



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.

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi

Metode penelitian yang akan digunakan dalam penelitian ini antara lain adalah sebagai berikut:

1. Studi Literatur

Tahap awal dilakukan studi mengenai referensi-referensi yang berhubungan dengan pokok bahasan penelitian, seperti algoritma jaringan syaraf tiruan, pembuatan *web service*, penggunaan Android untuk mengambil gambar, dan pengolahan citra. Referensi-referensi tersebut dapat berupa jurnal, artikel, buku, dan lain-lain.

2. Perancangan Jaringan Syaraf Tiruan

Pada perancangan jaringan syaraf tiruan akan dirancang bagaimana bentuk dari jaringan syaraf tiruan yang akan digunakan untuk pelatihan dan pengenalan bunga. Hal seperti jumlah masukan, bagaimana data akan diolah dari gambar menjadi *input* jaringan syaraf tiruan, penggunaan *hidden layer*, jumlah *hidden layer*, jumlah *node* pada *hidden layer*, serta bagaimana membuat jumlah *output* menjadi dinamis sesuai data keluaran yang ada. Selain itu juga dirancang bagaimana cara untuk mendapatkan akurasi yang maksimal.

3. Perancangan dan Pembangunan Aplikasi

Perancangan dan pembangunan aplikasi ini dibagi menjadi dua tahapan. Pertama, melakukan perancangan awal terhadap aplikasi *server* yang meliputi perancangan alur aplikasi dalam bentuk *flowchart* dan perancangan awal terhadap

aplikasi *client* dalam bentuk *flowchart*. Selain itu juga dibuat *use case diagram*, *class diagram*, dan *sequence diagram*. Terakhir, melakukan perancangan *user interface* untuk *server* dan *client* serta pembuatan aplikasi. Aplikasi *server* dibangun menggunakan C# dengan WCF serta .NET Framework versi 4.5.1. Aplikasi *client* dibangun menggunakan Android dengan dasar bahasa pemrograman Java.

4. Uji Coba dan Evaluasi

Melakukan uji coba terhadap aplikasi disertai dengan evaluasi hasil yang didapatkan. Ditahap ini juga mencari jaringan syaraf tiruan yang sudah optimal untuk digunakan. Selain itu uji coba ini bertujuan untuk mencari *bug* yang masih ada pada aplikasi tersebut dan memperbaiki aplikasi ini.

5. Penulisan Skripsi

Pada tahap ini, akan dilakukan penyusunan laporan yang memuat dokumentasi mengenai perancangan, pembuatan, serta hasil dari aplikasi yang telah dibuat dalam suatu karya ilmiah

3.2 Perancangan Jaringan Syaraf Tiruan

Perancangan jaringan syaraf tiruan yang akan dilakukan mempunyai dua tahap, yaitu tahap pelatihan dan tahap pengenalan. Pada tahap pelatihan jaringan syaraf tiruan, akan digunakan algoritma *Backpropagation* untuk mencari bobot yang sesuai untuk jaringan syaraf tiruan tersebut. Pada nantinya setelah jaringan syaraf tiruan selesai proses pelatihan, jaringan syaraf tiruan akan disimpan ke dalam *file*. Jaringan yang dilatih akan mengubah bobotnya dengan kecepatan sesuai *learning rate* yang ditentukan dan *momentum rate* yang ditentukan agar mencapai suatu

kriteria dimana jaringan tersebut dapat mengenali input data sesuai yang diharapkan.

Tahap selanjutnya adalah tahap pengenalan, dimana di tahap ini aliran pemrosesan data jaringan hanya bergerak maju (*forwardpropagation*). Jaringan syaraf tiruan yang sudah disimpan ke dalam *file* akan dibuka kembali dan akan digunakan bobotnya untuk mengenali data *input*.

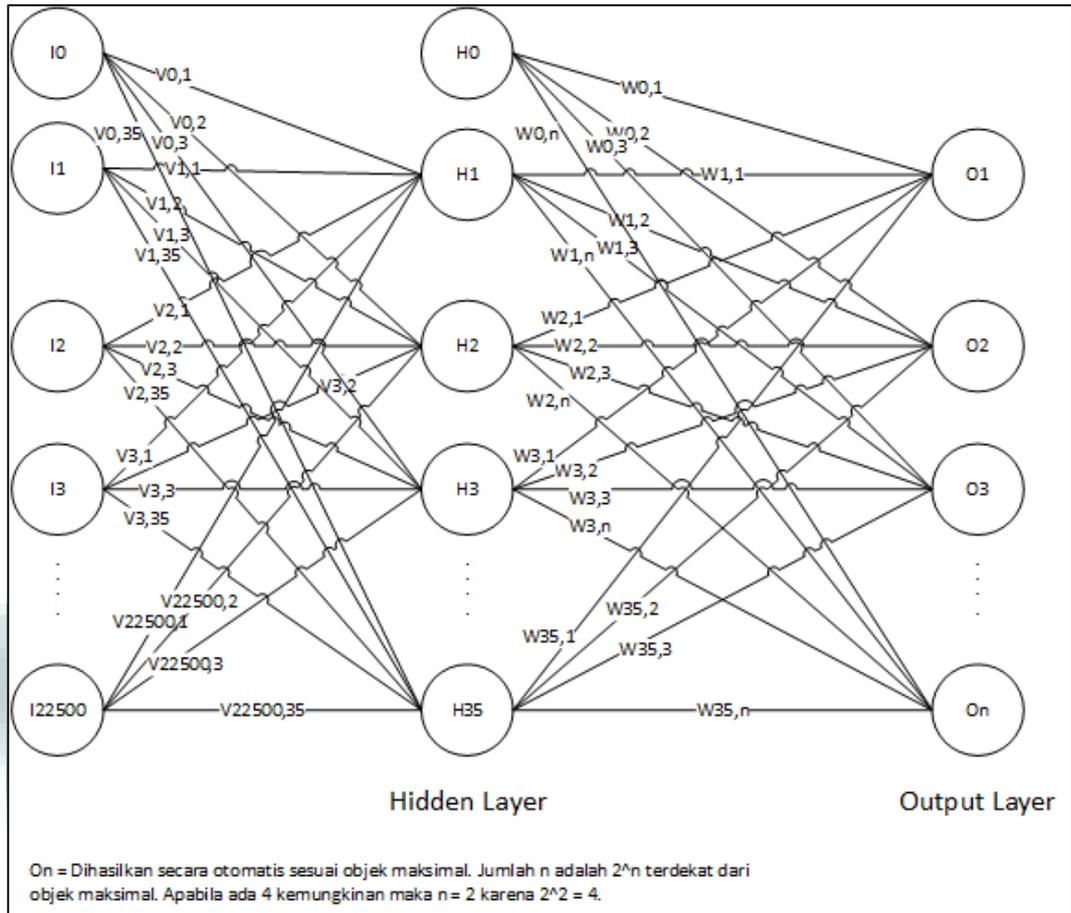
Kedua tahap di atas akan membutuhkan data dari gambar input yang sudah diproses terlebih dahulu. Gambar akan diproses dan diekstrak fiturnya dengan deteksi tepi sobel. Setelah itu akan *resize* dan akan diperjelas dengan *threshold*. Setelah itu setiap *pixel* akan diubah ke barisan *double* yang berupa *double* berkisar dari [0,1] yang didapat dari hasil rata-rata R,G,B setiap *pixel*. Setelah barisan input sudah didapat baru dimasukkan ke dalam jaringan syaraf tiruan. Jumlah *output* yang ada pada jaringan akan dihasilkan secara otomatis dari jumlah jenis bunga yang ada pada *database*. Dari nomor *index database* akan diubah menjadi biner lalu dijadikan barisan target *output*.

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

1. Jaringan syaraf tiruan akan terdiri dari tiga buah *layer* seperti yang dijelaskan pada gambar 3.1.
2. Jumlah masukan pada *input layer* pada aplikasi ini ditetapkan untuk mengenali gambar yang berresolusi 150x150 *pixel* sehingga jumlah masukan yang diperlukan adalah 22500.

3. *Hidden layer* yang dicoba berjumlah satu *layer* dan jumlah *neuron* pada *hidden layer* akan ditentukan setelah proses pencarian arsitektur jaringan optimum.
4. Antara *input layer* dan *hidden layer* akan digunakan optimisasi Nguyen dan Widrow(1990).
5. Digunakan *momentum* agar mempercepat proses pelatihan. *Momentum* adalah penambahan bobot dari perubahan bobot selanjutnya.
6. Bobot dasar awalnya akan dipilih secara acak antara -0.5 dan 0.5.
7. Jumlah keluaran akan dihasilkan secara otomatis oleh program untuk saat pelatihan sesuai data yang ada.
8. Data pelatihan dan data *testing* berbanding 2 : 1 dimana yang akan digunakan adalah data pelatihan sejumlah 61 foto dan data *testing* 30 foto.

UMMN



Gambar 3.1 Perancangan Jaringan Syaraf Tiruan

3.3 Perancangan Sistem

3.3.1 Diagram Alir

Diagram Alir merupakan bagan-bagan yang memiliki arus yang menggambarkan langkah-langkah dan proses dari suatu sistem. Oleh karena sistem yang dibuat berupa *web service* yang akan digunakan oleh *client*, maka diagram alir dapat dibagi menjadi 14 diagram alir yang sesuai dengan jumlah hal yang bisa dilakukan oleh *client*. Diagram tersebut yaitu diagram alir sistem, diagram alir penambahan gambar, diagram alir melihat gambar, diagram alir penghapusan gambar, diagram alir proses pembuatan jaringan syaraf tiruan baru, diagram alir proses *reset* jaringan syaraf tiruan, diagram alir penggunaan jaringan syaraf tiruan,

diagram alir pengecekan akurasi, diagram alir pengecekan status *training*, diagram alir pelatihan jaringan syaraf tiruan, diagram alir sub proses pelatihan jaringan syaraf tiruan, diagram alir penyimpanan jaringan syaraf tiruan yang sudah ada, diagram alir untuk membuka kembali jaringan syaraf tiruan yang sudah disimpan, dan diagram alir untuk merubah parameter pelatihan jaringan syaraf tiruan.

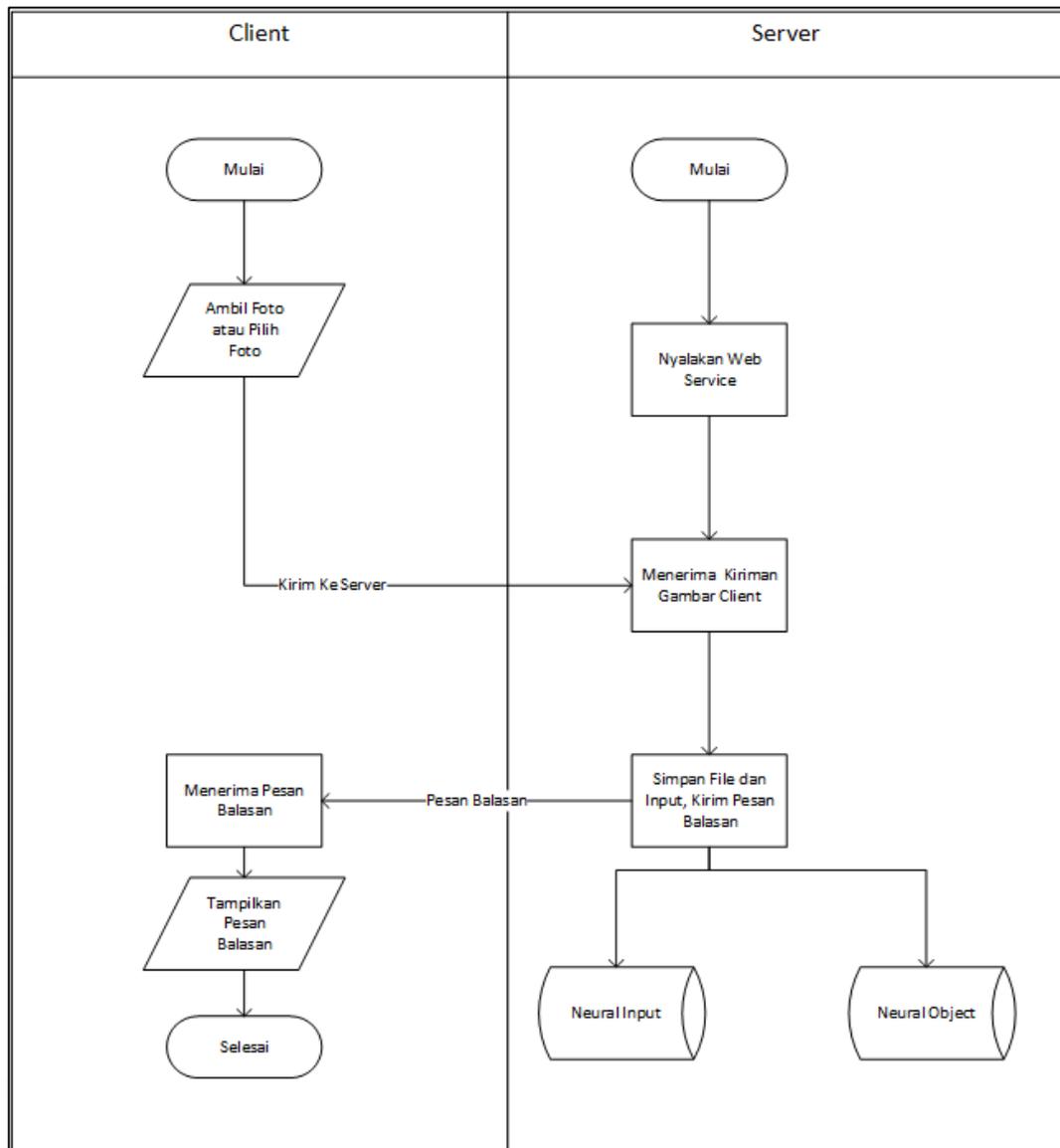
A. Diagram Alir Sistem

Proses dimulai dengan masuk ke menu utama. Setelah dari menu utama, client memilih apakah ingin untuk menambah gambar. Apabila ingin untuk menambah gambar, maka masuk ke proses untuk menambah gambar. Apabila tidak ingin untuk menambah gambar, maka lanjut ke pilihan berikutnya. Kemudian apakah client ingin untuk menghapus gambar. Apabila ingin untuk menghapus gambar, maka masuk ke proses untuk menghapus gambar. Apabila tidak ingin untuk menghapus gambar, maka lanjut ke pilihan selanjutnya. Hal tersebut diulangi untuk setiap pilihan yang ada. Mulai dari melihat gambar, *create network*, *reset network*, *load network*, *save network*, *use network*, ubah parameter *training*, *training* jaringan syaraf tiruan, *check is training*, dan check akurasi jaringan syaraf tiruan. Apabila tidak masuk ke pilihan tersebut, maka sistem akan selesai.



Gambar 3.2 Diagram alir sistem

B. Diagram Alir Penambahan Gambar



Gambar 3.3 Diagram alir penambahan gambar

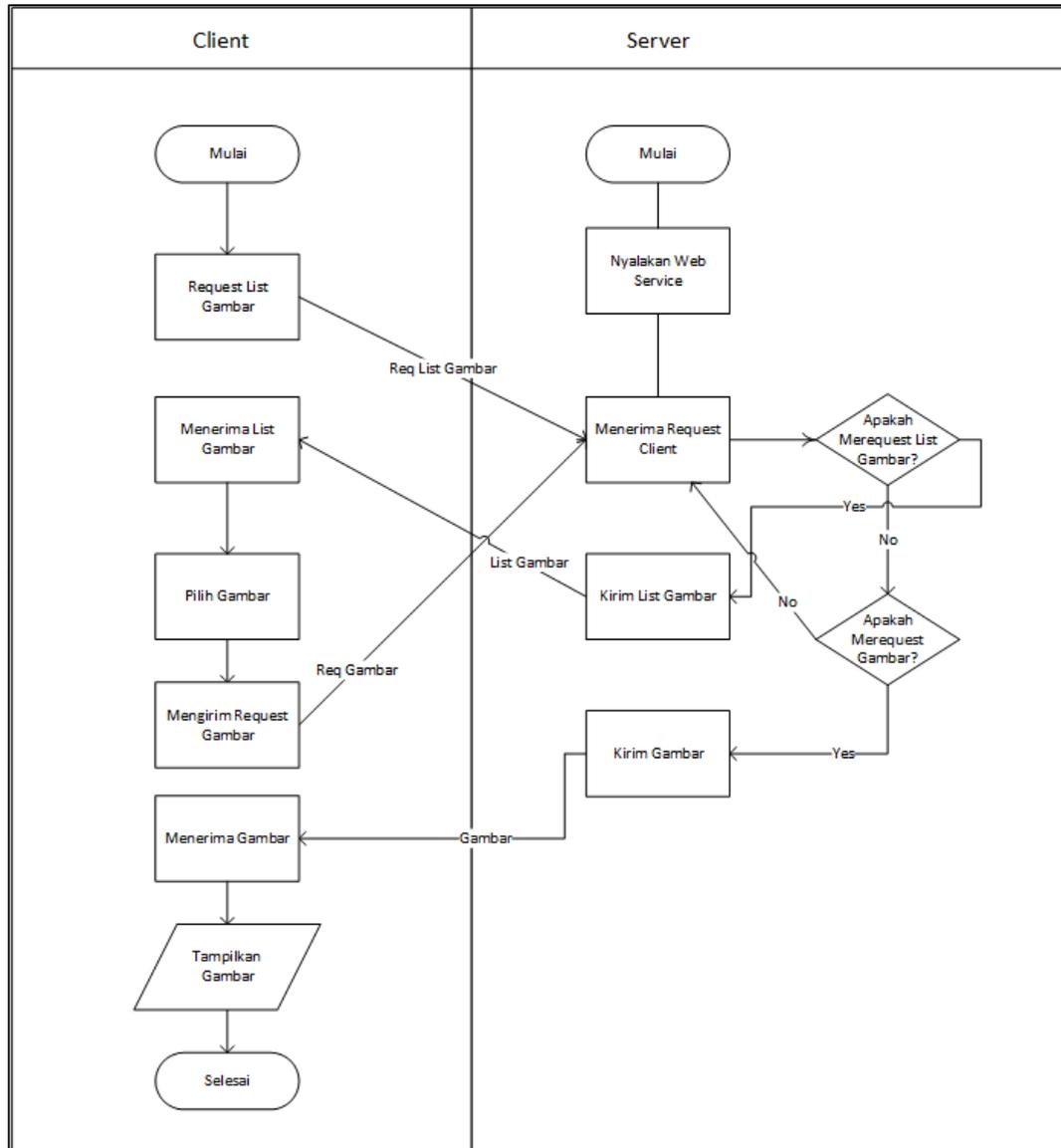
Proses dimulai dengan *server* pertama kali akan menyalakan *web service* terlebih dahulu, lalu akan menunggu untuk menerima *input* dari *client*. Kemudian *client* memilih gambar atau mengambil gambar secara langsung kemudian memasukkan data yang diperlukan, seperti nama bunga lalu tentukan gambar tersebut sebagai data *training* atau data *testing*. Kemudian *client* akan mengirimkan

data-data tersebut ke *server*. *Server* akan menerima gambar tersebut dan akan menyimpan gambar tersebut pada *harddisk server* dan akan memasukkan nama *file* serta info gambar tersebut ke *table* *neural_input*. Apabila nama bunga adalah nama baru di database, maka *server* juga memasukkan nama tersebut ke *table* *neuralobject*. Setelah itu *server* akan mengembalikan balasan kepada *client* yang memberitahu *client* bahwa proses penambahan gambar telah selesai. *Client* akan menerima pesan balasan tersebut dan menampilkannya kepada *user*.

C. Diagram Alir Melihat Gambar

Proses dimulai dengan *server* menyalakan *web service* terlebih dahulu. Setelah itu, *client* mengirim *request list* gambar yang ada pada *server*. *Server* menerima *request* tersebut dan mengembalikan *list* dari gambar yang ada pada *server*. Setelah *client* menerima *list* gambar tersebut, maka *client* memilih salah satu dari gambar untuk ditampilkan. Setelah *client* memilih suatu gambar, maka *client* mengirim *request* gambar tersebut ke *server* dan *server* akan mengirim gambar tersebut ke *client*.

UMMN

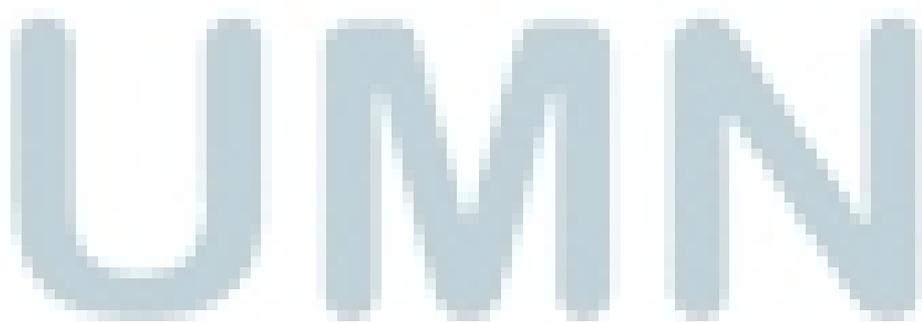


Gambar 3.4 Diagram alir melihat gambar

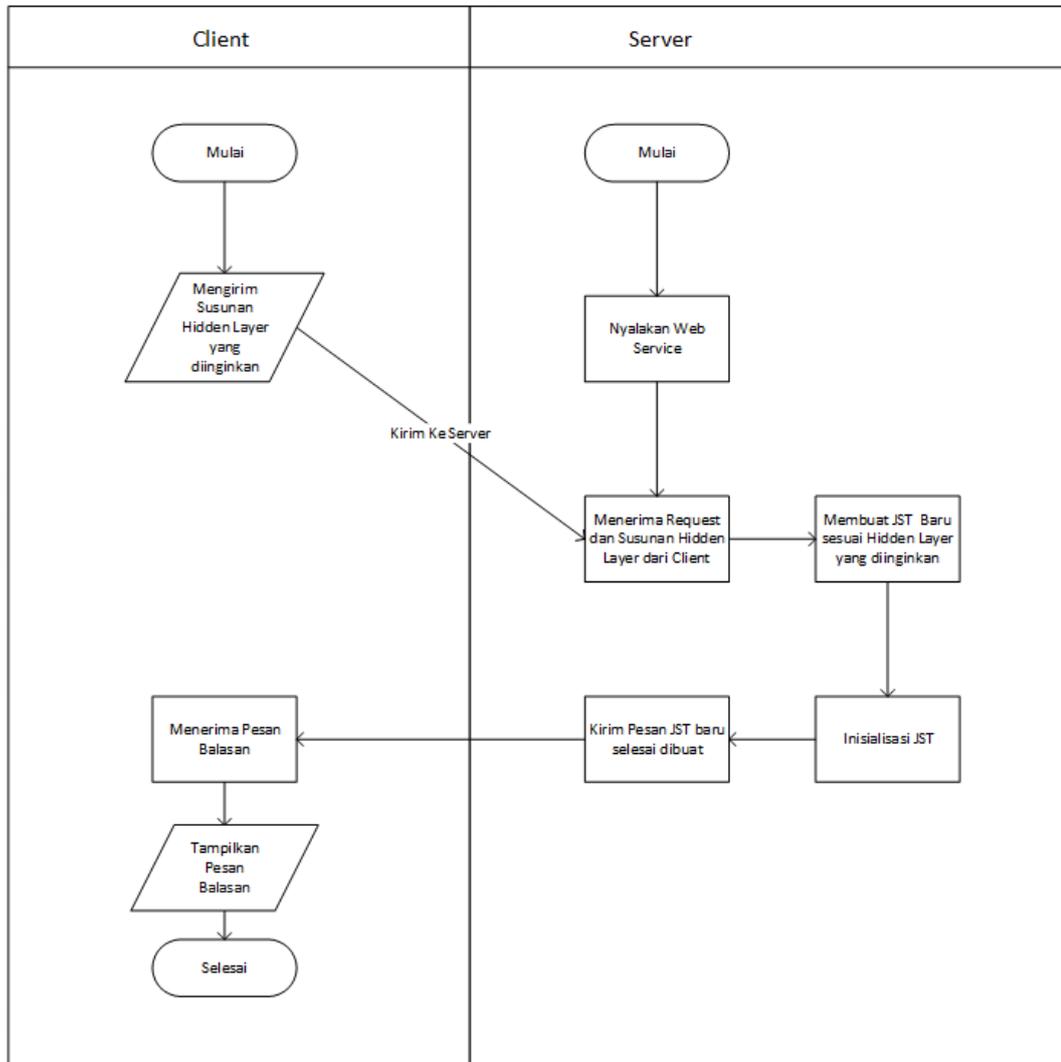
D. Diagram Alir Penghapusan Gambar

Proses diagram alir di bawah dimulai dengan *server* menyalakan *web service* terlebih dahulu. Setelah itu *client* mengirim *request list* gambar yang ada pada *server*. *Server* menerima *request* tersebut dan mengembalikan *list* dari gambar yang ada pada *server*. Setelah *client* menerima *list* gambar tersebut, maka *client* memilih salah satu dari gambar untuk ditampilkan. Setelah *client* memilih suatu gambar,

maka client mengirim *request* gambar tersebut ke *server* dan *server* akan mengirim gambar tersebut ke *client*. Setelah itu, *client* akan menerima gambar dan memperlihatkan gambar tersebut ke *user*. Setelah melihat gambar yang dipilih dan yakin akan menghapusnya, maka *client* mengirim perintah ke *server* untuk menghapus gambar tersebut. Setelah *server* menerima pesan tersebut, *server* akan menghapus gambar tersebut dari *harddisk* dan dari *table* neuralinput. Setelah itu *server* mengirim pesan ke *client* bahwa gambar sudah dihapus.

The logo of Universitas Muhammadiyah Negeri (UMMN) is displayed. It consists of a circular emblem containing a stylized building with several windows, and the acronym 'UMMN' written in large, bold, blue capital letters below the emblem.

E. Diagram Alir Proses Pembuatan Jaringan Syaraf Tiruan Baru

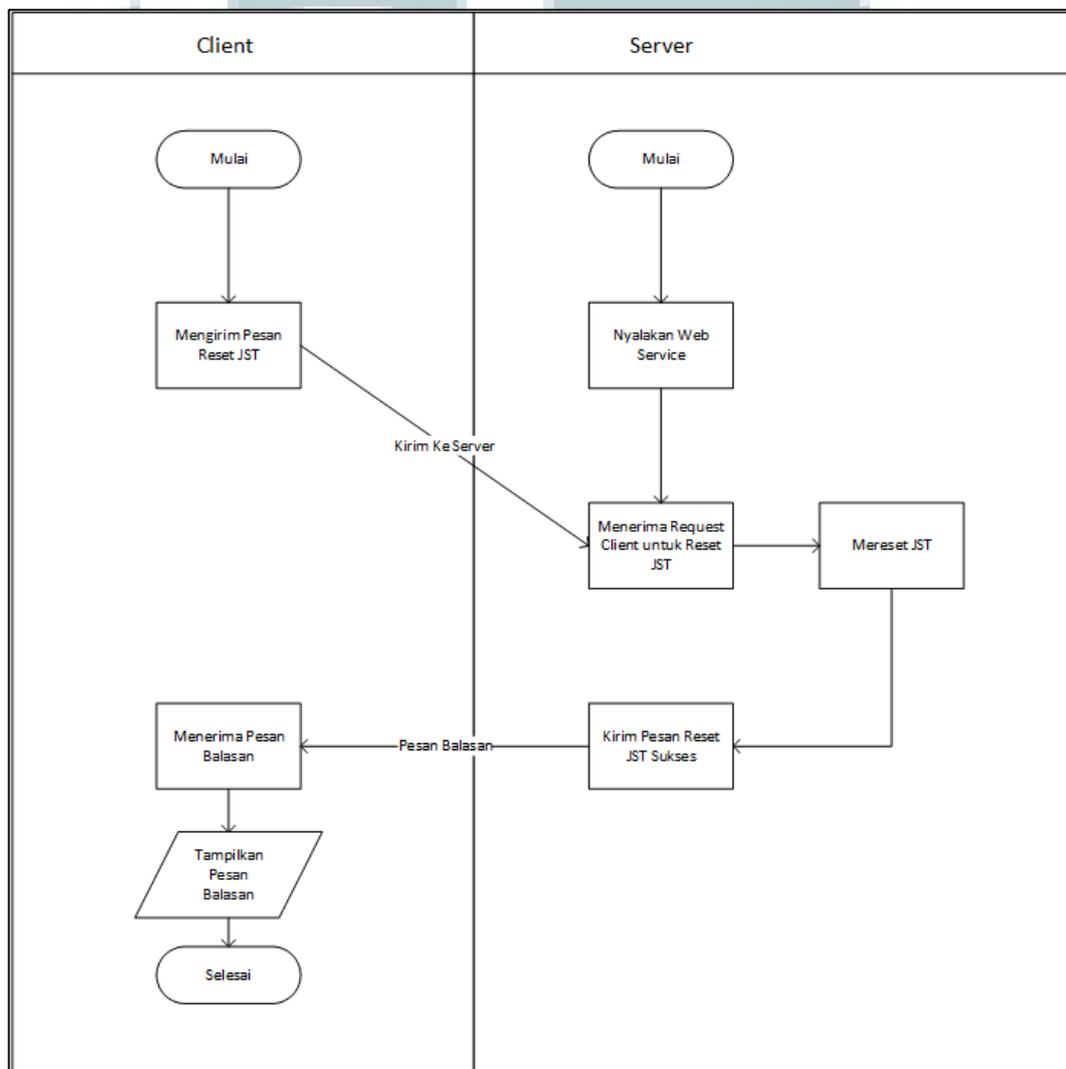


Gambar 3.6 Diagram alir proses pembuatan jaringan syaraf tiruan baru

Proses diagram alir di atas dimulai dengan *server* menyalakan *web service* terlebih dahulu. Setelah itu *client* mengirim susunan *hidden layer* yang diinginkan kepada pada *server*. *Server* menerima susunan *hidden layer* tersebut dan membuat jaringan syaraf tiruan yang baru sesuai dengan susunan *hidden layer* tersebut. Setelah itu jaringan syaraf tiruan yang baru diinisialisasikan bobot awalnya dan

server mengirim pesan ke *client* bahwa jaringan syaraf tiruan yang baru telah selesai dibuat. *Client* akan menerima pesan balasan dari *server* dan setelah itu menampilkan pesan balasan tersebut ke *user*.

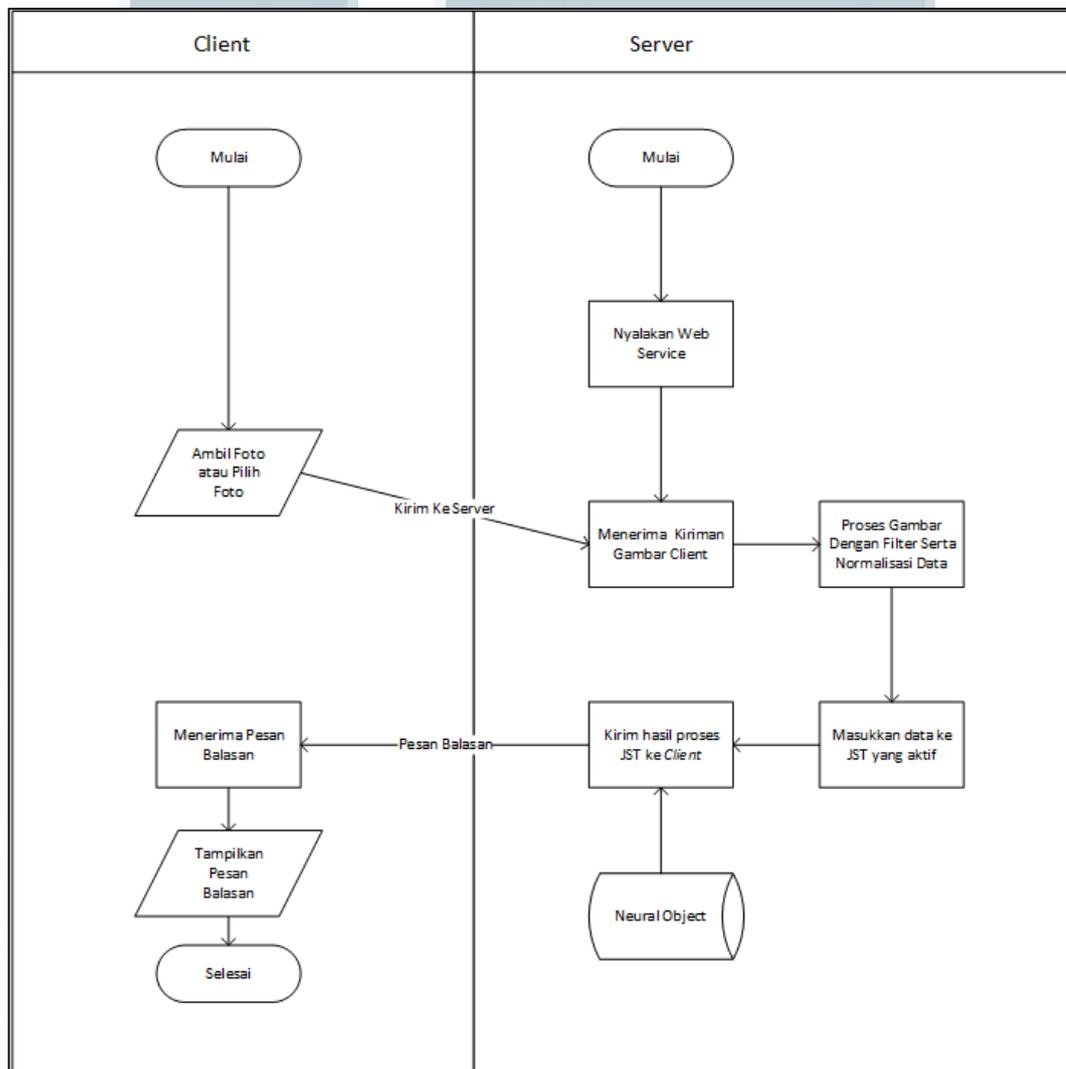
F. Diagram Alir Proses *Reset* Jaringan Syaraf Tiruan



Gambar 3.7 Diagram alir proses *reset* jaringan syaraf tiruan

Proses diagram alir di atas dimulai dengan *server* menyalakan *web service* terlebih dahulu. Setelah itu *client* mengirim *request* untuk *reset* jaringan syaraf tiruan kepada pada *server*. *Server* menerima *request* tersebut dan melakukan *reset* jaringan syaraf tiruan. Setelah itu, *server* mengirim pesan ke *client* bahwa jaringan syaraf tiruan telah selesai dilakukan *reset*. *Client* menerima pesan balasan dari *server* dan menampilkannya kepada *user*.

G. Diagram Alir Proses Penggunaan Jaringan Syaraf Tiruan

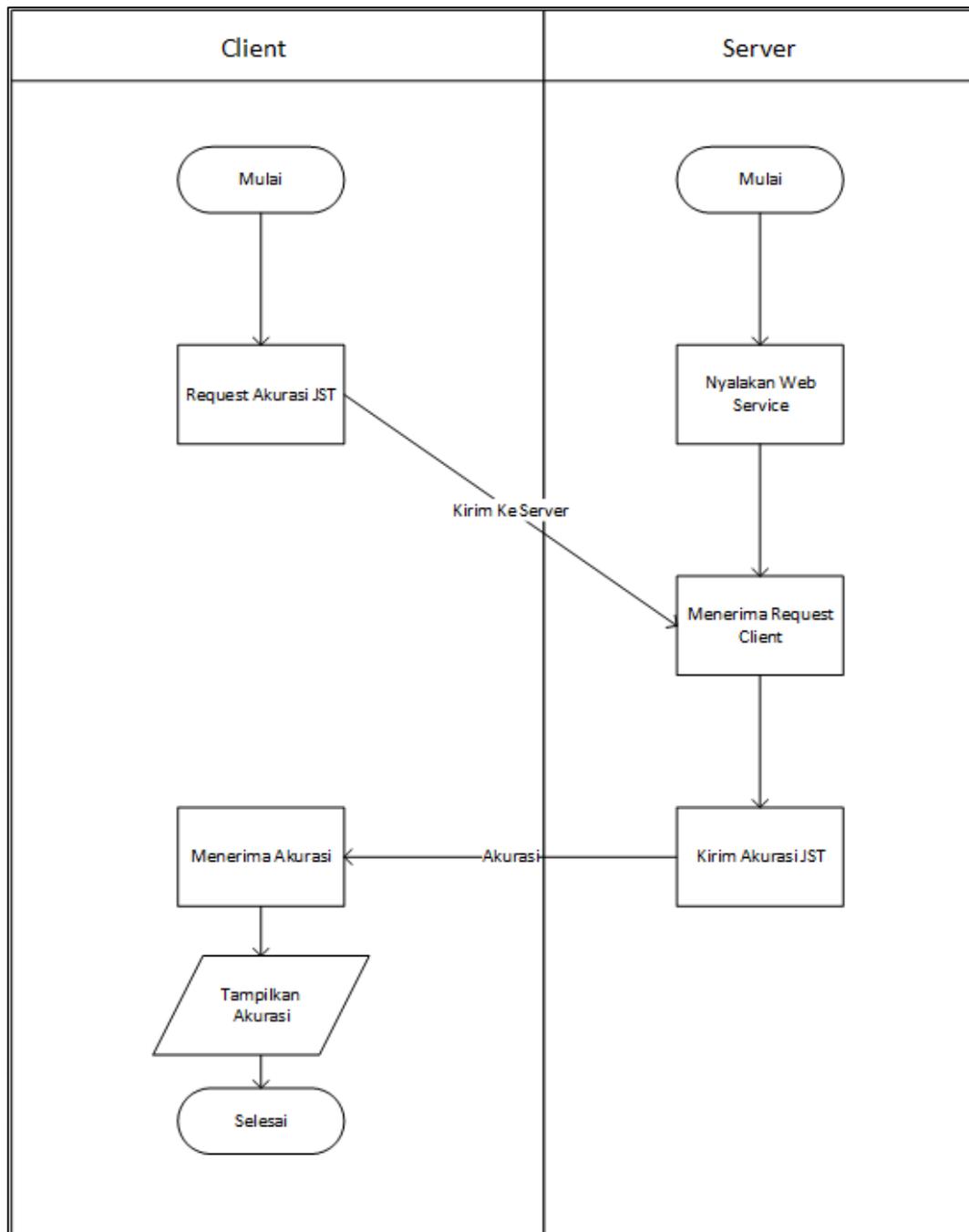


Gambar 3.8 Diagram alir proses penggunaan jaringan syaraf tiruan

Proses diagram alir dimulai dengan *server* menyalakan *web service* terlebih dahulu. Setelah itu *client* mengambil gambar atau memilih gambar dan mengirimkannya ke *server*. *Server* menerima gambar tersebut dan melakukan pengolahan gambar dan diekstrak fiturnya serta dimasukkan ke dalam jaringan syaraf tiruan untuk dikenali fotonya. Setelah itu *server* mencari hasil keluaran pada *database* neuralobject dan mengirim nama bunga yang ditemukan pada *database* ke *client*. *Client* menerima pesan balasan hasil pengenalan jaringan syaraf tiruan pada *server* dan menampilkannya kepada *user*.

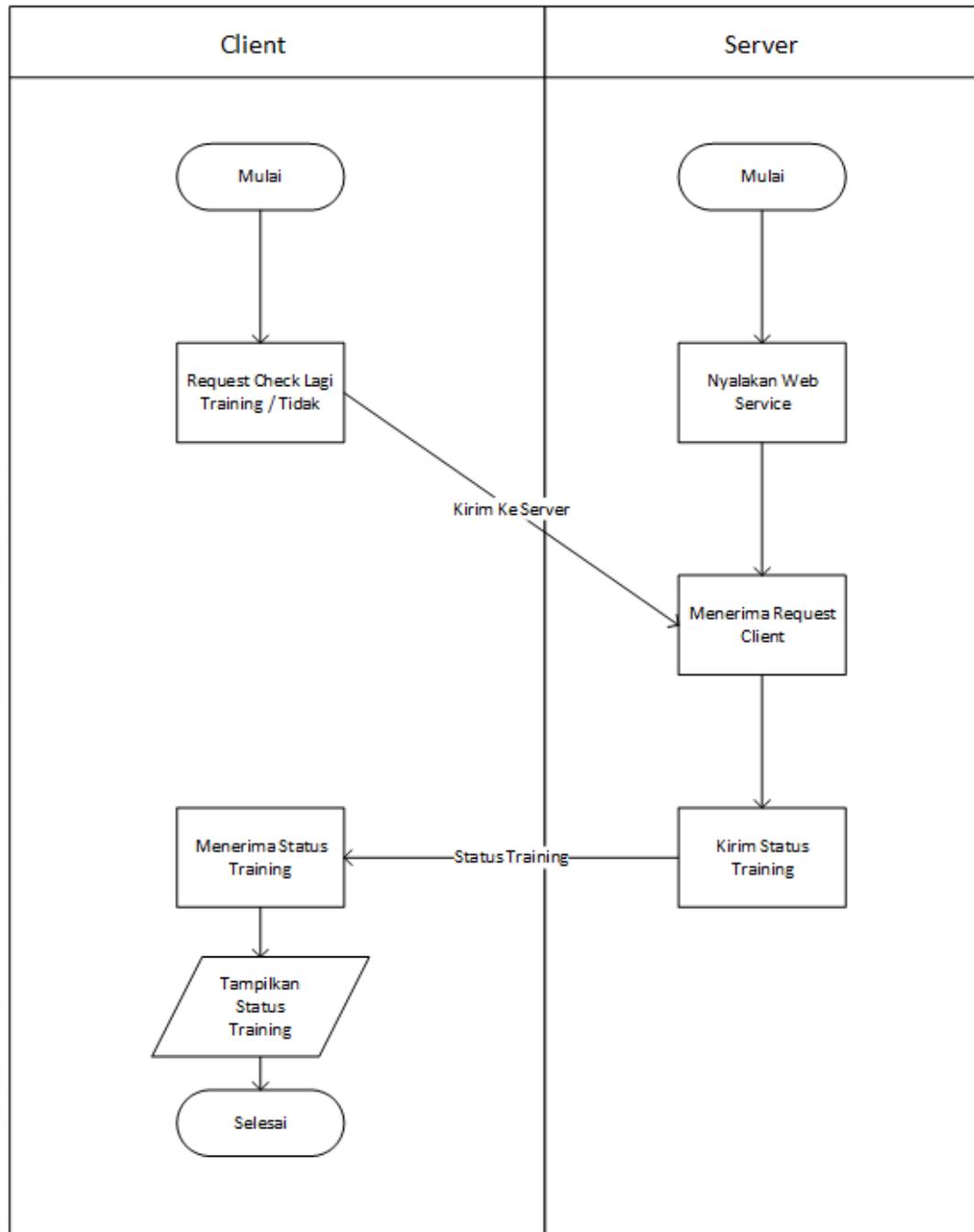
H. Diagram Alir Pengecekan Akurasi

Proses diagram alir di bawah dimulai dengan *server* menyalakan *web service* terlebih dahulu. Setelah itu *client* mengirim *request* akurasi jaringan syaraf tiruan yang ada pada *server*. *Server* menerima *request* tersebut dan mengembalikan akurasi jaringan syaraf tiruan yang sudah dilatih sebelumnya. Akurasi tersebut dihitung dari pelatihan terakhir dan hanya dihitung menggunakan data *testing* yang ada dari pelatihan terakhir. Setelah *client* menerima akurasi jaringan syaraf tiruan tersebut, maka *client* menampilkan akurasi tersebut.



Gambar 3.9 Diagram alir pengecekan akurasi jaringan syaraf tiruan

I. Diagram Alir Pengecekan Status *Training*

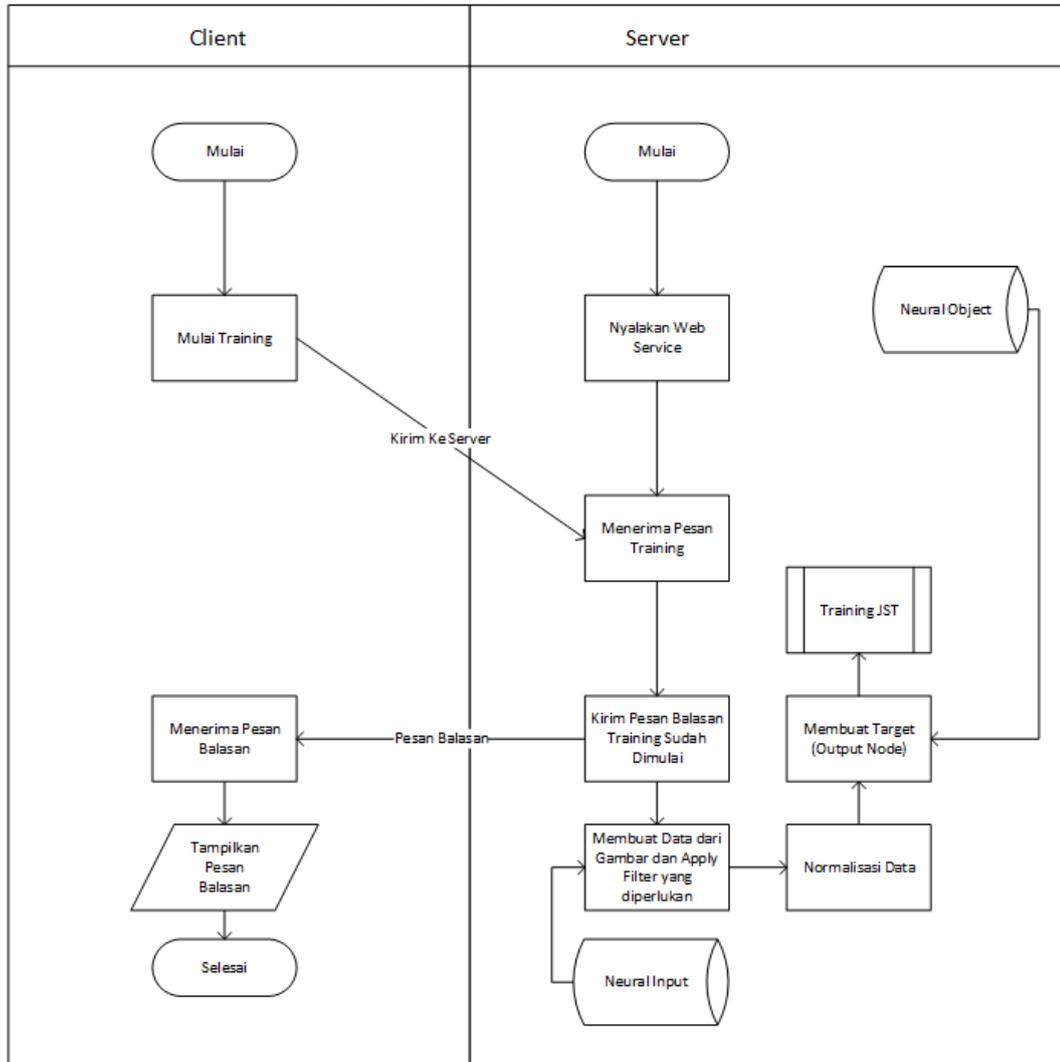


Gambar 3.10 Diagram alir pengecekan status *training*

Proses diagram alir dimulai dengan *server* menyalakan *web service* terlebih dahulu. Setelah itu *client* mengirim *request* status *training* jaringan syaraf tiruan

yang ada pada *server*. *Server* menerima *request* tersebut dan mengirimkan *status training*. *Client* menerimanya dan menampilkan balasan *server*.

J. Diagram Alir Pelatihan Jaringan Syaraf Tiruan

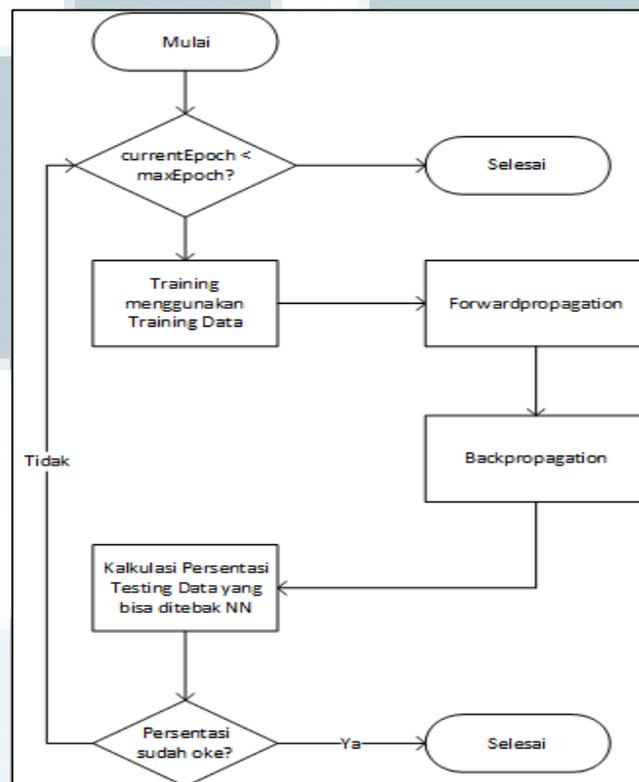


Gambar 3.11 Diagram alir pelatihan jaringan syaraf tiruan

Proses diagram alir dimulai dengan *server* menyalakan *web service* terlebih dahulu. Setelah itu *client* mengirim *request* untuk melakukan *training* jaringan syaraf tiruan yang ada pada *server*. *Server* menerima *request* tersebut dan memulai *training* serta mengirimkan pesan balasan ke *client* bahwa *training* sudah dimulai.

Client menerima pesan tersebut dan menampilkannya. *Server* kemudian mengambil *list* gambar dan *info* dari *table* neuralinput dan membuat data *training* dan *testing* dari gambar-gambar yang ada. Setelah itu data-data tersebut dilakukan normalisasi dan gambarnya diproses agar fitur yang diekstrak dapat dimasukkan ke dalam jaringan syaraf tiruan. Kemudian *server* membuat target dari objek yang ada pada *database* neuralobject. Setelah itu dilakukan *training* jaringan syaraf tiruan sampai tercapai akurasi yang diinginkan atau tercapai *epoch* maksimal.

K. Diagram Alir Sub Proses Pelatihan Jaringan Syaraf Tiruan

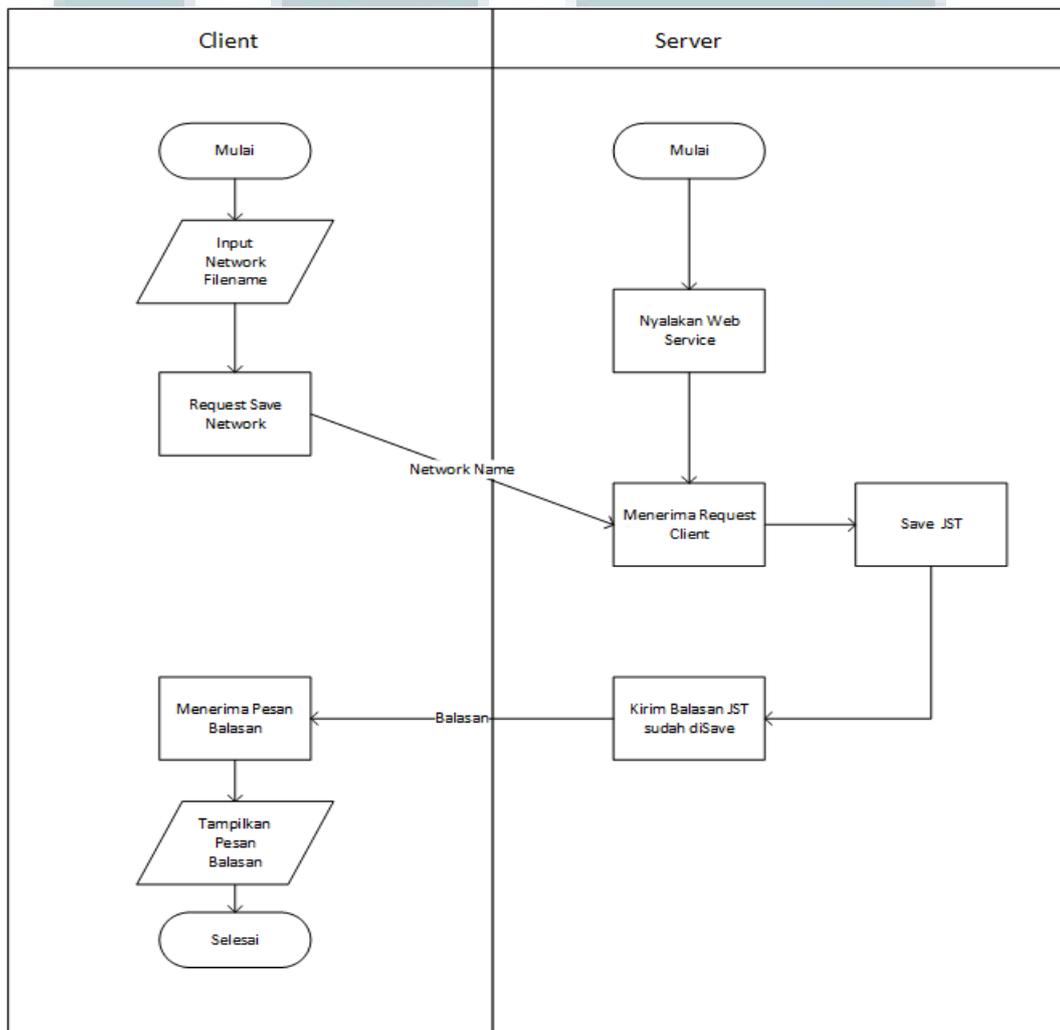


Gambar 3.12 Diagram alir sub proses pelatihan jaringan syaraf tiruan

Pada diagram alir ini dimulai dengan pengecekan apakah *currentEpoch* belum mencapai *epoch* maksimum. Setelah itu dilakukan *forwardpropagation* dan

backpropagation. Pada *forwardpropagation* hanya menjalankan data maju, menghitung hasil keluaran pada jaringan syaraf tiruan. Pada *backpropagation* mengubah bobot, sehingga bobot mengarah ke jaringan syaraf tiruan yang diinginkan. Setelah itu dikalkulasikan persentase keberhasilan jaringan syaraf tiruan untuk menebak data *input testing* dengan benar. Apabila sudah mencapai persentase tertentu, maka *training* dianggap sudah cukup. Apabila jumlah *epoch* sudah melewati *maximumEpoch*, maka juga dianggap sudah cukup.

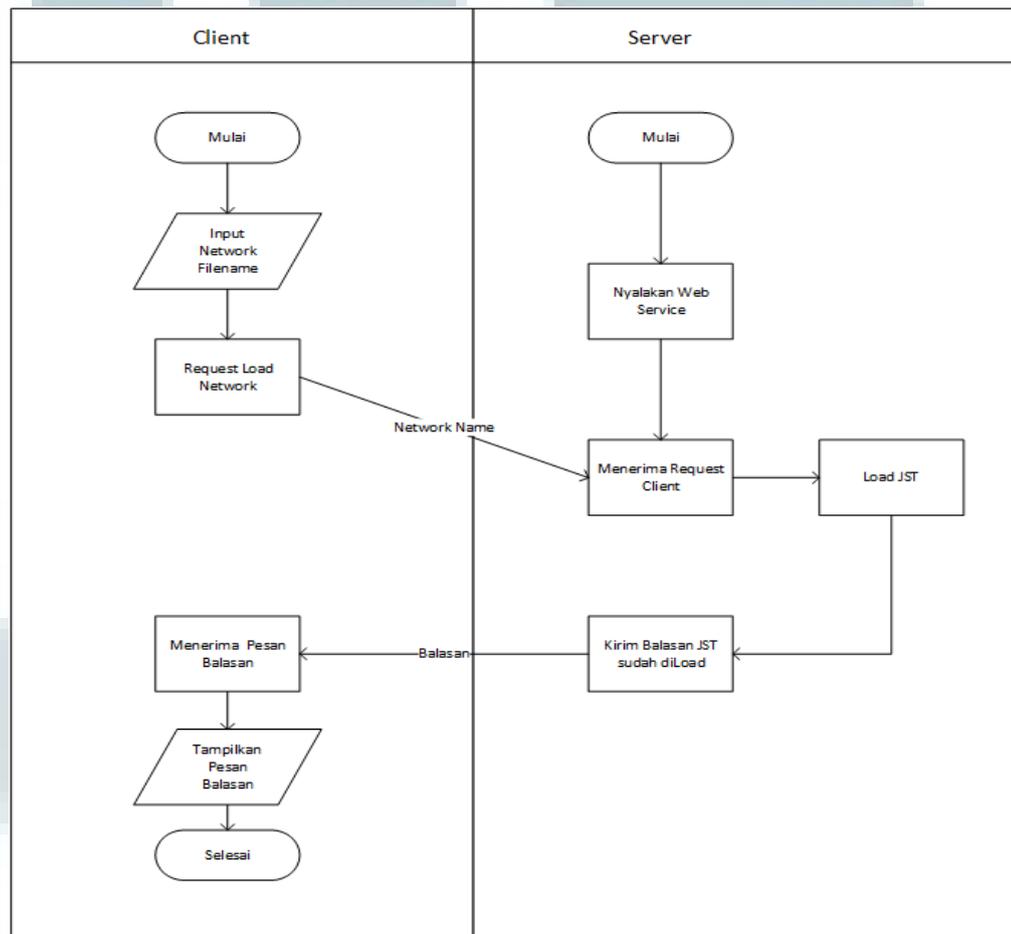
L. Diagram Alir Penyimpanan Jaringan Syaraf Tiruan



Gambar 3.13 Diagram alir penyimpanan jaringan syaraf tiruan

Proses diagram alir dimulai dengan *server* menyalakan *web service* terlebih dahulu. Setelah itu *client* menginput nama *file* yang akan digunakan untuk *filename* jaringan syaraf tiruan yang akan disimpan dan mengirim *request* untuk melakukan penyimpanan jaringan syaraf tiruan yang ada pada *server*. *Server* menerima *request* tersebut dan menyimpan jaringan syaraf tiruan sesuai nama *file* yang diberikan *client* ke *harddisk* dan mengirimkan pesan balasan ke *client* bahwa jaringan syaraf tiruan sudah disimpan. *Client* menerima pesan tersebut dan menampilkannya.

M. Diagram Alir *Load* Jaringan Syaraf Tiruan



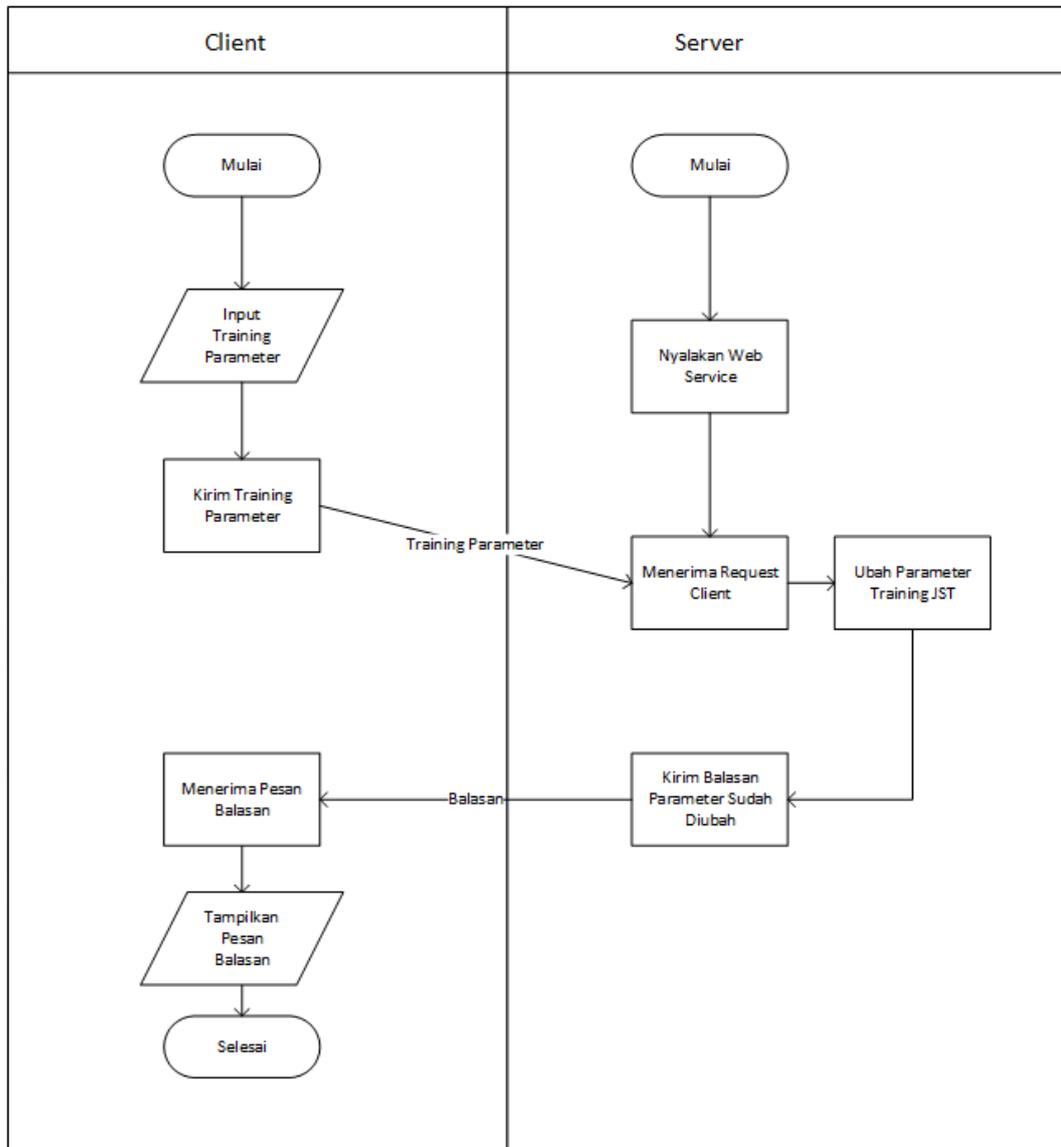
Gambar 3.14 Diagram alir *load* jaringan syaraf tiruan

Proses diagram alir dimulai dengan *server* menyalakan *web service* terlebih dahulu. Setelah itu *client* menginput nama *file* yang akan digunakan untuk *filename* jaringan syaraf tiruan yang akan di-*load* dan mengirim *request* untuk melakukan *load* jaringan syaraf tiruan dengan nama *file* tertentu yang ada pada *server*. *Server* menerima *request* tersebut dan membuka kembali jaringan syaraf tiruan yang sudah disimpan sesuai nama *file* yang diberikan *client* dari *harddisk* dan mengirimkan pesan balasan ke *client* bahwa jaringan syaraf tiruan sudah di-*load*. *Client* menerima pesan tersebut dan menampilkannya.

N. Diagram Alir Perubahan *Training Parameter*

Proses diagram alir di bawah dimulai dengan *server* menyalakan *web service* terlebih dahulu. Setelah itu *client* meng-input *training parameter* dan mengirim *request* perubahan *training parameter* jaringan syaraf tiruan yang ada pada *server*. *Server* menerima *request* tersebut dan mengubah *parameter* jaringan syaraf tiruan sesuai permintaan *client*. *Server* juga mengirimkan pesan bahwa *parameter* sudah berhasil diubah. *Client* menerima pesan tersebut dan menampilkannya.

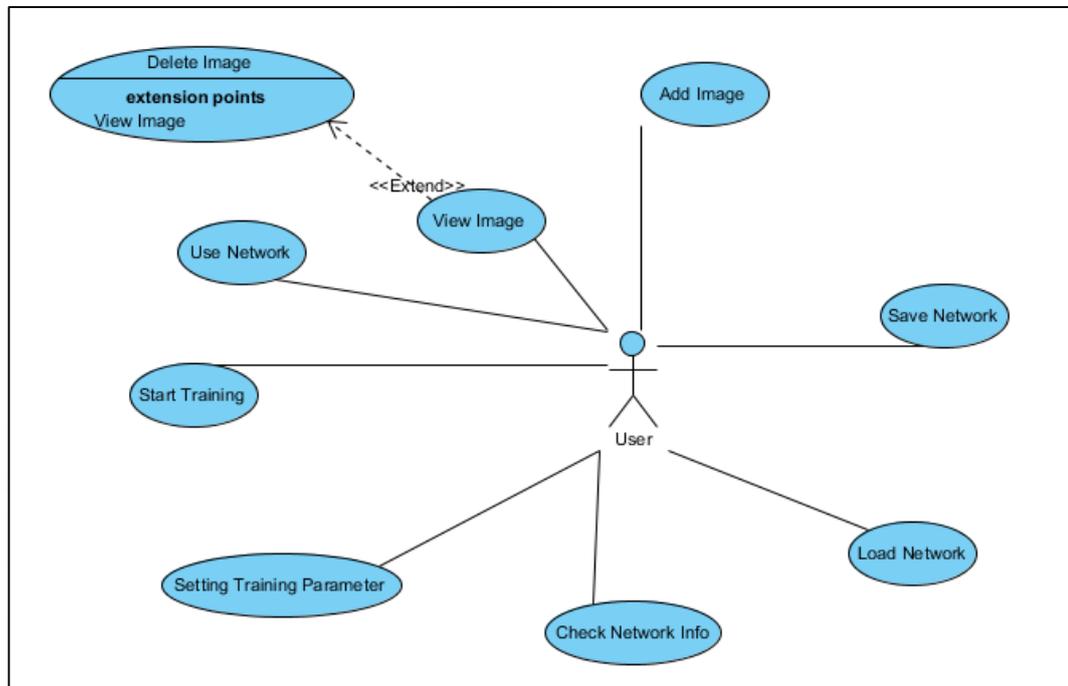
UMMN



Gambar 3.15 Diagram alir perubahan *training parameter*

UMMN

3.3.2 Use Case Diagram

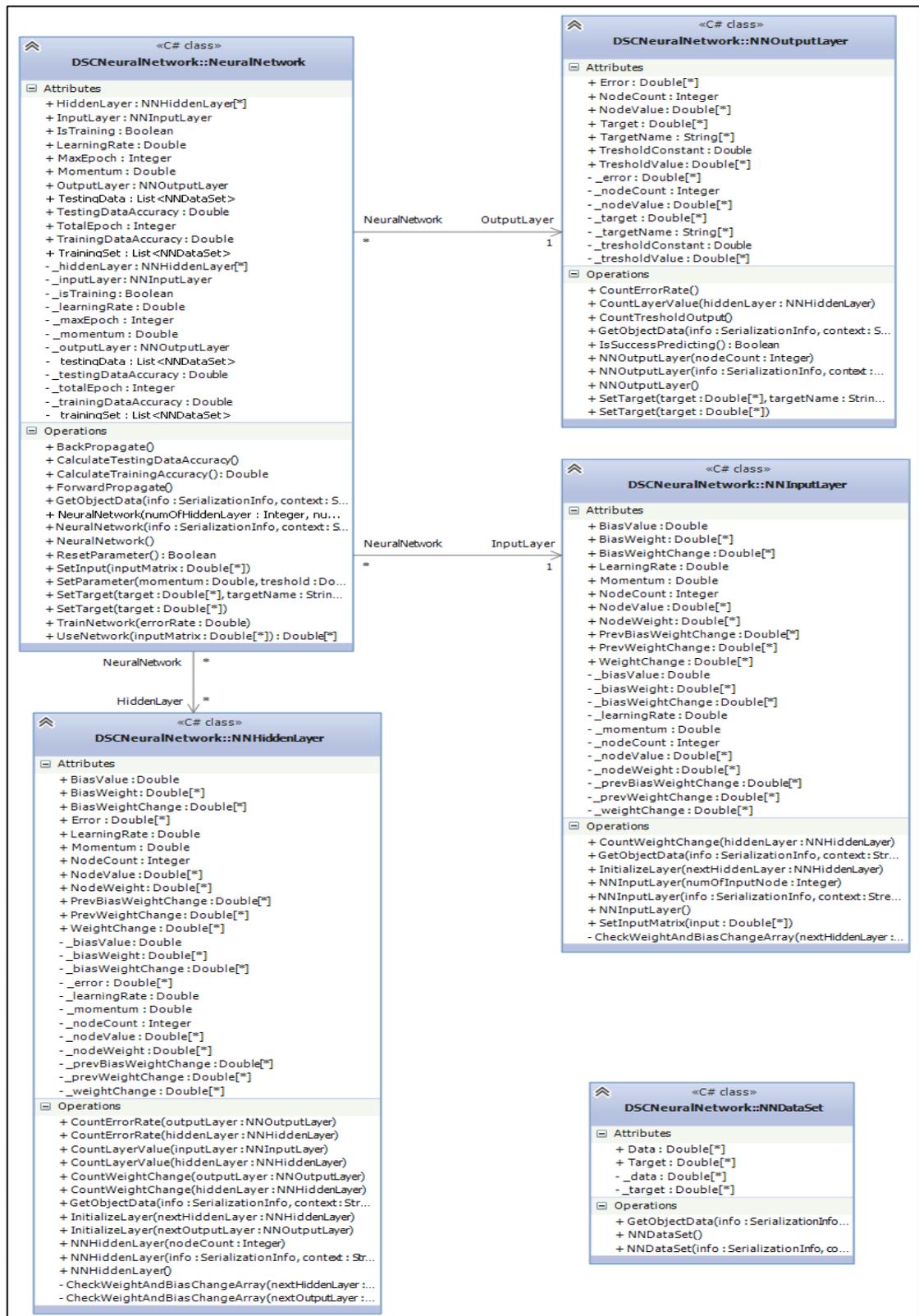


Gambar 3.16 Use case diagram

Gambar di atas adalah *use case* yang bisa dilakukan oleh *user*. *User* dapat melihat gambar, menghapus gambar, menambah gambar, menggunakan *network*, memulai *training*, menyimpan *network*, *load network*, *check network info*, *setting training parameter*.

U
M
M
N

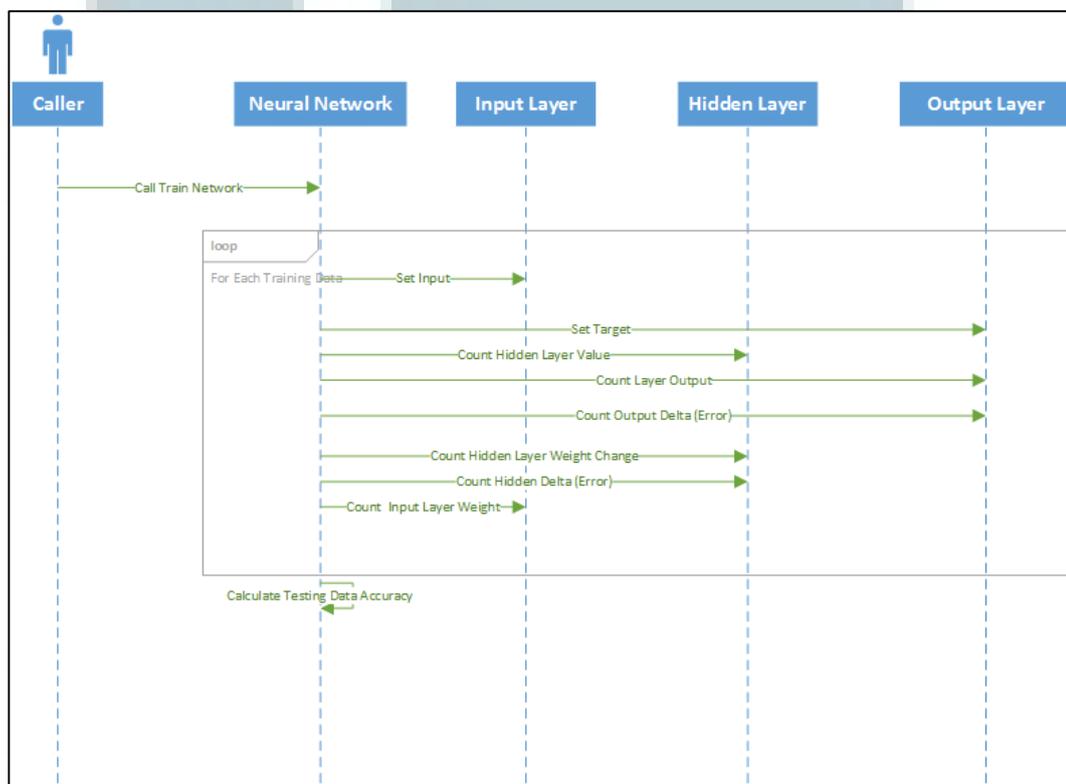
3.3.3 Class Diagram



Gambar 3.17 Class diagram

Gambar di atas adalah *class diagram* dari jaringan syaraf tiruan yang akan digunakan. Jaringan syaraf tiruan akan menjadi satu objek *NeuralNetwork* dan akan mempunyai objek *NNInputLayer*, *NNHiddenLayer*, *NNOutputLayer* sesuai dengan hubungan di atas.

3.3.4 Sequence Diagram



Gambar 3.18 *Sequence Diagram Training Network*

Proses *training network* dimulai dengan panggilan ke *neural network*. Setelah itu akan dilakukan iterasi untuk setiap *training data*. *Neural network* akan melakukan *set input* pada *input layer* dan *set target* pada *output layer*. Setelah *neural network* akan melakukan panggilan untuk *hidden layer* agar menghitung *layer value* untuk *hidden layer*. Setelah itu *neural network* juga melakukan hal yang

sama untuk *output layer*. Kemudian *neural network* akan menghitung *error* untuk *output layer* dan *hidden layer* dan akan memperbaiki bobot dari *hidden layer* serta *input layer*. Setelah itu dihitung akurasi untuk menebak *testing data*.

3.4 Perancangan Database

Database yang digunakan oleh sistem adalah *MySQL*. Berikut struktur tabel-tabel yang terdapat di dalamnya.

1. Nama Tabel : *neuralinput*
- Fungsi : menyimpan nama *file* gambar dan info gambar
- Primary Key* : *id*
- Foreign Key* : *object_id*

Tabel 3.1 Tabel *neuralinput*

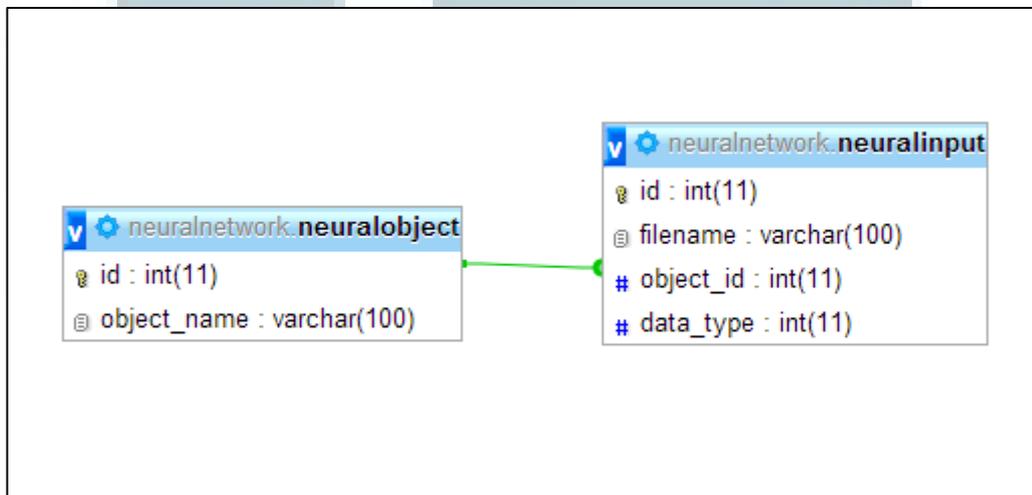
| No | Nama Field | Tipe Data | Panjang | Keterangan |
|----|------------------|----------------|---------|---|
| 1 | <i>id</i> | <i>int</i> | 11 | Primary key tabel |
| 2 | <i>filename</i> | <i>varchar</i> | 100 | Nama file gambar |
| 3 | <i>object_id</i> | <i>int</i> | 11 | ID object |
| 4 | <i>data_type</i> | <i>int</i> | 11 | Tipe data apakah data training atau testing |

2. Nama Tabel : *neuralobject*
- Fungsi : menyimpan nama *neural object*
- Primary Key* : *id*
- Foreign Key* : -

Tabel 3.2 Tabel neuralobject

| No | Nama Field | Tipe Data | Panjang | Keterangan |
|----|-------------|-----------|---------|---------------------------|
| 1 | id | int | 11 | Id <i>neural object</i> |
| 2 | object_name | varchar | 100 | Nama <i>neural object</i> |

Adapun hubungan antar tabel yang dibuat dalam *database* adalah sebagai berikut.



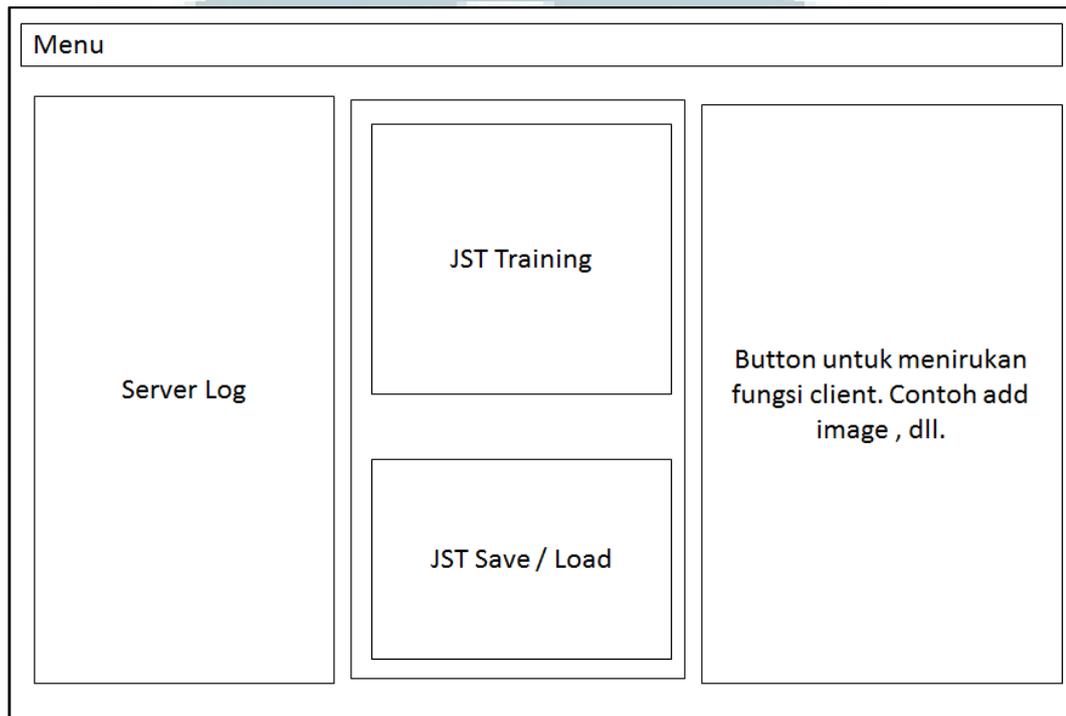
Gambar 3.19 ERD

Tabel neuralobject dengan neuralinput dipisah karena neuralinput hanya menyimpan nama *file* untuk *input* jaringan syaraf tiruan. Neuralobject menyimpan nama dari jenis bunga yang disimpan. Agar tidak mengulang pada neuralinput maka dibuat neuralobject dan neuralinput dipisah.

3.5 Desain Antarmuka

Pada bagian ini, ditampilkan beberapa sketsa antarmuka sistem dari sisi pengguna.

3.5.1 Form Server

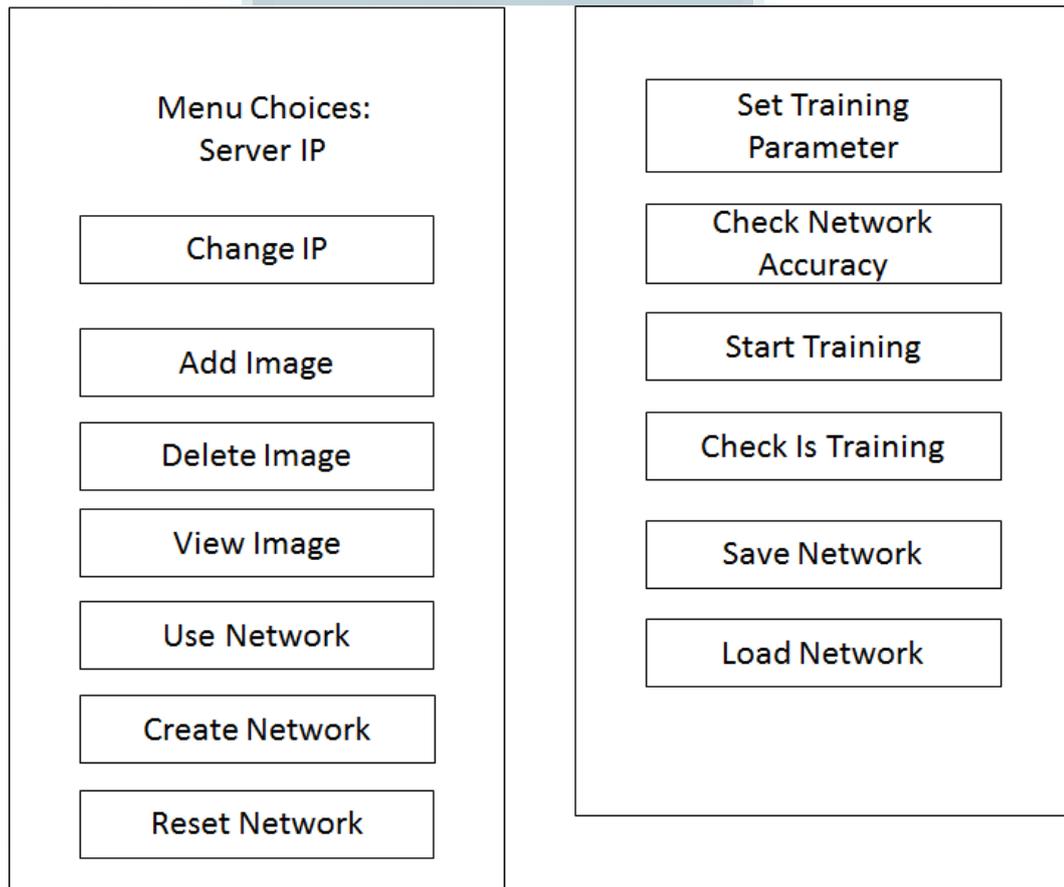


Gambar 3.20 Desain *form server*

Form ini adalah *form* yang pertama kali muncul saat aplikasi *server* dijalankan. Pada bagian menu terdapat pilihan untuk menyalakan *web service*. Pada tampilan ini terdapat *server log* dimana semua *request* yang diterima *server* atau proses yang dijalankan *server* akan ditampilkan. Kemudian untuk memudahkan pelatihan juga terdapat bagian untuk melatih jaringan syaraf tiruan pada bagian *JST Training*. Selain itu juga terdapat bagian untuk menyimpan jaringan syaraf tiruan atau membuka kembali jaringan syaraf tiruan yang sudah disimpan. Untuk memudahkan

dalam proses pengetesan juga terdapat beberapa fungsi untuk menirukan *request* dari *client*. Fungsi-fungsi tersebut melakukan panggilan ke *web service* bukan secara langsung memanggil fungsi dari dalam.

3.5.2 Activity Menu Utama *Client*

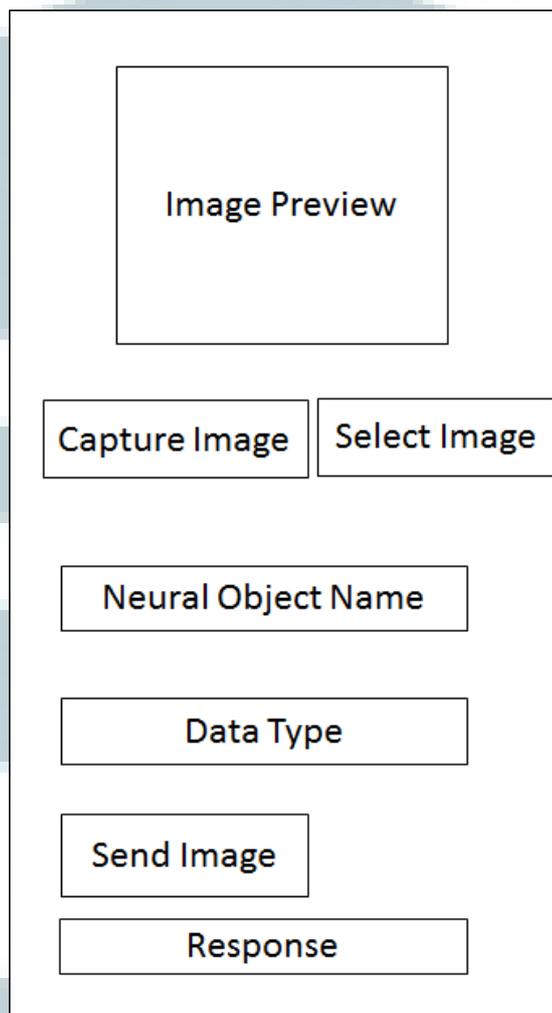


Gambar 3.21 Desain menu utama *client*

Menu utama ini ditampilkan sewaktu aplikasi *client* pada *platform* android dibuka. Untuk menu change ip, reset network, check network accuracy, start training, check is training, save network, dan load network tidak mempunyai *activity* tersendiri. Apabila hanya memerlukan satu data *input*, maka hanya memakai AlertDialog. Menu utama ini memakai *scrollview* agar pilihan menu yang

cukup banyak dapat di-*scroll* ke bawah. *Client* harus menentukan *server ip* untuk mengirim *request* ke *server* dengan benar.

3.5.3 Activity Add Image

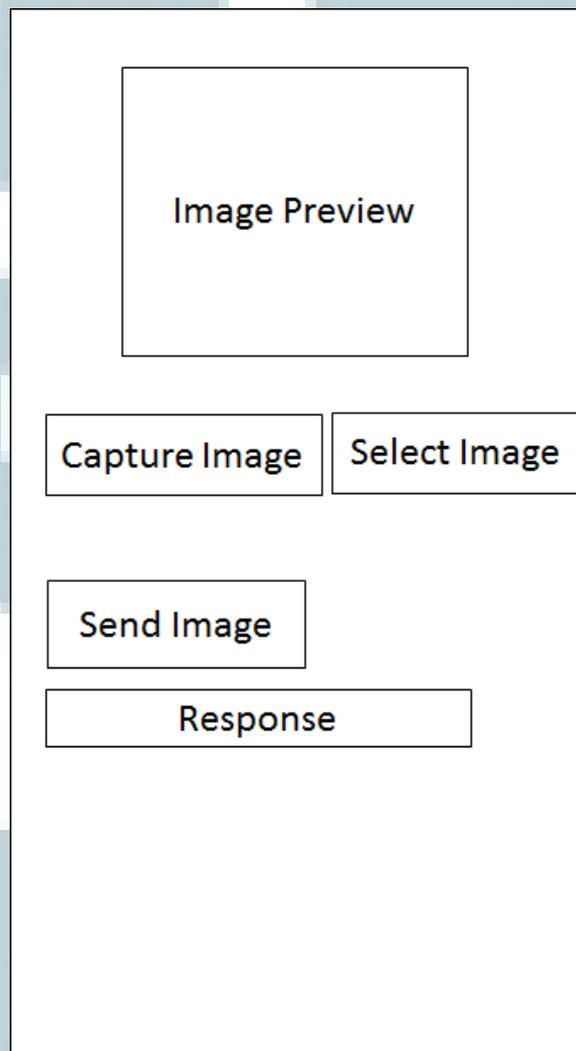


Gambar 3.22 Desain *activity add image*

Pada *activity* ini *client* dapat mengambil gambar langsung dengan kamera dengan menekan tombol *capture image*. Apabila gambar sudah ada pada *gallery*, maka bisa menekan tombol *select image*. Setelah mengambil atau memilih gambar yang akan dikirim ke *server* dan memotong gambar tersebut, *client* harus mengisi

data nama bunga pada *neural object name* dan menentukan bahwa data tersebut digunakan untuk *training* atau *testing*. Setelah itu, dengan menekan tombol *send image client* akan mengirimkan gambar tersebut beserta data info yang lain ke *server*. Balasan dari *server* akan ditampilkan pada bagian *response*.

3.5.4 Activity Use Network



Gambar 3.23 Desain use network

Pada *activity* ini, *client* dapat mengambil gambar langsung dengan kamera dengan menekan tombol *capture image*. Apabila gambar sudah ada pada *gallery*

maka bisa menekan tombol *select image*. Setelah mengambil atau memilih gambar yang akan dikirim ke *server* dan memotong gambar tersebut, *client* mengirimkan gambar tersebut ke *server*. Setelah *server* menerima dan memproses gambar tersebut, *server* akan mengirimkan hasil pengenalan jaringan syaraf tiruan terhadap gambar tersebut. Hasil dari *server* akan ditampilkan pada bagian *response*.

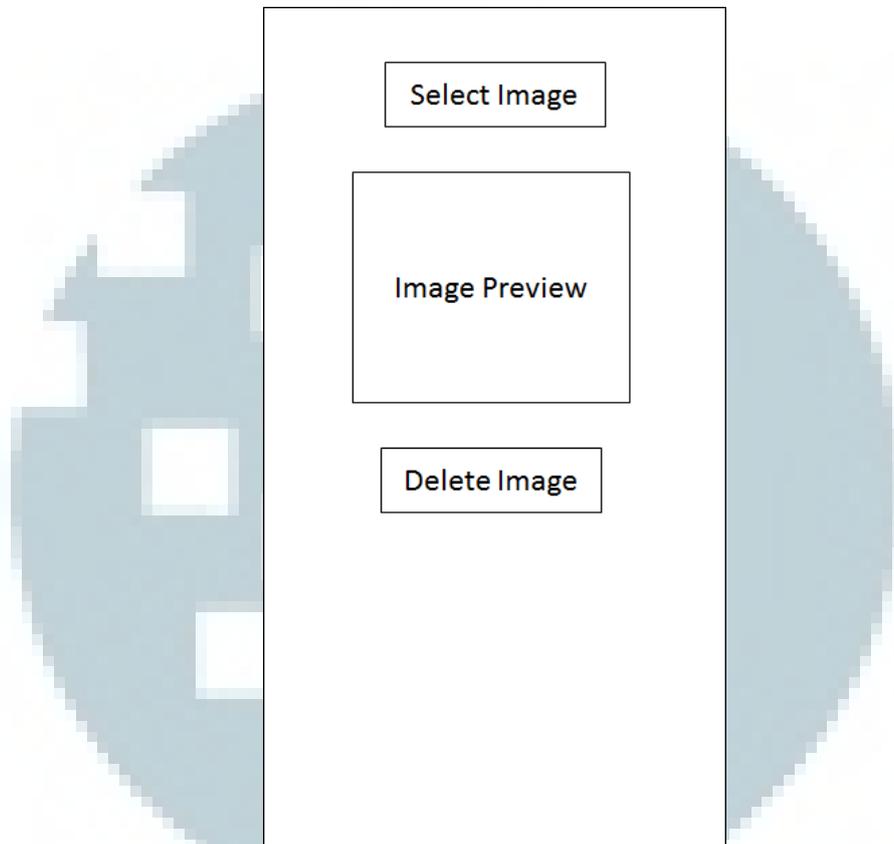
3.5.5 Activity View Image

Pada *activity* ini, *client* dapat mengambil gambar pada *server* pada *area select image*. Apabila sudah memilih gambar apa yang ingin dilihat, maka *client* akan mengirim *request* gambar tersebut ke *server* dan *server* akan mengembalikan gambar yang diinginkan dan *client* akan menampilkannya pada *area image preview*.



Gambar 3.24 Desain *view image*

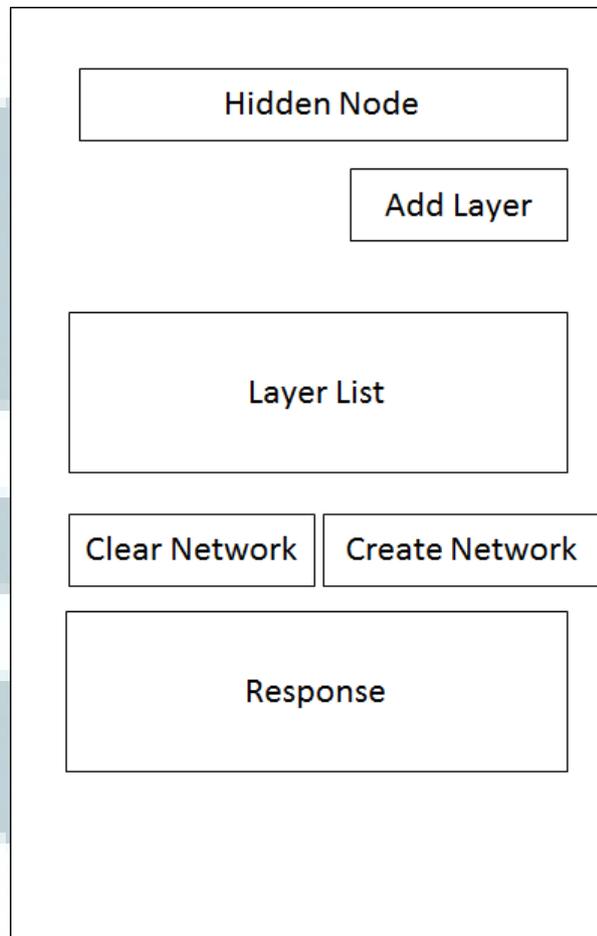
3.5.6 Activity Delete Image



Gambar 3.25 Desain *delete image*

Pada *activity* ini, *client* dapat mengambil gambar pada *server* pada *area select image*. Apabila sudah memilih gambar apa yang ingin dilihat, maka *client* akan mengirim *request* gambar tersebut ke *server* dan *server* akan mengembalikan gambar yang diinginkan dan *client* akan menampilkannya pada *area image preview*. Setelah yakin gambar tersebut yang akan dihapus maka *client* dapat memilih tombol *delete image* untuk menghapus gambar tersebut dari *server*.

3.5.7 Activity Create Network

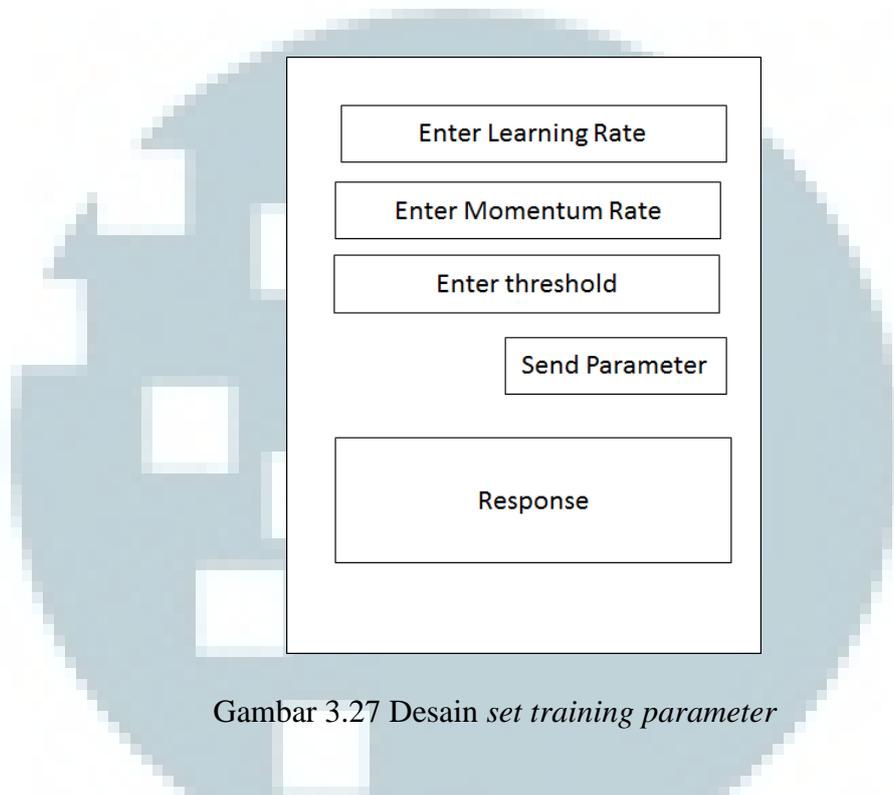


Gambar 3.26 Desain *create network*

Pada *activity* ini, *client* menyusun susunan *hidden layer* yang diinginkan. Jumlah *hidden node* diisi pada *area hidden node* dan dengan menekan tombol *add layer* maka pada *layer list* akan ditambahkan satu *layer* yang berisi jumlah *hidden node* tersebut. Untuk mengulang susunan *hidden layer* dapat menekan tombol *clear network*. Apabila susunan sudah sesuai yang diinginkan dapat mengirim pesan untuk membuat jaringan syaraf tiruan sesuai dengan susunan *hidden layer* yang

diinginkan dengan menekan *create network*. Setelah *client* menerima pesan balasan dari *server*, maka akan ditampilkan di *area response*.

3.5.8 Activity Set Training Parameter



Gambar 3.27 Desain *set training parameter*

Pada *activity* ini, *client* dapat menentukan *learning rate*, *momentum rate*, dan *threshold* pada *area* yang sudah disediakan. Setelah mengisi data-data tersebut dapat menekan tombol *send parameter* untuk mengirimkan *training parameter* tersebut ke *server*. Setelah *client* menerima balasan dari *server*, maka akan ditampilkan di bagian *response*.