkan *table* berupa nama dan email pengguna. *Source code* halaman *manage\_users.html* dapat dilihat pada gambar 5.2 di bawah ini.

```

1 <div class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">
2     <h2 class="text-2xl font-bold mb-4">Manage Users</h2>
3     <div class="overflow-auto" style="max-height: 400px;">
4         <table class="table-auto w-full">
5             <thead>
6                 <tr>
7                     <th class="px-4 py-2">No</th>
8                     <th class="px-4 py-2">Username</th>
9                     <th class="px-4 py-2">Email</th>
10                    <th class="px-4 py-2">Actions</th>
11                </tr>
12            </thead>
13            <tbody>
14                {% for user in pengguna_list %}
15                <tr class="bg-gray-100">
16                    <td class="border px-4 py-2">{{ loop.index }}</td>
17                    <td class="border px-4 py-2">{{ user.username }}</td>
18                    <td class="border px-4 py-2">{{ user.email }}</td>
19                    <td class="border px-4 py-2">
20                        <form action="{{ url_for('delete_user', user_id=user.id) }}"
21                            method="POST" style="display:inline;">
22                            <button type="submit"
23                                class="bg-red-500 hover:bg-red-700 text-white
24                                    font-bold py-2 px-4 rounded">Delete</button>
25                        </form>
26                    </td>
27                </tr>
28                {% endfor %}
29            </tbody>
30        </table>
31    </div>
32</div>

```

Gambar 5.2 *Source Code Manage Users Html*

## 6. Manage *Freelancers*

*Source code* ini merupakan *route* untuk halaman manage *freelancers*. Pada *route* ini terdapat *variable list freelancer* untuk menampilkan semua data *freelancer*. *Source code route* halaman manage *freelancers* dapat dilihat pada gambar 6.1 di bawah ini.

```
1 @app.route('/manage_freelancers')
2 def manage_freelancers():
3     if session.get('role') != 'admin':
4         return redirect(url_for('login'))
5
6     username = session.get('username')
7     page = request.args.get('page', 1, type=int)
8     # Ensure joinedload for related models
9     freelancer_list = User.query.filter_by(role='freelancer')\
10         .outerjoin(User.profile)\
11         .outerjoin(Profile.freelancer)\
12         .paginate(page=page, per_page=10, error_out=False)
13
14     return render_template('admin/manage_freelancers.html', freelancer_list=freelancer_list, username=username)
```

Gambar 6.1 *Route Halaman Manage Freelancers*

*Source code* ini merupakan *code html* untuk halaman manage *freelancers* yang menampilkan *table* data *freelancer*. *Source code* halaman `manage_freelancers.html` dapat dilihat pada gambar 6.2 di bawah ini.

```

1 <h2 class="text-2xl font-bold mb-4">Manage Freelancers</h2>
2 <div class="overflow-auto" style="max-height: 400px;">
3 <table class="table-auto w-full">
4 <thead>
5 <tr>
6 <th class="px-4 py-2">No</th>
7 <th class="px-4 py-2">Username</th>
8 <th class="px-4 py-2">Email</th>
9 <th class="px-4 py-2">Action</th> <!-- Action Column -->
10 </tr>
11 </thead>
12 <tbody>
13 <!-- for user in freelancer_list.items %>
14 <tr class="bg-gray-100">
15 <td class="border px-4 py-2">{{ loop.index + (freelancer_list.page - 1) * freelancer_list.per_page }}</td>
16 <td class="border px-4 py-2">{{ user.username }}</td>
17 <td class="border px-4 py-2">{{ user.email }}</td>
18 <td class="border px-4 py-2 text-center flex flex-col md:flex-row items-center space-y-2 md:space-y-0 md:space-x-2">
19 <!-- Tombol Kunjungi -->
20 <!-- if user.profile and user.profile.freelancer %>
21 <a href="{{ url_for('info_fl', freelancer_id=user.profile.freelancer.id) }}"
22 class="bg-blue-500 text-white py-1 px-3 rounded hover:bg-blue-600 transition duration-300 text-center">
23 Kunjungi
24 </a>
25 <!-- else %>
26 <span class="bg-gray-300 text-gray-600 py-1 px-3 rounded">N/A</span>
27 <!-- endif %>
28
29 <!-- Delete Button -->
30 <!-- if user.profile and user.profile.freelancer %>
31 <form action="{{ url_for('delete_freelancer', freelancer_id=user.profile.freelancer.id) }}"
32 method="POST" class="inline-block ml-2">
33 <button type="submit">
34 class="bg-red-500 text-white py-1 px-3 rounded hover:bg-red-600 transition duration-300 text-center">
35 Hapus
36 </button>
37 </form>
38 <!-- else %>
39 <span class="bg-gray-300 text-gray-600 py-1 px-3 rounded">N/A</span>
40 <!-- endif %>
41 </td>
42 </tr>
43 <!-- % endfor %>
44 </tbody>
45 </table>
46 </div>

```

Gambar 6.2 Source Code Manage Freelancers Html

## 7. Halaman Rekomendasi

*Source code* pada halaman rekomendasi memiliki beberapa fungsi. Sebelum melakukan rekomendasi, muat model dan *tokenizer* yang dihasilkan pada *machine learning*. Berikut *source code* untuk memanggil model dan *tokenizer* dapat dilihat pada gambar 7.1 di bawah ini.

```

1 # Memuat model dan tokenizer
2 model = load_model('models/model.h5')
3 with open('models/tokenizer.pkl', 'rb') as file:
4     tokenizer = pickle.load(file)
5 with open('models/mlb.pkl', 'rb') as file:
6     mlb = pickle.load(file)

```

Gambar 7.1 Load Model

Pada halaman ini terdapat fungsi ‘*clean text*’ untuk melakukan pembersihan teks dari teks baru yang dimasukkan oleh *user*. *Source code* untuk membersihkan teks dapat dilihat pada gambar 7.2 di bawah ini.

```
1 # Fungsi untuk membersihkan teks
2 def clean_text(text):
3     lemmatizer = WordNetLemmatizer()
4     text = text.lower()
5     text = re.sub(r'^a-z0-9\s', '', text)
6     words = word_tokenize(text)
7     words = [lemmatizer.lemmatize(word) for word in words if word not in stopwords.words('indonesian')
8             and word not in stopwords.words('english')]
9     return ' '.join(words)
```

Gambar 7.2 Fungsi untuk Membersihkan Teks

Terdapat fungsi ‘*predict keywords*’ untuk memprediksi kata kunci dari teks yang dimasukkan oleh *user*. *Source code* untuk memprediksi kata kunci dapat dilihat pada gambar 7.3 di bawah ini.

```
1 # Fungsi untuk melakukan inferensi dengan model yang dimuat kembali
2 def predict_keywords(paragraph):
3     # Preprocessing teks paragraf
4     paragraph = clean_text(paragraph)
5     paragraph_sequence = tokenizer.texts_to_sequences([paragraph])
6     max_sequence_length = model.input_shape[1]
7     paragraph_sequence_padded = pad_sequences(paragraph_sequence, maxlen=max_sequence_length, padding='post')
8
9     # Melakukan prediksi
10    predictions = model.predict(paragraph_sequence_padded)
11
12    # Menggunakan threshold untuk menentukan kata kunci yang diprediksi
13    threshold = 0.3
14    keywords = mlb.classes_[predictions[0] > threshold]
15
16    return keywords.tolist()
```

Gambar 7.3 Fungsi Prediksi Kata Kunci

Terdapat juga fungsi ‘*recommend*’ untuk menampilkan data *freelancer* yang telah dicocokkan dengan hasil kata kunci. *Source code* pada fungsi ‘*recommend*’ dapat dilihat pada gambar 7.4 di bawah.

```
1 def recommend():
2     if not session.get('logged_in'):
3         return redirect(url_for('login'))
4
5     user_id = session.get('user_id')
6     user = User.query.get(user_id)
7
8     if request.method == 'POST':
9         paragraph = request.form['paragraph']
10        predicted_keywords = predict_keywords(paragraph)
11        predicted_keywords_str = ' '.join(predicted_keywords)
12
13        # Query to join freelancers, profile, and users tables
14        freelancers = db.session.query(
15            Freelancer.id,
16            Freelancer.tingkatan,
17            Freelancer.tentang,
18            Freelancer.pengalaman,
19            Freelancer.keahlian,
20            Profile.img_profil,
21            User.username
22        ).join(Profile, Freelancer.id_profile == Profile.id).join(User, Profile.id_user == User.id).all()
23
24        jasa = [
25            f"{freelancer.tentang}.{freelancer.pengalaman}.{freelancer.keahlian}"
26            for freelancer in freelancers
27        ]
28
29        tfidf_vectorizer = TfidfVectorizer()
30        tfidf_matrix = tfidf_vectorizer.fit_transform(jasa)
31        predicted_keyword_vector = tfidf_vectorizer.transform([predicted_keywords_str])
32        similarities = cosine_similarity(predicted_keyword_vector, tfidf_matrix)
33        threshold = max(similarities[0]) * 0.8
34        top_freelancer_indices = [i for i, similarity in enumerate(similarities[0]) if similarity > threshold]
35        sorted_indices = sorted(top_freelancer_indices, key=lambda x: similarities[0][x], reverse=True)
36
37        recommendations = []
38        for index in sorted_indices:
39            freelancer = freelancers[index]
40            is_bookmarked = Bookmark.query.filter_by(id_user=user.id, id_freelancer=freelancer.id).first() is not None
41            recommendations.append({
42                'id': freelancer.id,
43                'tingkatan': freelancer.tingkatan,
44                'tentang': freelancer.tentang,
45                'pengalaman': freelancer.pengalaman,
46                'keahlian': freelancer.keahlian,
47                'username': freelancer.username,
48                'img_profil': freelancer.img_profil,
49                'is_bookmarked': is_bookmarked
50            })
51
52        return render_template('rekomendasi.html', recommendations=recommendations, paragraph=paragraph)
53
54    return render_template('rekomendasi.html')
```

Gambar 7.4 Route Halaman Rekomendasi

Terdapat juga *source code* untuk menampilkan halaman rekomendasi.html. Pada halaman ini terdapat bagian *input* teks dan hasil rekomendasi. Berikut *source code* pada bagian *input* teks dapat dilihat pada gambar 7.5 di bawah ini.

```
1 <!-- Input -->
2 <div class="w-full md:w-4/5 px-4">
3 <h2 class="text-xl font-semibold mb-2 text-center pt-4">Enter Text</h2>
4 <form action="/recommend" method="POST" class="w-full">
5 <div class="mb-2">
6 <textarea id="paragraph" name="paragraph" rows="4"
7 class="bg-gray-100 shadow appearance-none border-2
8 border-gray-800 rounded w-full py-2 px-3 placeholder-gray-500 text-gray-800
9 leading-tight focus:outline-none focus:bg-white focus:border-blue-500"
10 placeholder="Enter your text here...">{{ paragraph }}</textarea>
11 </div>
12 <div class="flex justify-end mb-4">
13 <button type="button" id="refreshButton"
14 class="bg-blue-500 hover:bg-blue-600 text-white py-2 px-4 rounded border-2
15 border-blue-500 hover:border-blue-600 transition">
16 <i class="fas fa-redo"></i>
17 </button>
18 <button type="submit" value="get recommendations"
19 class="bg-gray-800 hover:bg-gray-900 text-white py-2 px-4 rounded border-2
20 border-gray-800 hover:border-gray-900 ml-2 transition">
21 <i class="fas fa-paper-plane"></i>
22 </button>
23 </div>
24 </form>
25 </div>
```

Gambar 7.5 *Source Code Input Text Html*

Pada halaman ini terdapat hasil rekomendasi. *Source code* untuk menampilkan hasil rekomendasi dapat dilihat pada gambar 7.6 di bawah ini.

```

1 <!-- Card -->
2 {% if recommendations %}
3 <h2 class="text-xl font-semibold mb-8 text-center pt-4">Freelancer Recommendations</h2>
4 <div id="freelancerCards" class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-4 mx-4">
5   {% for recommendation in recommendations %}
6     <div class="relative border border-gray-800 bg-white text-gray-800 rounded overflow-hidden"
7       data-tingkatan="{{ recommendation.tingkatan }}">
8       <a href="{{ url_for('profile_fl', freelancer_id=recommendation.id) }}">
9         <div class="p-4 flex flex-col items-center">
10          
12          <p class="font-semibold text-center mt-2">{{ recommendation.username }}</p>
13
14          <!-- Tambahkan Tingkatan di bawah Username -->
15          <p class="text-gray-600 mt-1">
16            {% if recommendation.tingkatan == 'junior' %}
17              Junior
18            {% elif recommendation.tingkatan == 'menengah' %}
19              Menengah
20            {% elif recommendation.tingkatan == 'senior' %}
21              Senior
22            {% endif %}
23          </p>
24        </div>
25      </a>
26      <hr class="border-t border-gray-300 my-2"> <!-- Garis pemisah -->
27      <div class="p-2"> <!-- Container untuk tombol -->
28        <button type="button" class="bookmark-btn py-2 px-4 rounded text-white w-full"
29          onclick="addBookmark('{{ recommendation.id }})"
30          style="background-color: {% if recommendation.is_bookmarked %} #fbbf24; {% else %} #3b82f6; {% endif %};">
31          {% if recommendation.is_bookmarked %} Bookmarked {% else %} Bookmark {% endif %}
32        </button>
33      </div>
34    </div>
35  {% endfor %}
36 </div>
37 {% else %}
38 <h2 class="text-xl font-semibold mb-8 text-center pt-4">Freelancer Recommendations</h2>
39 <p class="text-center">Tidak ada Freelancer yang ditemukan</p>
40 {% endif %}

```

Gambar 7.6 Source Code Hasil Rekomendasi

## 8. Filter Tingkatan *Freelancer*

Filter Tingkatan *Freelancer* ini terletak pada halaman rekomendasi. Berikut *source code* untuk tampilan bagian filter tingkatan *freelancer* yang dapat dilihat pada gambar 8.1 di bawah ini.

```

<div class="w-full md:w-1/5 px-4">
  <h2 class="text-xl font-semibold mb-2 text-center pt-4">Filter Freelancer</h2>
  <div class="mb-4">
    <label for="tingkatanFilter" class="block text-gray-700 font-bold mb-2">
      Tingkatan:
    </label>
    <select id="tingkatanFilter" class="bg-gray-100 shadow appearance-none border-2 border-gray-800
      rounded w-full py-2 px-3 text-gray-800 leading-tight focus:outline-none
      focus:bg-white focus:border-blue-500">
      <option value="">Pilih Tingkatan</option>
      <option value="junior">Junior</option>
      <option value="menengah">Menengah</option>
      <option value="senior">Senior</option>
    </select>
  </div>
  <div class="flex justify-end mb-4">
    <button type="button" id="filterButton"
      class="bg-green-500 hover:bg-green-600 text-white py-2 px-4 rounded border-2
      border-green-500 hover:border-green-600 transition">
      <i class="fas fa-filter"></i> Filter
    </button>
  </div>
</div>

```

Gambar 8.1 *Source Code Tingkatan Freelancer*

## 9. Halaman *Bookmark*

*Source code* ini merupakan *route* untuk halaman *bookmark*. Pada *route* ini terdapat *variable* berupa data *freelancer* yang ditambahkan oleh *user*. *Source code route* halaman *bookmark* dapat dilihat pada gambar 9.1 di bawah ini.

```

1 @app.route('/bookmark')
2 def bookmark():
3     if not session.get('logged_in'):
4         return redirect(url_for('login'))
5
6     user_id = session.get('user_id')
7
8     # Ambil freelancer yang dibookmark oleh pengguna
9     bookmarks = Bookmark.query.filter_by(id_user=user_id).all()
10
11     # Ambil freelancer berdasarkan ID yang ada di bookmark
12     freelancer_ids = [bookmark.id_freelancer for bookmark in bookmarks]
13
14     # Query untuk mendapatkan freelancers dan username
15     # Bergabung dengan tabel Profile dan User melalui relasi yang jelas
16     bookmarked_freelancers = db.session.query(Freelancer, User.username).\
17         join(Profile, Freelancer.id_profile == Profile.id).\
18         join(User, Profile.id_user == User.id).\
19         filter(Freelancer.id.in_(freelancer_ids)).\
20         all()
21
22     return render_template('bookmark.html', freelancers=bookmarked_freelancers)

```

Gambar 9.1 *Route Halaman Bookmark*

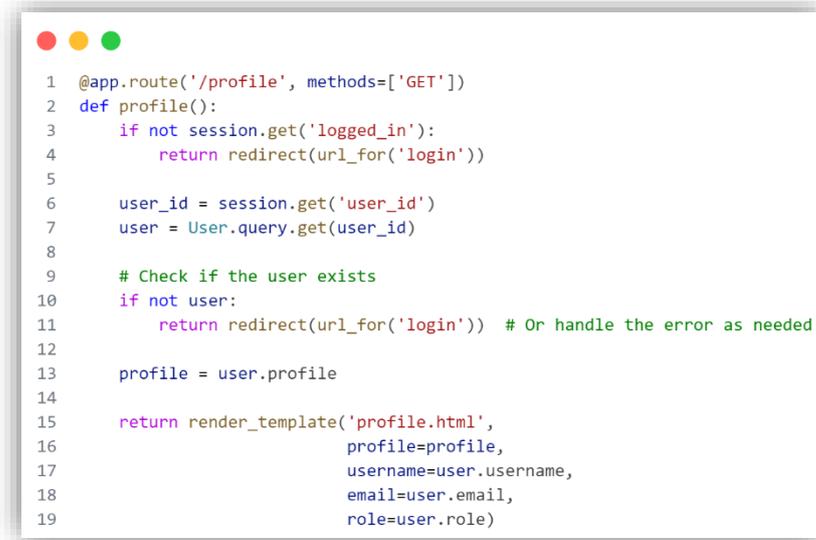
*Source code* ini merupakan *code* html untuk halaman *bookmark* yang menampilkan data *freelancer* yang telah ditambahkan oleh *user*. *Source code* halaman *bookmark.html* dapat dilihat pada gambar 9.2 di bawah ini.

```
1 <!-- Main Content -->
2 <div class="container mx-auto px-4 mt-4">
3 <!-- Freelancer Cards -->
4 <div id="freelancerCards" class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-4 mx-4">
5   {% for freelancer, username in freelancers %}
6   <div class="relative border border-gray-800 bg-white text-gray-800 rounded overflow-hidden">
7     <a href="{{ url_for('profile_fl', freelancer_id=freelancer.id) }}">
8       <div class="p-4 flex flex-col items-center">
9         
11        <p class="font-semibold text-center mt-2">{{ username }}</p>
12      </div>
13    </a>
14    <hr class="border-t border-gray-300 my-2"> <!-- Separator Line -->
15    <div class="p-2"> <!-- Container for the button -->
16      <button type="button" class="delete-btn py-2 px-4 rounded text-white w-full"
17      onclick="deleteBookmark({{ freelancer.id }})" style="background-color: #ef4444;" <!-- Red -->
18        Delete
19      </button>
20    </div>
21  </div>
22  {% endfor %}
23 </div>
24 </div>
```

Gambar 9.2 *Source Code Bookmark Html*

## 10. Halaman *Profile* Pribadi

*Source code* ini merupakan *route* untuk halaman *profile* pribadi. Pada *route* ini terdapat *variable* berupa *username* dan email yang akan ditampilkan pada halaman *profile*. *Source code route* halaman *profile* dapat dilihat pada gambar 10.1 di bawah ini.

A screenshot of a code editor window showing Python code for a Django route. The code is numbered from 1 to 19. It defines a route for '/profile' using the GET method. The profile() function checks if the user is logged in; if not, it redirects to the login page. It then retrieves the user ID from the session, queries the user, and checks if the user exists. If not, it redirects to the login page. Finally, it retrieves the user's profile and renders the 'profile.html' template with the user's profile, username, email, and role.

```
1 @app.route('/profile', methods=['GET'])
2 def profile():
3     if not session.get('logged_in'):
4         return redirect(url_for('login'))
5
6     user_id = session.get('user_id')
7     user = User.query.get(user_id)
8
9     # Check if the user exists
10    if not user:
11        return redirect(url_for('login')) # Or handle the error as needed
12
13    profile = user.profile
14
15    return render_template('profile.html',
16                           profile=profile,
17                           username=user.username,
18                           email=user.email,
19                           role=user.role)
```

Gambar 10.1 *Route* Halaman *Profile* Pribadi

*Source code* ini merupakan *code* html untuk halaman *profile* yang menampilkan *username* dan email. *Source code* halaman *profile.html* dapat dilihat pada gambar 10.2 di bawah ini.

```

1 <div class="bg-white rounded-lg shadow-lg p-6 max-w-md w-full mx-auto">
2   <div class="flex flex-col items-center">
3     
5     <h2 class="text-2xl font-semibold text-gray-800">{{ username }}</h2>
6     <p class="text-gray-500 text-lg">
7       {% if role == 'admin' %}
8         Admin
9       {% elif role == 'pengguna' %}
10        User
11      {% elif role == 'freelancer' %}
12        Freelancer
13      {% endif %}
14    </p>
15    <p class="text-gray-600">{{ email }}</p>
16  </div>
17
18  <div class="mt-6">
19    <div class="flex items-center justify-between">
20      <span class="text-gray-600">Jenis Kelamin:</span>
21      <span class="text-gray-800">
22        {% if profile.jenis_kelamin is not none %}
23          {{ 'Perempuan' if profile.jenis_kelamin == 'perempuan' else 'Laki-laki' }}
24        {% else %}
25          -
26        {% endif %}
27      </span>
28    </div>
29    <div class="flex items-center justify-between mt-3">
30      <span class="text-gray-600">No HP:</span>
31      <span class="text-gray-800">+62{{ profile.no_hp or '-' }}</span>
32    </div>
33    <div class="flex items-center justify-between mt-3">
34      <span class="text-gray-600">Alamat:</span>
35      <span class="text-gray-800">{{ profile.alamat or '-' }}</span>
36    </div>
37  </div>
38
39  <div class="mt-6 flex justify-end">
40    <a href="{{ url_for('edit_profile') }}"
41    class="bg-blue-500 hover:bg-blue-600 text-white font-semibold py-2 px-4 rounded-lg">
42      Edit Profile
43    </a>
44  </div>
45 </div>

```

Gambar 10.2 Source Code Profile Pribadi Html

## 11. Halaman *Profile Freelancer*

*Source code* ini merupakan *route* untuk halaman *profile freelancer*. Pada *route* ini terdapat informasi dari *freelancer* yang akan ditampilkan pada halaman *profile freelancer*. *Source code route* halaman *profile freelancer* dapat dilihat pada gambar 11.1 di bawah ini.

```
1 @app.route('/freelancer_profile')
2 def freelancer_profile():
3     if not session.get('logged_in'):
4         return redirect(url_for('login'))
5
6     role = session.get('role')
7     user_id = session.get('user_id')
8
9     # Cek apakah role adalah freelancer
10    if role != 'freelancer':
11        flash('You do not have access to this page.', 'danger')
12        return redirect(url_for('home'))
13
14    user = User.query.get(user_id)
15
16    if not user:
17        flash('User not found.', 'danger')
18        return redirect(url_for('login'))
19
20    profile = user.profile
21    freelancer_profile = profile.freelancer
22
23    return render_template('freelancer_profile.html',
24                           freelancer_profile=freelancer_profile,
25                           role=role)
```

Gambar 11.1 *Route Halaman Profile Freelancer*

*Source code* ini merupakan *code html* untuk halaman *profile freelancer* yang menampilkan informasi *freelancer*. *Source code* halaman *profile.html* dapat dilihat pada gambar 11.2 di bawah ini.

```

1 <div class="bg-white rounded-lg shadow-lg p-6 max-w-md w-full mx-auto">
2   <div class="mb-4">
3     <h2 class="text-2xl font-semibold text-gray-800">Tingkatan</h2>
4     <p class="text-gray-600">
5       {% if freelancer_profile.tingkatan == 'junior' %}
6         Junior
7       {% elif freelancer_profile.tingkatan == 'menengah' %}
8         Menengah
9       {% elif freelancer_profile.tingkatan == 'senior' %}
10        Senior
11      {% else %}
12        Belum diisi
13      {% endif %}
14    </p>
15  </div>
16  <div class="mb-4">
17    <h2 class="text-2xl font-semibold text-gray-800">Tentang</h2>
18    <p class="text-gray-600">{{ freelancer_profile.tentang or 'Belum diisi' }}</p>
19  </div>
20  <div class="mb-4">
21    <h2 class="text-2xl font-semibold text-gray-800">Pengalaman</h2>
22    <p class="text-gray-600">{{ freelancer_profile.pengalaman or 'Belum diisi' }}</p>
23  </div>
24  <div class="mb-4">
25    <h2 class="text-2xl font-semibold text-gray-800">Keahlian</h2>
26    <p class="text-gray-600">{{ freelancer_profile.keahlian or 'Belum diisi' }}</p>
27  </div>
28  <div class="mb-4">
29    <h2 class="text-2xl font-semibold text-gray-800">Link Dokumen Pendukung</h2>
30    {% if freelancer_profile.link_dokumen %}
31      <a href="{{ freelancer_profile.link_dokumen }}" target="_blank"
32        class="inline-block bg-blue-500 hover:bg-blue-600 text-white font-semibold py-2 px-4 rounded-lg">
33        <i class="fas fa-link"></i> Lihat Dokumen
34      </a>
35    {% else %}
36      <p class="text-gray-600">Belum diisi</p>
37    {% endif %}
38  </div>
39  <div class="mb-4">
40    <h2 class="text-2xl font-semibold text-gray-800">Link Project</h2>
41    {% if freelancer_profile.link_project %}
42      <a href="{{ freelancer_profile.link_project }}" target="_blank"
43        class="inline-block bg-blue-500 hover:bg-blue-600 text-white font-semibold py-2 px-4 rounded-lg">
44        <i class="fas fa-link"></i> Lihat Project
45      </a>
46    {% else %}
47      <p class="text-gray-600">Belum diisi</p>
48    {% endif %}
49  </div>
50  <div class="mt-6 flex justify-end">
51    <a href="{{ url_for('edit_freelancer_profile') }}"
52      class="bg-blue-500 hover:bg-blue-600 text-white font-semibold py-2 px-4 rounded-lg">
53      Edit Profile
54    </a>
55  </div>
56 </div>

```

Gambar 10.2 Source Code Profile Freelancer Html

## Lampiran 5 Sertifikat HKI yang Terbit

  
**REPUBLIK INDONESIA**  
**KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA**

# SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202466338, 16 Juli 2024

**Pencipta**

Nama : **R. Rekyan Maulana Setyawan**

Alamat : Randusari RT 01/RW 01 Kec.Pagerbarang Kab.Tegal Provinsi Jawa Tengah 52462, Pagerbarang, Tegal, Jawa Tengah, 52462

Kewarganegaraan : Indonesia

**Pemegang Hak Cipta**

Nama : **Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama**

Alamat : Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana 52142, Margadana, Tegal, Jawa Tengah 52142

Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**

Judul Ciptaan : **Penerapan Model LSTM Dan Cosine Similarity Dalam Rekomendasi Freelancer Untuk Meningkatkan Layanan Jasa Berbasis Website**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 16 Juli 2024, di Tegal

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000641693

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.  
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA  
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL  
u.b  
Direktur Hak Cipta dan Desain Industri

  
IGNATIUS M.T. SILALAHI  
NIP. 196812301996031001



Disclaimer:  
Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.