Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Dalam penelitian ini, metodologi penelitian yang dilakukan adalah sebagai berikut.

1. Studi literatur

   Dalam proses ini, dipelajari teori-teori yang berhubungan dengan aplikasi yang dibuat, seperti *Neural Network, backpropagation* dan teori-teori mengenai penyakit *Lower Back Pain.*

2. Wawancara dan konsultasi

   Wawancara dan konsultasi dilakukan untuk mendapatkan gejala-gejala dan jenis-jenis penyakit *Lower Back Pain*. Wawancara dilakukan dengan pakar yang sesungguhnya yaitu Dr. Yudi yang memiliki spesialisasi dibidang penyakit *Lower Back Pain*.

3. Perancangan jaringan syaraf tiruan

   Dari hasil studi dan wawancara tersebut, peneliti merancang jaringan syaraf tiruan yang diterapkan pada aplikasi. Peneliti mulai menentukan jumlah layer, neuron dan *input* yang dibutuhkan di dalam *Neural Network*. Pertanyaan-pertanyaan untuk diberikan kepada pengguna juga didefinisikan pada tahap ini

4. Perancangan sistem

   Pada tahap ini, peneliti merancang alur kerja dari aplikasi seperti data-data yang diperlukan sebagai basis pengetahuan sistem pakar dan tampilan dari aplikasi. Pembuatan diagram-diagram seperti *use case diagram, activity diagram, entity relationship diagram, sequence diagram* dan perancangan *user interface* dilakukan pada tahap ini.

5. Implementasi

   Implementasi *coding* dilakukan berdasarkan hasil rancangan yang telah dibuat sebelumnya. *Coding* dilakukan menggunakan 2 bahasa pemrograman yaitu C# dan PHP. C# digunakan untuk membuat *training application* yang berfungsi sebagai *backend* dan digunakan oleh admin, sedangkan PHP digunakan untuk membuat *web application* yang berfungsi sebagai *frontend* dan digunakan oleh *user*.

6. Melakukan pengujian *(testing)*.

   Setelah pembuatan dan implementasi selesai, maka dilakukan pengujian untuk mengetahui kehandalan (*reliability*) dan kebenaran *output* dari aplikasi yang telah dibangun. *Testing* dilakukan dengan membandingkan *output* dari aplikasi dengan hasil diagnosa dari pakar yang sesungguhnya.

7. Analisa dan pengambilan kesimpulan.

   Setelah melakukan *testing* terhadap aplikasi, hasil kemudian dianalisa. Analisa tersebut berguna untuk mengambil kesimpulan akhir dari penelitian yang telah dilakukan.

3.2 Perancangan Jaringan Syaraf Tiruan

Perancangan jaringan syaraf tiruan dilakukan menggunakan 2 tahap yaitu tahap *training* dan tahap pengenalan. Tahap *training* dilakukan pada aplikasi *training* yang digunakan oleh *admin*. Pada tahap *training*, *admin* akan diminta untuk memasukan *file* data training yang didapat dari seorang pakar. *Neural network* menggunakan data training tersebut untuk *learning* menggunakan algoritma *backpropagation* sehingga mendapatkan hasil yang mendekati atau sama dengan diagnosa yang diberikan oleh seorang pakar.

Selanjutnya, hasil dari proses *learning* yang berupa bobot-bobot dari *neuron* akan disimpan di dalam sebuah *file* .txt. *Admin* akan meng-*upload file* ini ke dalam situs *web* agar dapat digunakan pada tahap pengenalan yang dilakukan oleh *user*. Saat *user* memasukkan gejala sebagai *input* di dalam sistem, maka *input* tersebut akan diterjemahkan ke dalam bilangan *biner* agar dapat dimasukkan ke dalam sebuah jaringan syaraf tiruan. Jaringan syaraf tiruan yang dibangun akan menggunakan bobot yang telah didapatkan dari *training application*. Setelah bobot-bobot dimasukkan maka proses dalam jaringan akan bergerak maju dan memberikan *output* kepada *user*.

Berikut adalah tahap-tahap perancangan jaringan syaraf tiruan yang digunakan dalam aplikasi ini.

3.2.1 Menentukan Jumlah Input dan Output

Dari hasil wawancara yang dilakukan dengan pakar *lower back pain* yang sesungguhnya, ditentukan jumlah *input* yang berupa gejala dimasukkan ke dalam

jaringan dan *output* yang berupa kemungkinan penyakit-penyakit yang diderita oleh pengguna aplikasi. Jumlah input dari aplikasi ini yaitu 33, sedangkan jumlah outputnya yaitu 11. Semua *input* dan *output* dimasukkan ke dalam jaringan dalam bentuk *binary*. Berdasarkan analisa ini, maka disediakan 33 neuron pada lapisan pertama (*input*) dan 11 neuron pada lapisan terakhir. Karena sistem memiliki kombinasi *input* dan *output* yang cukup banyak, maka jaringan tidak bisa menggunakan fungsi aktivasi yang sederhana (*threshold* dan indentitas). Oleh karena itu, pilihan fungsi aktivasi menjadi 2 macam yaitu *hyperbolic tangent* dan *sigmoid*. Dari percobaan yang dilakukan di dalam aplikasi, *sigmoid* dan *hyperbolic tangent* memiliki keluaran yang hampir sama. Namun, pada saat *training* fungsi *hyperbolic tangent* memerlukan waktu yang lebih singkat dibandingkan *sigmoid*. Melihat dari percobaan yang telah dilakukan, maka *hyperbolic tangent* dipilih sebagai fungsi aktivasi yang digunakan pada *Neural Network*.

3.2.2 Menentukan Jumlah Hidden Layer

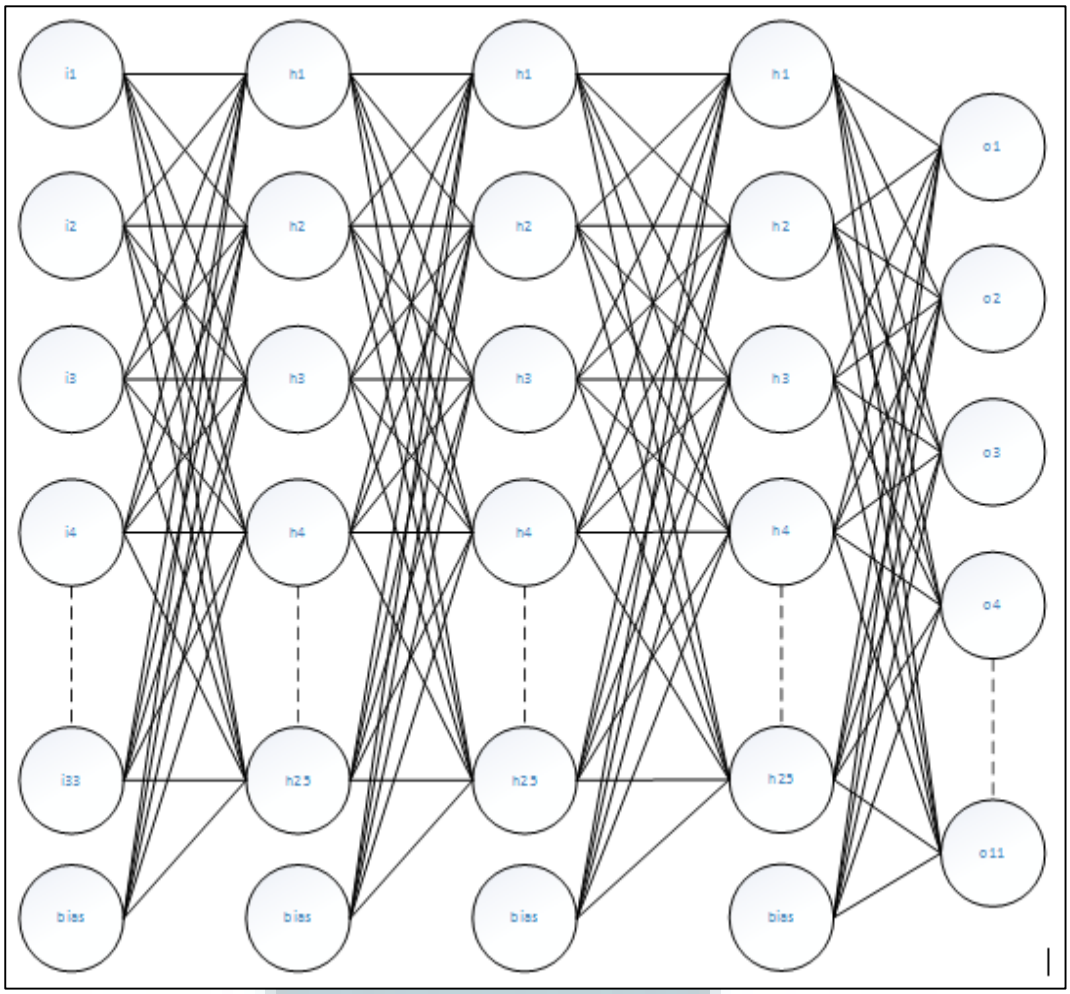
Berdasarkan teori pada halaman 20 menyatakan bahwa jaringan layar banyak dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan layar tunggal, meskipun kadangkala proses *training* lebih kompleks dan lama. Berdasarkan fakta diatas maka dilakukan percobaan-percobaan untuk menentukan berapa jumlah *hidden layer* dan *neuron* yang cukup baik untuk digunakan di dalam sistem pakar ini. Berikut adalah tabel dari hasil beberapa percobaan dengan membandingkan keakuratan data dan waktu proses perhitungan dari setiap percobaan. Akurasi dihitung dengan membandingkan hasil dari data training dan

hasil dari sistem yang telah melakukan *learning*. Jumlah data training yang berhasil didapatkan dari pakar yaitu sebanyak 151. Beberapa contoh perbandingan dalam menentukan layer akan diberikan pada tabel 3.1.

Tabel 3.1 Tabel Perbandingan untuk Menentukan Arsitektur Jaringan

No	N - hidden layer	Repetisi / Epoch	Akurasi	Waktu
1	0 (33,11)	10000	75.4 % (37 F)	33 s
2	1 (33, 25, 11)	10000	94.0 % (9F)	1m 41 s
3	1 (33, 50, 11)	10000	93.4 % (10 F)	3m 33s
4	2 (33, 50, 50, 11)	10000	91.4 % (13 F)	7 m 40s
5	2 (33, 25, 25, 11)	10000	94.7 % (8 F)	2 m 49 s
6	2 (33, 25, 25, 11)	20000	92.7 % (11 F)	5m 47 s
7	3 (33, 25,25,25,11)	10000	95.3 % (7 F)	4 m 2 s
8	3 (33, 25,25,25,11)	20000	96.0 % (6 F)	8 m 6 s
9	4 (33,25, 25,25,25,11)	10000	90.7 % (14 F)	5m 7 s

Dari beberapa percobaan yang dilakukan, ditetapkan 5 *layer* dengan kombinasi neuron 33, 25, 25, 25, 11. Untuk setiap layer pada jaringan syaraf tiruan ditambakan 1 neuron yang digunakan sebagai bias dengan nilai 1. Arsitektur dari jaringan yang digunakan dalam sistem pakar ini dapat dilihat pada gambar 3.1 dimana i adalah *input*, h adalah *hidden*, dan o adalah *output*.



Gambar 3.1 Perancangan Jaringan Syaraf Tiruan

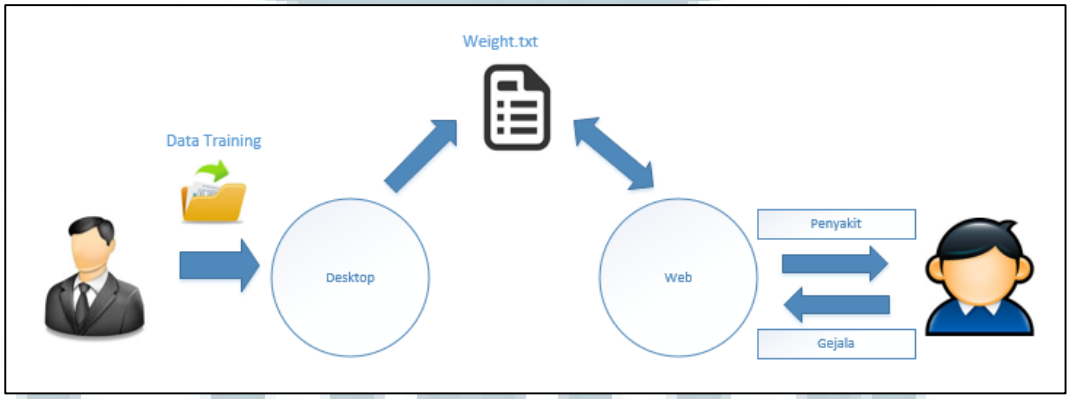
3.3 Perancangan Sistem

Setelah mendapatkan rancangan arsitektur jaringan dari aplikasi yang akan dibuat, selanjutnya dilakukan perancangan dari sistem secara keseluruhan. Diagram-diagram yang dibuat dalam perancangan sistem ini yaitu *use case diagram*, *activity diagram*, *class diagram* dan *sequence diagram*.

3.3.1 Gambaran Umum Sistem

Sistem pakar yang dibangun terdiri dari 2 buah aplikasi yaitu *training application* dan *web Application*. *Training application* digunakan karena untuk

melakukan training *Neural Network* membutuhkan *resources* yang cukup besar sehingga lebih baik dilakukan pada *desktop*. Sedangkan *web* digunakan agar *user* dapat mengakses aplikasi ini dimanapun dan kapanpun selama terdapat koneksi intenet. *Admin* akan memasukkan data *training* di *training application* dan memulai proses pelatihan jaringan. Selanjutnya bobot-bobot dari jaringan yang telah terlatih disimpan ke dalam file.txt yang diberi nama weight.txt. *File* ini kemudian di-*upload* ke dalam *web*. Hanya admin yang memiliki *priviledge* mengupload suatu *file*. Dengan menggunakan *file* weight.txt yang berisi bobot-bobot jaringan, *neural network* pada *web* membuat arsitektur jaringan yang sama seperti jaringan *neural* yang telah terlatih di *training application*. Pada saat *user* memasukkan *input* berupa gejala-gejala *Lower Back Pain* maka *input* akan dimasukkan ke dalam jaringan *neural* dan mendapatkan *output* berupa kemungkinan penyakit yang diderita oleh *user* tersebut. Diagram dari sistem pakar ini dapat dilihat pada gambar 3.2.



Gambar 3.2 Gambaran Umum Sistem

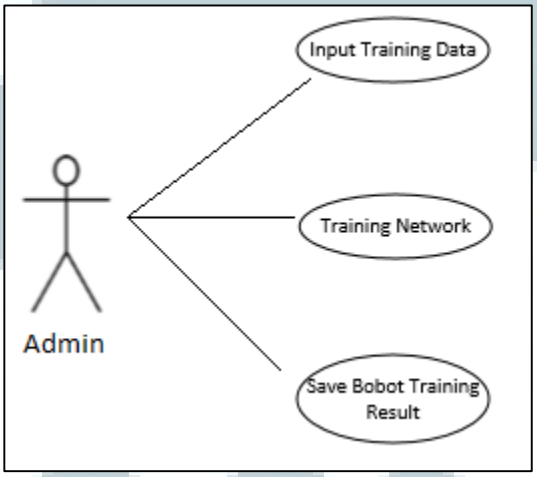
3.3.2 Use Case Diagram

Use Case Diagram adalah gambaran *graphical* dari beberapa atau semua aktor, use-case, dan interaksi di antara komponen-komponen tersebut yang

memperkenalkan suatu sistem yang akan dibangun. *Use case diagram* menjelaskan manfaat suatu sistem jika dilihat menurut pandangan orang yang berada di luar sistem. *Diagram use case* menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar. *Use Case Diagram* pada sistem ini dibagi menjadi 2 yaitu untuk *web* dan *training application.*

A. *Use Case Diagram Training application*

training application application hanya memiliki satu orang aktor yaitu *admin*. *Admin* dapat memilih file data training sebagai *input* untuk *training*, memulai *training* sistem dan menyimpan bobot hasil dari *training* yang dilakukan. *Use Case Diagram training application* dapat dilihat pada Gambar 3.3.

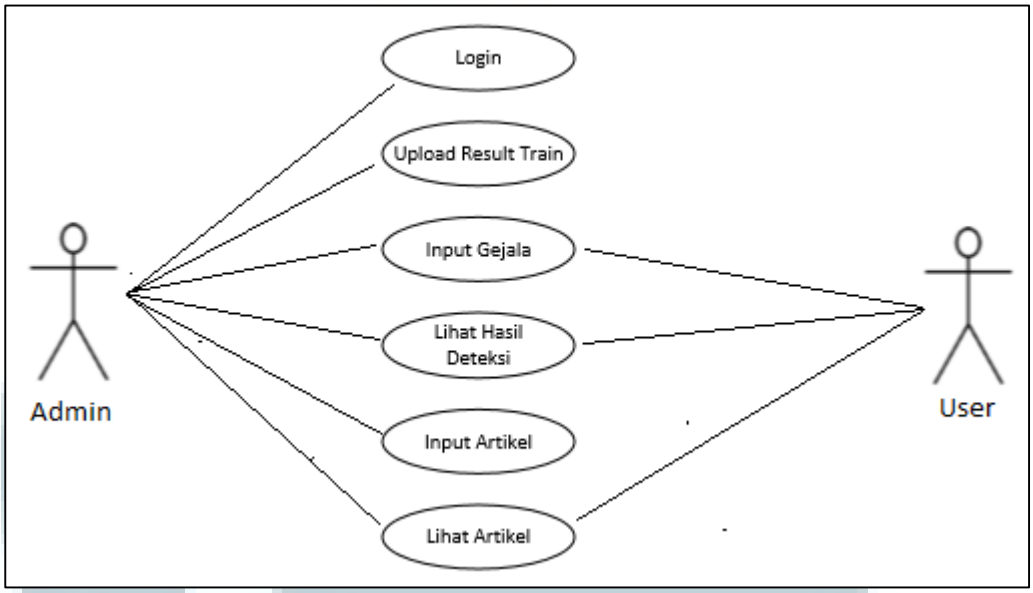


Gambar 3.3 *Use Case Diagram training application*

B. *Use Case Diagram Web Application*

Web Applcation pada sistem ini memiliki 2 orang aktor yaitu *admin* dan *user*. *Admin* bisa meng-*upload* file weight.txt yang merupakan hasil training dari *training application*. File ini yang akan digunakan untuk menginisialisasikan jaringan pada *web application*. Untuk fitur-fitur lain pada web seperti *input*

gejala, lihat hasil diagnosa, *input* artikel, dan lihat artikel dapat dilakukan oleh kedua aktor. *Use Case Diagram web application* dapat dilihat pada gambar 3.4.



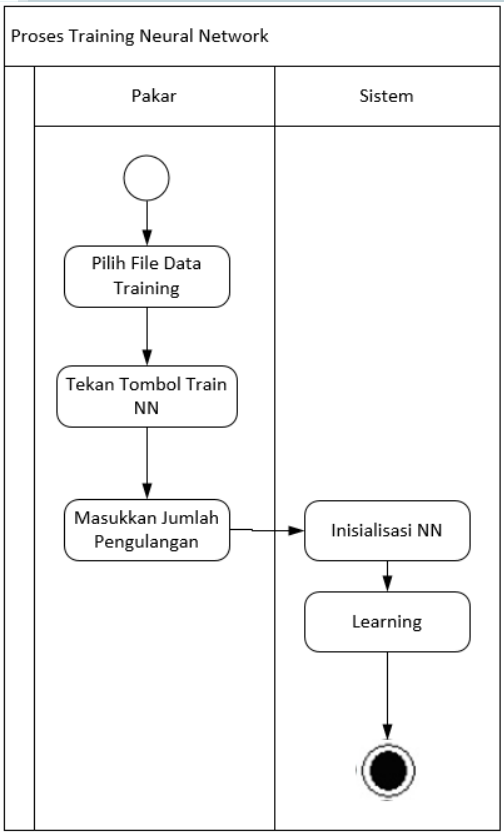
Gambar 3.4 *Use Case Diagram web application*

3.3.3 Activity Diagram

Diagram Aktivitas atau *Activity Diagram* adalah representasi grafis dari seluruh tahapan alur kerja. Diagram ini mengandung aktivitas, pilihan tindakan, perulangan dan hasil dari aktivitas tersebut. Pada pemodelan UML, diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen suatu sistem. *Activity diagram* pada aplikasi terdiri menjadi lima bagian, yaitu proses *training Neural Network*, proses penyimpanaan bobot, proses meng-*upload* bobot, proses melihat artikel dan proses memulai diagnosa. Proses *training Neural Network* dan penyimpanan bobot terjadi pada *training application*, sedangkan proses meng-*upload* bobot, melihat artikel, dan memulai deteksi dilakukan pada *web*.

A. Activity Diagram Proses Training Neural Network

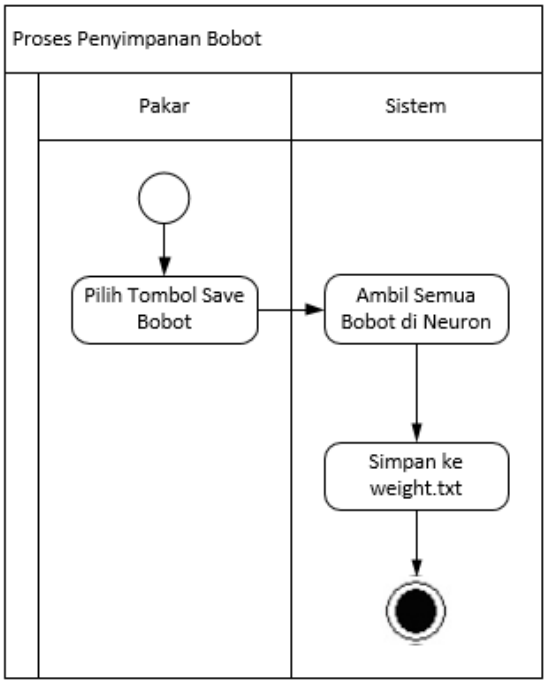
Proses ini dilakukan oleh *admin* di *training application* untuk mendapatkan sebuah jaringan terlatih dengan bobot-bobot yang tepat dalam setiap relasi antar neuron. *Admin* akan diminta untuk memilih file yang berisi data training dengan format [input(33)_(space)ouput(11)] dalam bentuk biner yang merupakan kombinasi antara gejala dan penyakit yang diderita. Selanjutnya, admin akan dimasukkan jumlah iterasi yang akan dilakukan terhadap data training tersebut agar mendapatkan jaringan NN dengan weight yang tepat. Ketika menekan tombol "Train NN" maka sistem akan menginisialisasi NN dan melakukan *learning* dari data *training* yang dipilih. *Activity diagram* dari proses *training Neural Network* dapat dilihat pada gambar 3.5.



Gambar 3.5 *Activity Diagram Training Neural Network*

B. Activity Diagram Proses Penyimpanan Bobot

Setelah sistem selesai melakukan *training Neural Network*, selanjutnya *Admin* dapat menekan tombol "Save Bobot". Ketika tombol tersebut ditekan maka sistem akan mengambil semua bobot-bobot koneksi antar network dan menyimpannya ke dalam file weight.txt. File ini akan digunakan untuk menginisialisasi jaringan syaraf tiruan yang berada pada website. Dengan menggunakan arsitektur jaringan dan bobot yang sama, tentunya akan membuat web menghasilkan *output* yang sama seperti *training application*. *Activity Diagram* dari proses penyimpanan bobot dapat dilihat pada gambar 3.6.

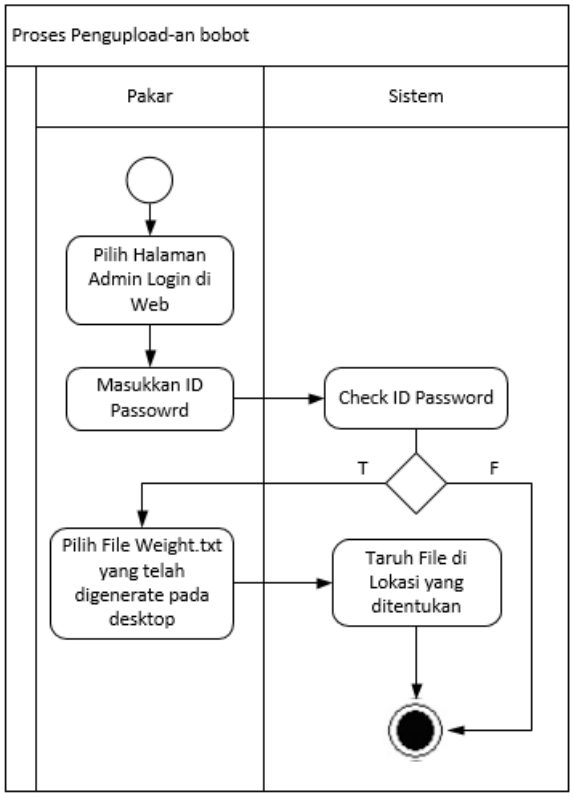


Gambar 3.6 *Activity Diagram* proses penyimpanan bobot

C. Activity Diagram Proses Upload Bobot

Proses ini digunakan untuk *upload* weight.txt setelah sistem melakukan *training*. *Upload file* hanya dapat dilakukan oleh seorang admin, oleh karena itu diperlukan validasi dengan menggunakan *username* dan *password* pada halaman

login admin. Setelah *login* berhasil maka *admin* diminta untuk memilih file *weight.txt* dari komputernya. Setelah disubmit maka sistem akan menimpa file *weight.txt* yang lama dengan file *weight.txt* yang baru di lokasi yang telah ditentukan pada sistem. *Activity diagram* dari proses meng-*upload* bobot dapat dilihat pada gambar 3.7.

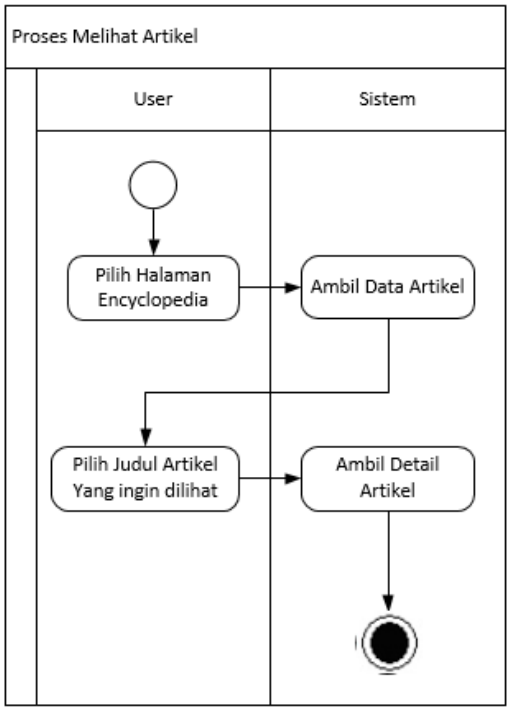


Gambar 3.7 *Activity Diagram* proses *Upload* bobot

D. Activity Diagram Proses Melihat Artikel

Halaman ini digunakan oleh *user* untuk melihat artikel-artikel yang berada pada website. Halaman artikel ini dapat diakses dengan menekan tab "Enclyclopedia" pada *menu* navigasi. Ketika menu ini ditekan maka sistem akan mengambil artikel-artikel dari database dan menampilkan judul-judul artikel beserta *summary*-nya dalam bentuk *list*. Jika *user* ingin melihat *detail* dari suatu

artikel, *user* dapat mengklik judul dari artikel tesebut. Ketika user mengklik judul artikel tersebut user akan dipindahkan ke halaman "detailencyclopedia.php". Sistem akan kembali mengambil data-data dari database dan menampilkan informasi detail dari artikel yang dipilih. *Activity diagram* dari proses melihat artikel dapat dilihat pada gambar 3.8.

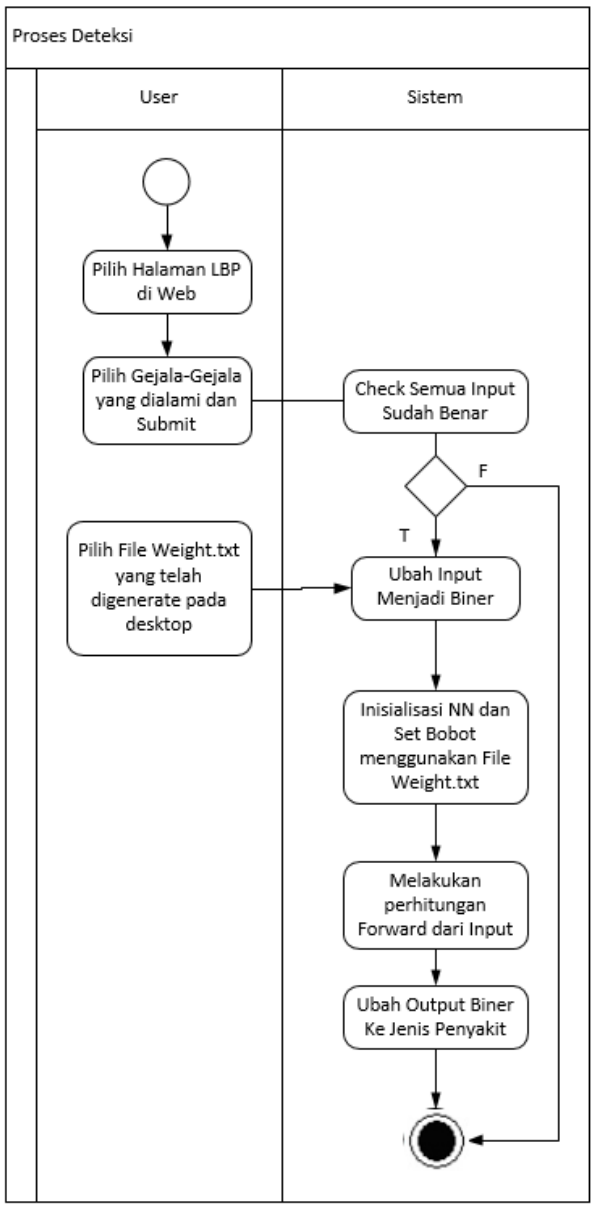


Gambar 3.8 *Activity Diagram* proses melihat artikel

E. Activity Diagram Proses Deteksi

Proses ini digunakan oleh *user* untuk mendapatkan hasil diagnosa dari penyakit *Lower Back Pain* yang mungkin dideritanya. Halaman ini dapat diakses dengan mengklik menu "LBP Detection" pada *menu* navigasi. Selanjutnya *user* akan diminta memasukkan jawaban dari gejala-gejala yang terdapat pada *table form* dan menekan tombol *Submit*. Ketika tombol *submit* di tekan, sistem akan memeriksa semua *input* yang dimasukkan dan meminta *user* memasukkan ulang jika terdapat data yang belum diisi. Jika *input* sudah terisi semua, maka sistem akan

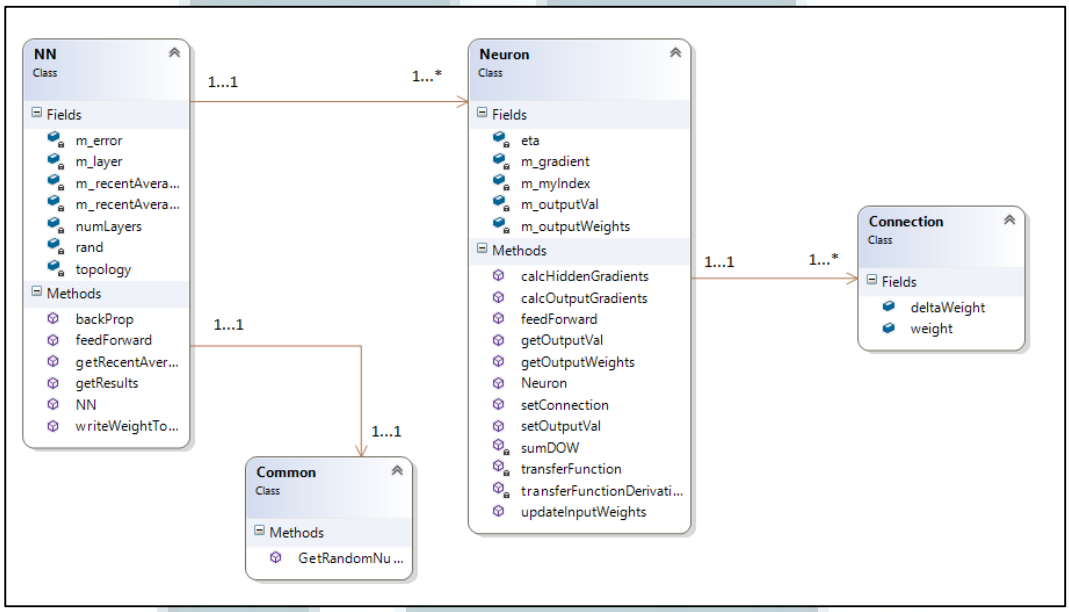
merubah *input* tersebut kedalam format *binary* dan menginisialisasikan jaringan syaraf tiruan menggunakan file *weight.txt*. *Input* akan dimasukkan ke dalam jaringan syaraf tiruan dan sistem akan melakukan perhitungan. Setelah mendapatkan *output* dari jaringan syaraf tiruan, *output* akan diubah menjadi jenis-jenis penyakit dan ditampilkan kepada *user* dalam bentuk *dialog box*. *Activity Diagram* dari proses deteksi dapat dilihat pada gambar 3.9.



Gambar 3.9 *Activity Diagram* proses deteksi

3.3.4 Class Diagram

Class Diagram atau Diagram Kelas adalah diagram UML yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan method atau operasi.

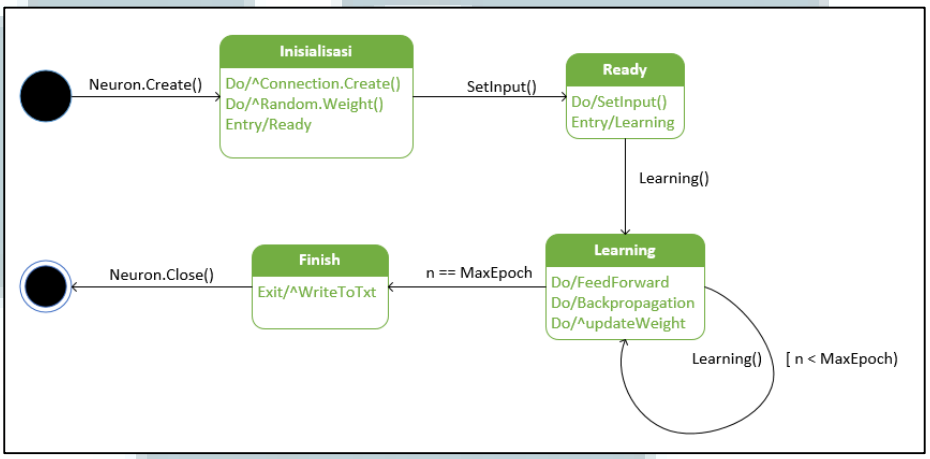


Gambar 3.10 *Class Diagram*

Pada gambar 3.10 merupakan *class diagram* dari jaringan syaraf tiruan yang digunakan pada sistem pakar di *training application*. Jaringan syaraf tiruan akan menjadi sebuah objek yaitu NN dan memiliki beberapa objek Neuron. Objek NN juga berelasi dengan Objek Common untuk menggunakan *method* random. Objek Neuron juga memiliki beberapa objek dari Connection yang berguna untuk menyimpan *weight* setiap koneksi antar Neuron. Pada *web application*, *class diagram* yang digunakan hampir sama dengan *training application*. Perbedaannya pada *web application*, tidak membutuhkan *function-function* yang mendukung backpropagation karena bobot telah didapat dari weight.txt.

3.3.5 State Chart Diagram

Dari *class diagram* yang telah digambarkan pada gambar 3.10, terdapat 2 *class* dinamis yaitu *class* neuron dan *class connection*. Akan tetapi *class connection* di dalam aplikasi ini bersifat seperti *struct* dimana *class* tersebut hanya memiliki *attribute* dan tidak memiliki *method*. Oleh karena itu, hanya *class* neuron yang memiliki *state chart diagram*. *State chart diagram* untuk *class* neuron dapat dilihat pada gambar 3.11.



Gambar 3.11 *State Chart Diagram Class Neuron*

3.3.6 Sequence Diagram

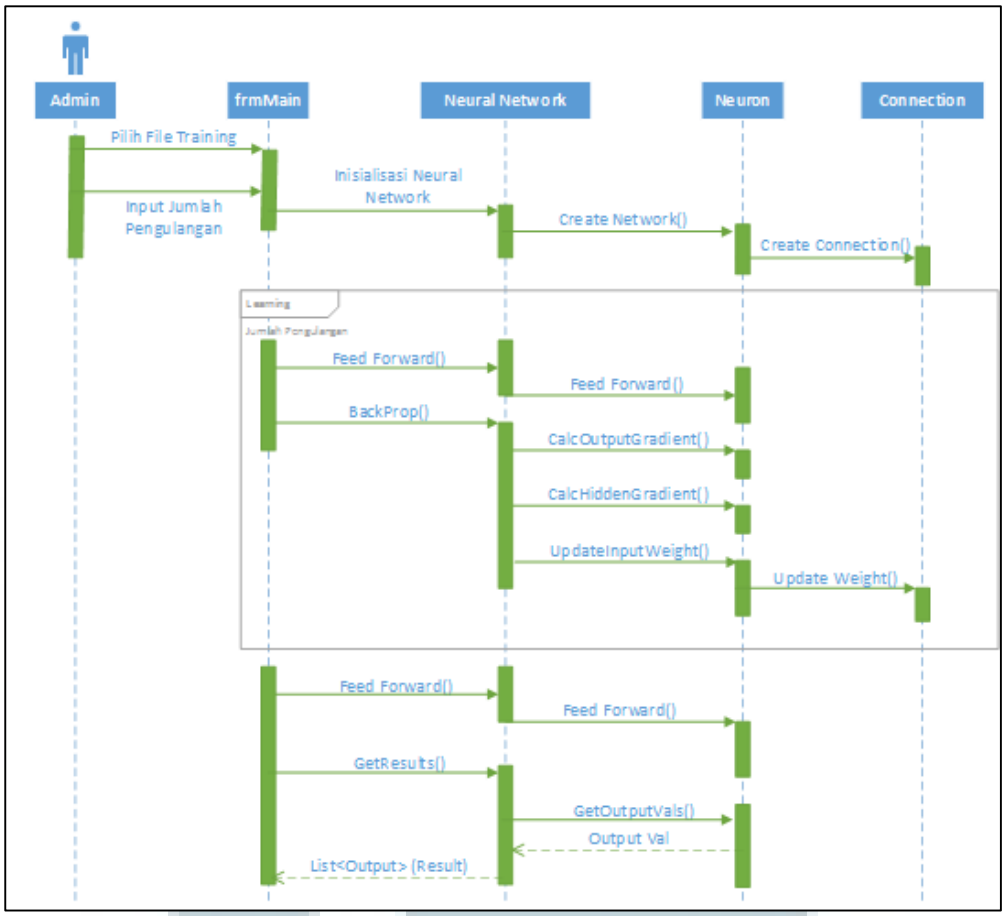
Sequence Diagram adalah suatu *diagram* yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi di antara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. *Sequence Diagram* pada sistem ini dibagi menjadi 2 yaitu untuk *web* dan *training application*.

A. *Sequence Diagram* pada *Training Application*

Sequence diagram pada *training application* terdiri dari 2 *diagram* yaitu *sequence diagram training Neural Network* dan *sequence diagram* penyimpanan bobot hasil *training*.

1. *Sequence Diagram* T*raining Neural Network.*

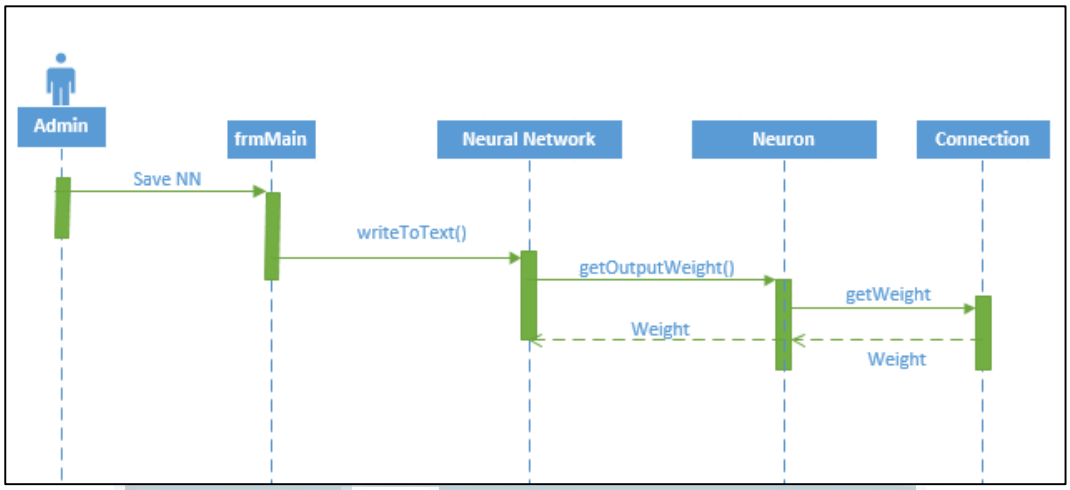
Proses dimulai ketika admin ingin melakukan *training* jaringan. Tahap pertama admin harus memilih file training yang akan digunakan, memasukkan jumlah pengulangan yang diinginkan dan menekan tombol "Train NN". frmMain akan membuat suatu jaringan syaraf tiruan yang terdiri dari beberapa layer dengan jumlah neuron-neuron yang telah ditetapkan yaitu 33,25,25,25,11. Pada tahap inisialisasi awal, semua nilai *weight* pada koneksi antar neuron diisi secara random. Setelah diinisialisasi frmMain akan memanggil fungsi FeedForward() pada NN dengan memasukkan *input* dari file training untuk mendapatkan *output* dari jaringan. *Ouput* jaringan akan dicocokan dengan *output* yang diinginkan dengan menggunakan *backpropagation.* Algoritma *backpropagation* akan membuat *output* dari jaringan syaraf tiruan semakin mendekati *output* yang diinginkan dalam setiap pengulangan. *Sequence diagram* proses *Training Neural Network* dapat dilihat pada gambar 3.12.



Gambar 3.12 *Sequence Diagram Training Neural Network*

2. *Sequence Diagram* Penyimpanan Bobot Hasil *Training*.

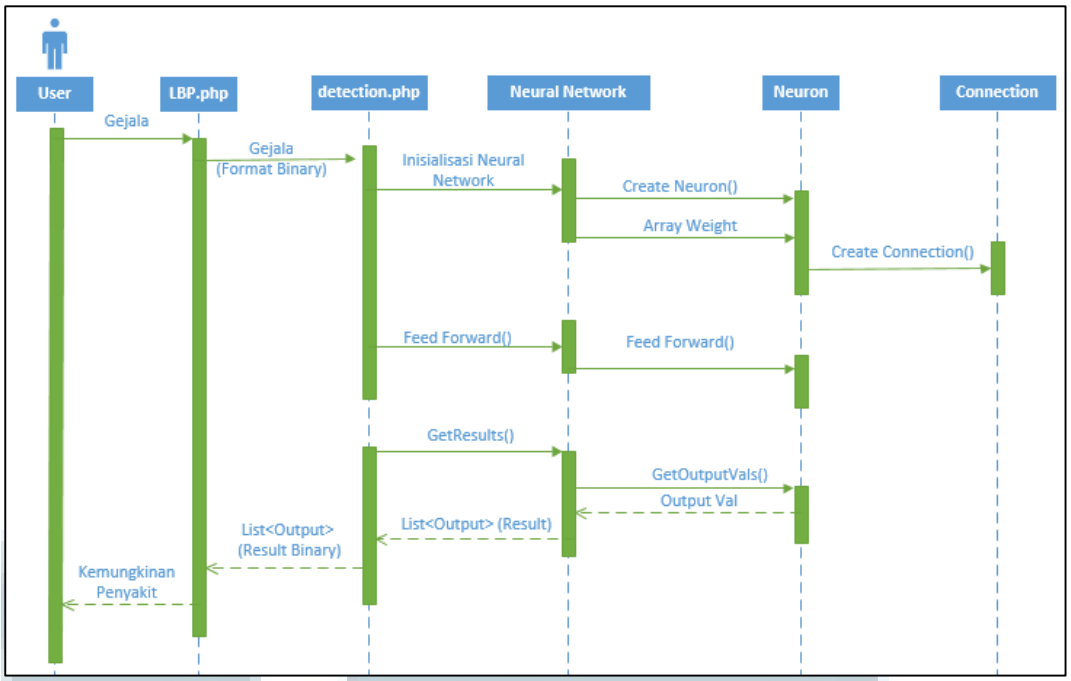
Proses dimulai ketika admin ingin menyimpan bobot hasil *training* neural network. Admin akan menekan tombol "Save NN" yang akan memerintahkan frmMain memanggil *function* writeToText() pada NN. Selanjutnya NN akan melakukan pengulangan sebanyak jumlah *neuron* dan menyimpan bobot-bobot setiap koneksi dari *neuron* ke dalam .txt file yang akan diberi nama weight.txt. *Sequence Diagram* untuk proses penyimpanan bobot jaringan dapat dilihat pada gambar 3.13.



Gambar 3.13 *Sequence Diagram* Penyimpanan Bobot Hasil *Training*

B. *Sequence Diagram* pada *Web Application* (Proses Deteksi)

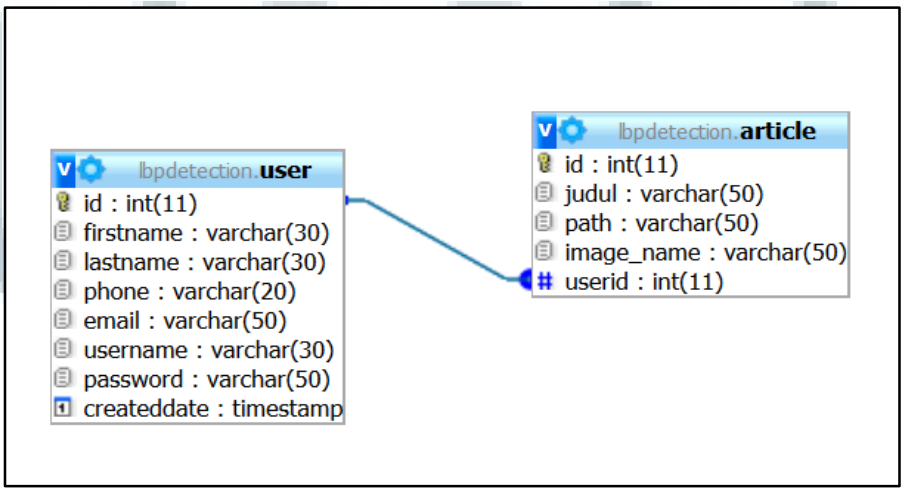
Sequence Diagram pada *Web Application* hanya terdiri dari 1 *Diagram* yaitu *Sequence Diagram* Proses Deteksi. Proses ini dimulai ketika *user* memasukkan gejala-gejala penyakit pada halaman LBP.php. Gejala-gejala tersebut diubah dalam bentuk binary dan dikirim ke halaman detection.php. Pada halaman detection.php, *Neural Network* diinisialisasi. Bobot-bobot pada jaringan di-*set* menggunakan data dari weight.txt. Setelah jaringan selesai diinisialisasi, maka input-input yang berupa binary dimasukkan ke dalam jaringan dan tahapan maju (*feedForward*) dilakukan untuk mendapatkan *output* dari *Neural Network*. Selanjutnya *ouput* yang berupa *binary* diterjemahkan menjadi jenis-jenis penyakit dan ditampilkan kepada *user*. Untuk penjelasan lebih lanjut, *Sequence Diagram* untuk proses deteksi pada *web* ini dapat dilihat pada gambar 3.14. Untuk aktivitas melihat artikel dan *upload* bobot tidak mempunyai *sequence diagram* karena aktivitas tersebut bukanlah berbasis objek.



Gambar 3.14 *Sequence Diagram* Proses Deteksi pada *web*

3.4 Perancangan Database

Database yang digunakan oleh sistem adalah MySQL. Untuk menjelaskan hubungan antar objek data dalam database maka digambarkan *Entity Relationship Diagram* (ERD). Terdapat dua buah entitas yakni User dan Article. Hubungan antar entitas dan atributnya digambarkan pada Gambar 3.15.



Gambar 3.15 *Entity Relationship Diagram*

1. Nama Tabel : User

   Fungsi : menyimpan data-data *user* yang menggunakan sistem.

   Primary Key : ID

   Foreign Key : -

Tabel 3.2 Tabel User

No	Nama Field	Tipe Data	Panjang	Keterangan
1	ID	int	11	ID pengguna
2	firstname	varchar	30	Nama depan
3	lastname	varchar	30	Nama belakang
4	phone	varchar	20	No telp pengguna
5	email	varchar	50	Alamat email pengguna
6	username	varchar	30	Username pengguna
7	passsword	varchar	50	Password pengguna
8	createddate	timestamp		Waktu ID dibuat

2. Nama Table : Article

   Fungsi : menyimpan data-data article di dalam *web*

   Primary Key : ID

   Foreign Key : UserID (ID)

Table 3.3 Table Article

No	Nama Field	Tipe Data	Panjang	Keterangan
1	ID	int	11	ID pengguna
2	judul	varchar	50	Judul Artikel
3	path	varchar	50	Lokasi artikel disimpan
4	image_name	varchar	50	Nama file gambar artikel
5	userid	varchar	11	ID pengguna yang mengunggah artikel

3.5 Perancangan Antarmuka

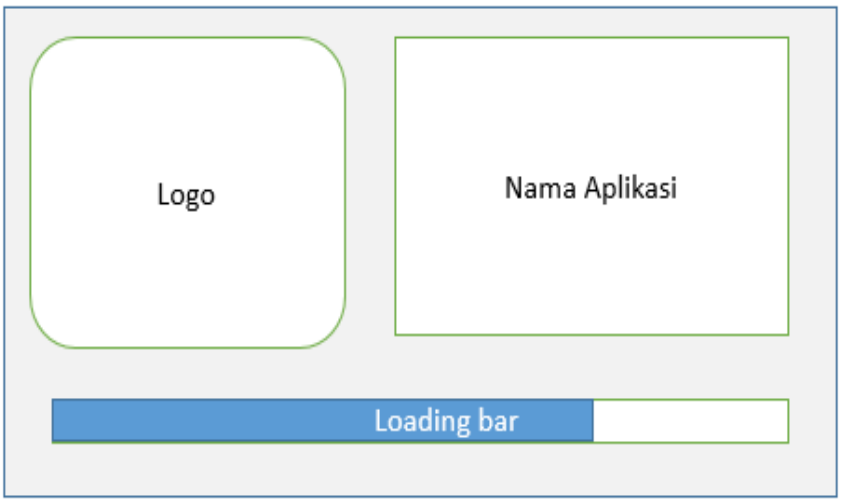
Perancangan antarmuka pada sistem ini dibagi menjadi 2 yaitu perancangan antarmuka *training application* dan perancangan antarmuka *web application*.

3.5.1 Perancangan Antarmuka Training Application

Tampilan antarmuka *training application* dibagi menjadi 2 bagian yang akan dijelaskan sebagai berikut.

1. Halaman Splash screen

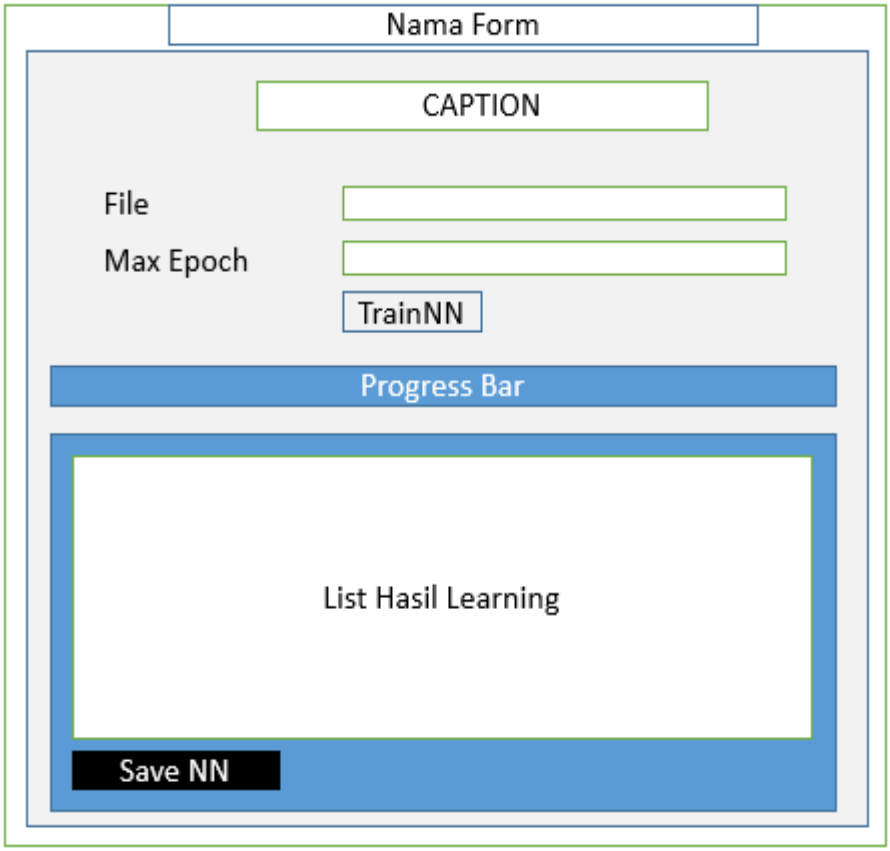
Halaman ini berisi logo dan nama dari aplikasi yang dibangun. Halaman ini dijalankan pada saat pertama kali aplikasi dibuka. Rancangan tampilan dari splash screen dapat dilihat pada gambar 3.16.



Gambar 3.16 Rancangan Halaman *splash screen.*

2. Halaman Utama

Halaman utama merupakan halaman pertama setelah *splash screen*. Pada halaman ini berisi fitur *training network* yang dapat diakses oleh admin. Rancangan tampilan dari halaman utama dapat dilihat pada gambar 3.17.



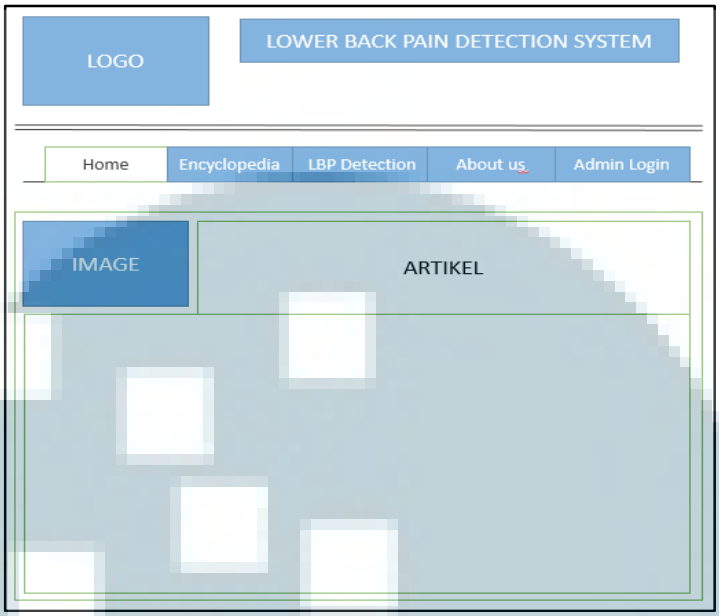
Gambar 3.17 Rancangan Halaman Utama *training application*

3.5.2 Perancangan Antarmuka Web

Tampilan antarmuka web dibagi menjadi 7 bagian yang akan dijelaskan sebagai berikut.

1. Halaman Utama

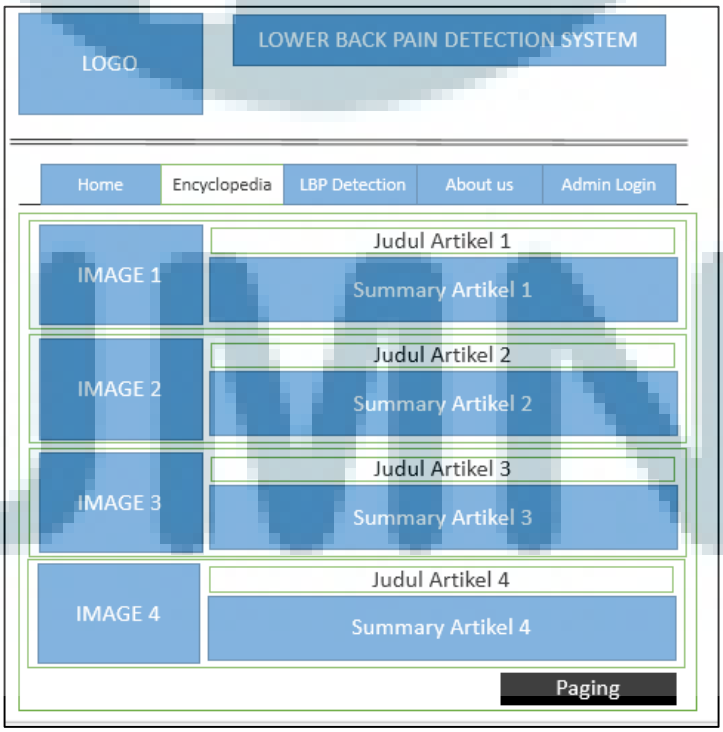
Halaman ini merupakan halaman pertama yang tampil ketika *web* diakses. Halaman ini berisi sekilas pengenalan tentang penyakit *Lower Back Pain*. Rancangan tampilan dari halaman utama *web* dapat dilihat pada gambar 3.18.



Gambar 3.18 Rancangan Halaman Utama *web*

2. Halaman Encyclopedia

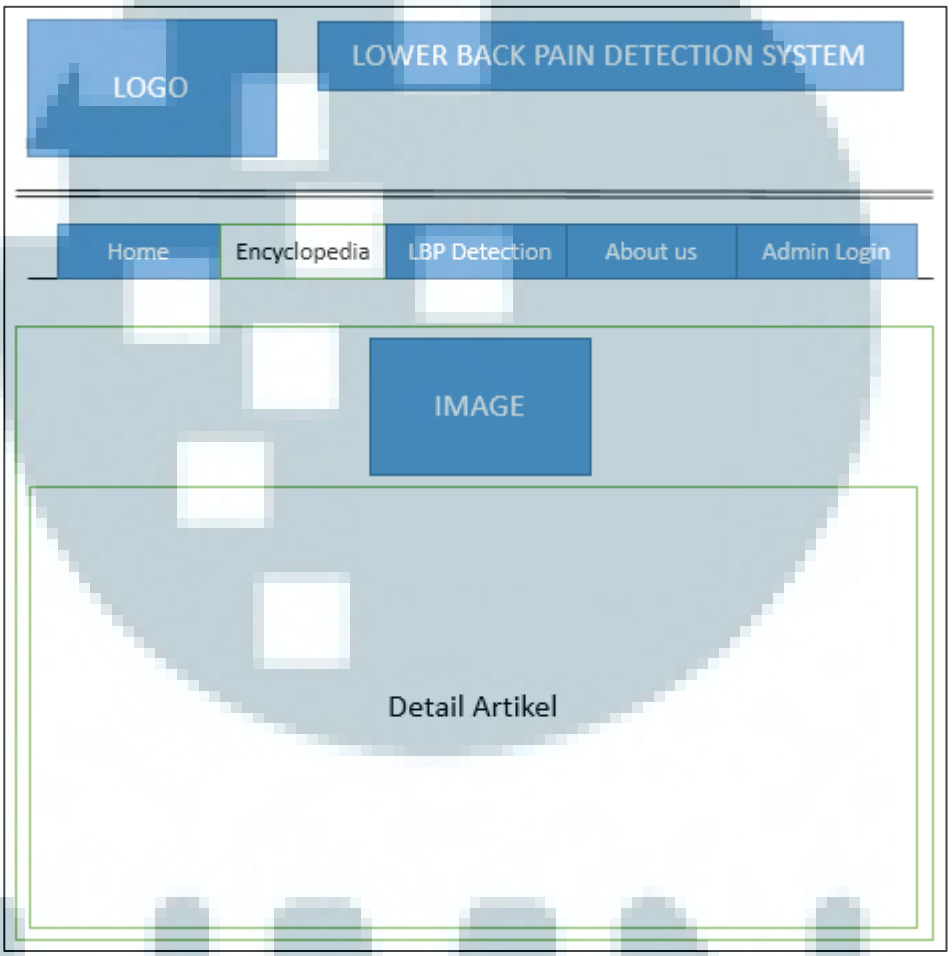
Halaman ini berisi list-list artikel yang berupa judul dan uraian singkat mengenai artikel tersebut. Rancangan halaman ini dapat dilihat pada gambar 3.19.



Gambar 3.19 Rancangan Halaman *encyclopedia*

3. Halaman Detail Encyclopedia

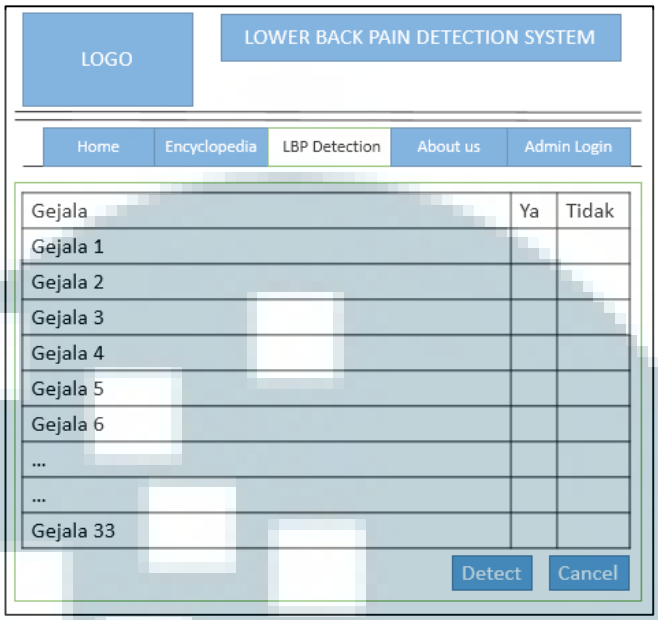
Halaman ini ditampilkan ketika *user* mengklik salah satu dari list *artikel* pada halaman "encyclopedia.php". Halaman ini berisi informasi detail dari artikel yang diklik. Rancangan halaman ini dapat dilihat pada gambar 3.20.



Gambar 3.20 Rancangan Halaman *detail encyclopedia*

4. Halaman LBP Detection

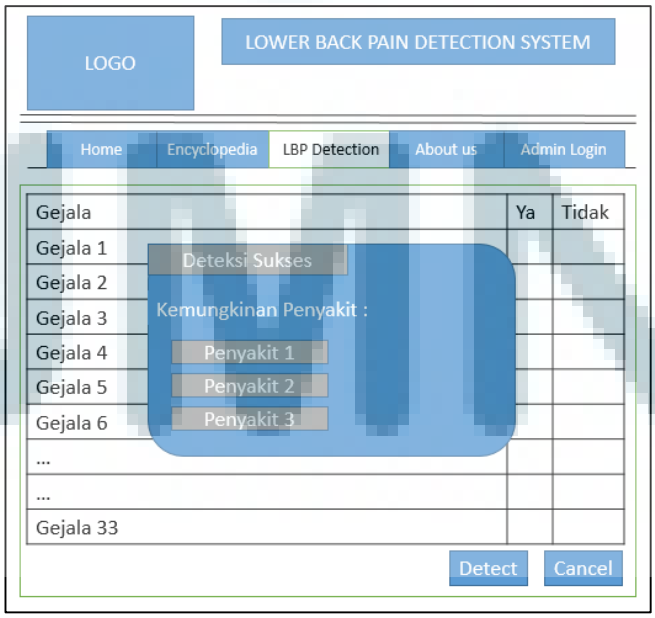
Halaman ini digunakan untuk memberikan diagnosa penyakit yang diderita seorang *user*. *User* akan diminta untuk menjawab semua gejala-gejala penyakit pada halaman ini dan menekan tombol *Detect*. Rancangan dari halaman ini dapat dilihat pada gambar 3.21.



Gambar 3.21 Rancangan Halaman LBP *detection*

5. Halaman Hasil Deteksi

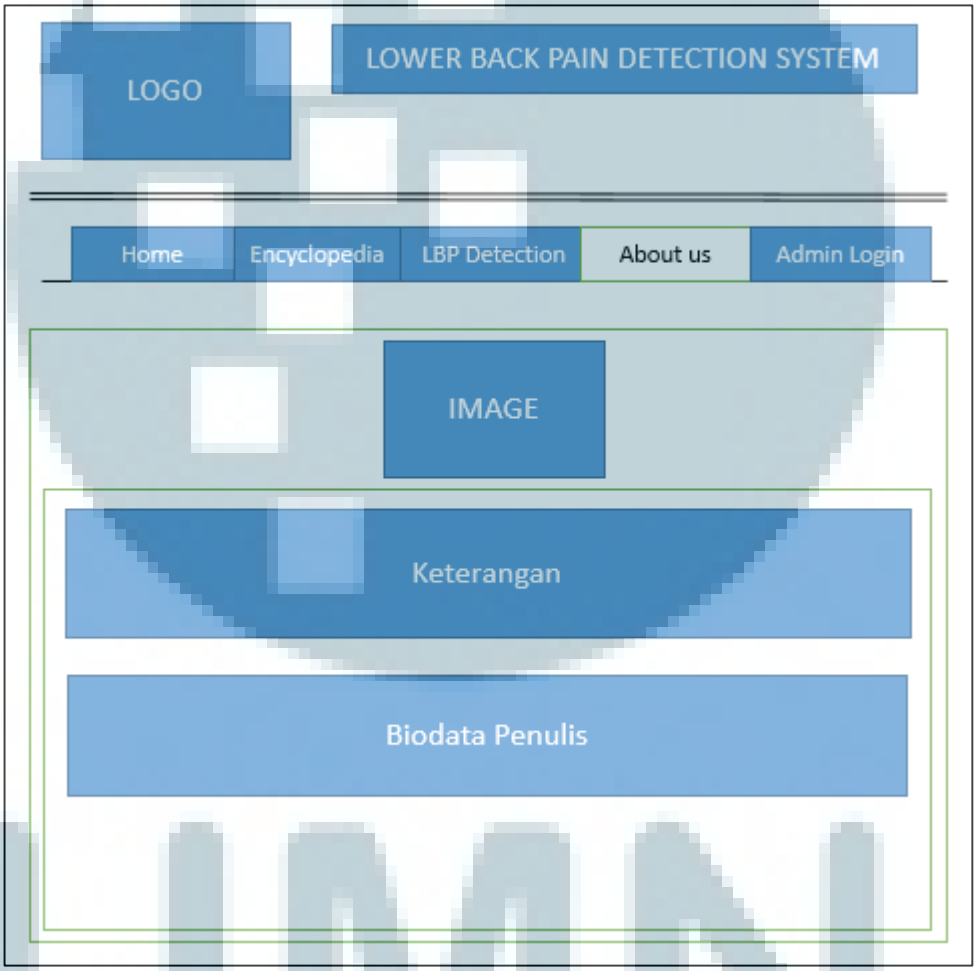
Halaman hasil deteksi keluar ketika *user* menekan tombol Detect pada halaman LBP Detection. Halaman ini berisi kemungkinan-kemungkinan penyakit yang diderita oleh pengguna dalam bentuk dialog box. Rancangan halaman ini dapat dilihat pada gambar 3.22.



Gambar 3.22 Rancangan Halaman Hasil Deteksi

6. Halaman About Us

Halaman ini berisi informasi-informasi mengenai penulis dan alasan mengapa software ini dibuat. Pada halaman ini terdapat keterangan dan data-data dari penulis jika dibutuhkan pada penelitian selanjutnya. Rancangan dari halaman ini dapat dilihat pada gambar 3.23.

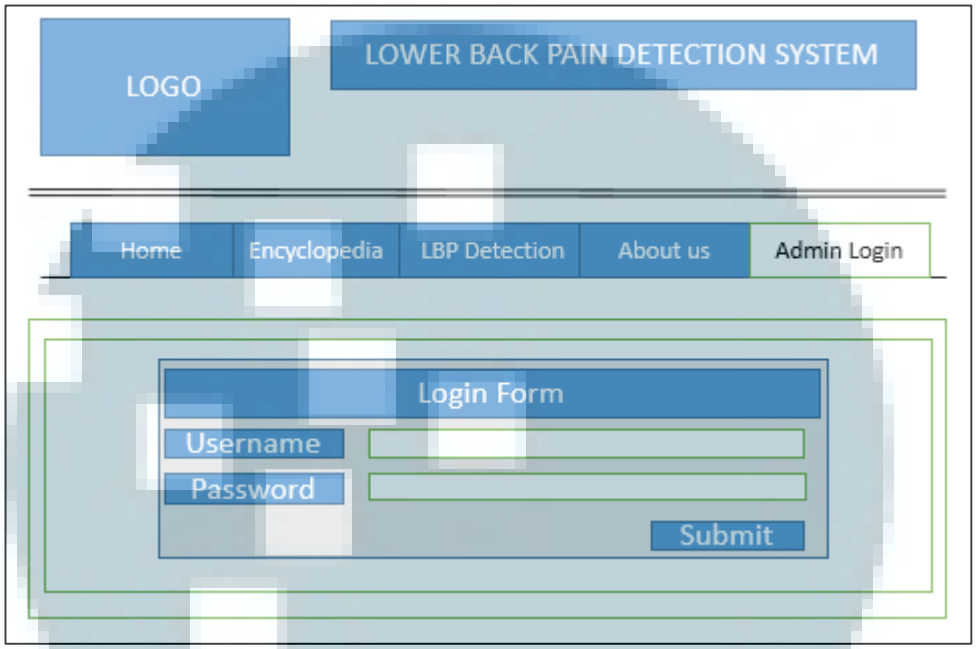


Gambar 3.23 Rancangan Halaman *About Us*

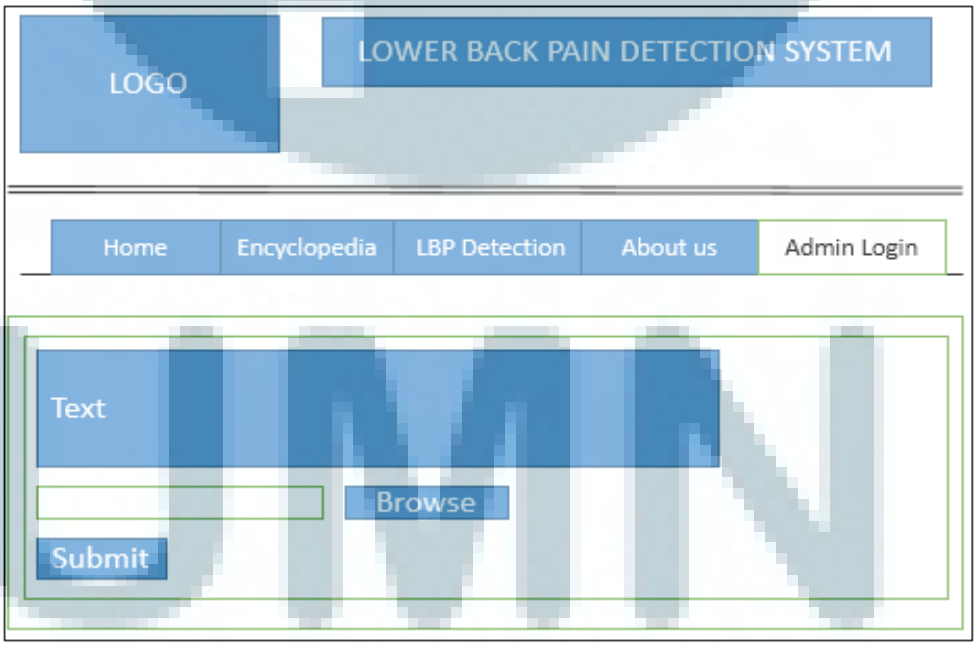
7. Halaman Admin

Halaman ini digunakan untuk memverifikasi seorang *admin* dan tempat dimana file weight.txt dapat diupload. Rancangan tampilan form *username* dan *password*

dapat dilihat pada gambar 3.24. Rancangan tampilan halaman tempat admin dapat meng-*upload* file weight.txt dapat dilihat pada gambar 3.25.



Gambar 3.24 Rancangan *Form Login*



Gambar 3.25 Rancangan *Upload weight*.txt