



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

METODE DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Dalam penelitian ini, menggunakan metodologi penelitian sebagai berikut:

1. Studi literatur

Pada tahap ini, dilakukan pencarian teori-teori mendasar mengenai pola tulang daun. Selain mencari data untuk objek kajian penelitian, dilakukan pula pencarian berbagai *paper* dan penelitian mengenai *backpropagation* sebagai program utama yang melakukan *pattern recognition* dan *edge detection* sebagai *image processing* untuk memproses citra warna menjadi *greyscale*.

2. Perancangan sistem

Tahap perancangan sistem mencakup pembuatan DFD untuk menganalisis aliran data, *flowchart* untuk penentuan alur program, dan perancangan desain antarmuka untuk merancang tampilan yang baik seperti posisi peletakan *button*, *text*, dan gambar untuk membuat suatu *user interface*.

3. Pemograman sistem

Pemograman aplikasi menggunakan Android Studio sebagai *client* yang digunakan oleh *user* untuk mengirim *image* daun ke *server*. *Software* Eclipse Luna digunakan untuk merancang sistem jaringan saraf tiruan *backpropagation* dengan bahasa pemograman Java sebagai *server* di *laptop* yang menerima masukan *image* dari *client* dan memberikan hasil *output* pola.

4. Pengujian dan survey

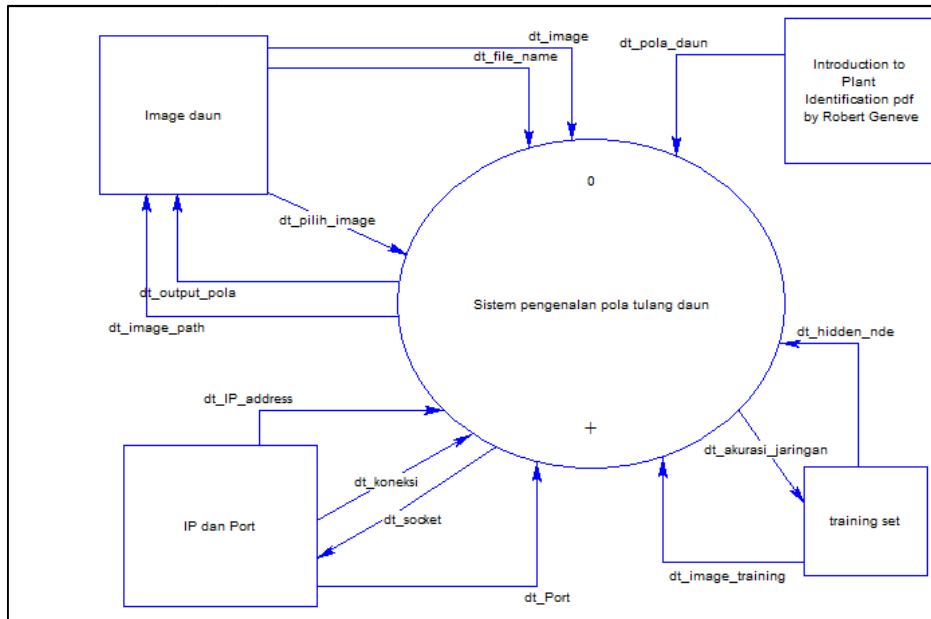
Proses pengujian dilakukan dengan menggunakan data daun ketiga pada sistem jaringan saraf tiruan yang sebelumnya sudah dilatih dengan data daun pertama dan kedua. Dihitung akurasi pengenalan pola masing-masing dan dihitung pula akurasi rata-ratanya. *Survey* dilakukan untuk mengetahui kemudahan dan manfaat dari aplikasi pengenalan pola tulang daun.

3.2 DFD (Data Flow Diagram)

Data flow diagram menggambarkan aliran data yang terjadi dalam sistem. DFD merupakan salah satu teknik yang penting untuk digunakan karena mampu menjelaskan bagaimana *output* dapat dihasilkan dari *input* data yang mengalami serangkaian proses perubahan (Vie, tanpa tahun).

3.2.1 DFD level 0 (Context Diagram)

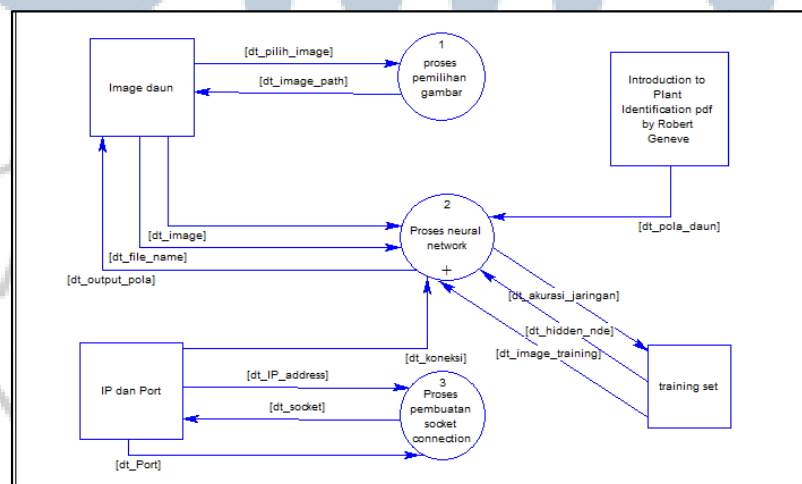
Seperti yang terlihat pada Gambar 3.1 terdapat empat entitas yaitu *image* daun, *IP* dan *port*, sumber data pola tulang daun, dan *training set*. Entitas sumber data pola tulang daun dan *training set* berperan dalam proses *training* data jaringan saraf tiruan yang mengirimkan data pola, data *image* yang akan dilatih dan *hidden node* untuk menentukan bobot yang paling baik. Entitas *image* daun dan entitas *IP* dan *port* berperan pada fase *testing* data yang mengirim data dari perangkat *smartphone*.



Gambar 3.1 DFD level 0 (Context Diagram)

3.2.2 DFD level 1

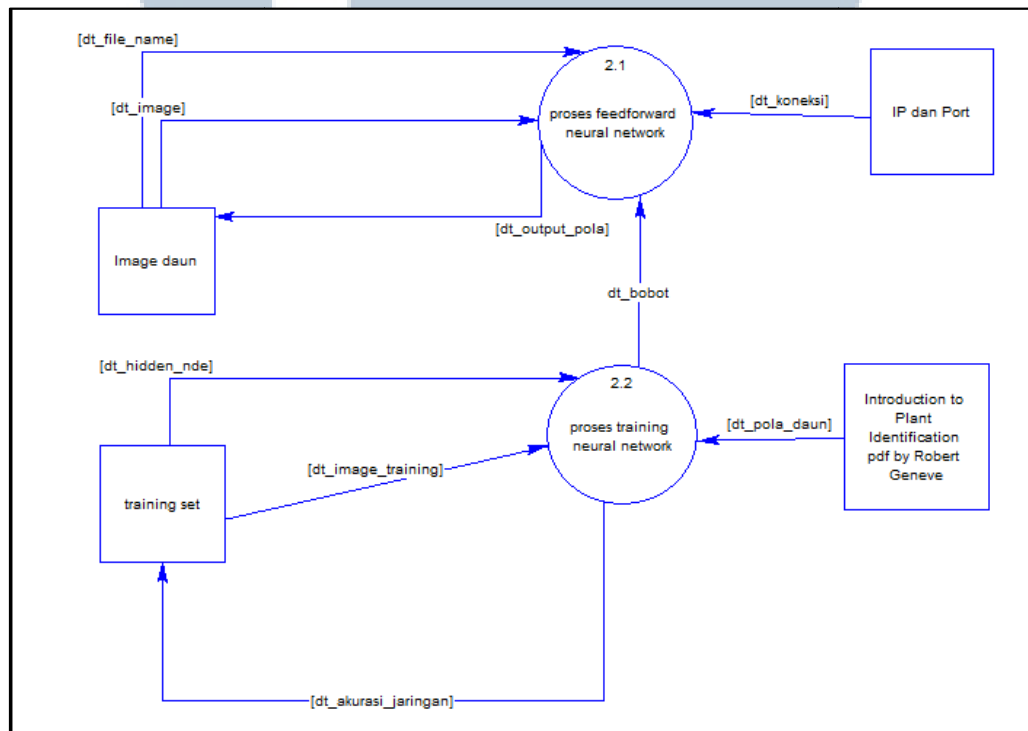
Gambar 3.2 menampilkan alur DFD *level 1* dari sistem pengenalan pola tulang daun. Terdapat tiga buah proses dalam diagram alur data. Data *dt_IP_address* dan *dt_port* digunakan pada proses pembuatan *socket connection* untuk menghasilkan *dt_socket*. Proses pemilihan gambar dilakukan untuk memperoleh *dt_image_path* yaitu lokasi direktori *image* yang nanti akan dikirim ke proses *neural network*.



Gambar 3.2 DFD level 1

3.2.3 DFD level 2

DFD *level 2* terlihat pada Gambar 3.3 yang menjelaskan *core system* dari *neural network*. Entitas *image* daun akan memberikan data *dt_image* yang dikirim lalu dihitung secara *feedforward* oleh *neural network* yang telah mempunyai bobot tetap yang optimal dari *dt_bobot* yang merupakan hasil dari proses *training neural network*. Di sisi lainnya, entitas *training set* dan sumber data pola tulang daun melakukan *training neural network* dari data *dt_image_training*, *dt_hidden_node*, dan *dt_pola_daun* sehingga diperoleh akurasi dalam data *dt_akurasi_jaringan*.



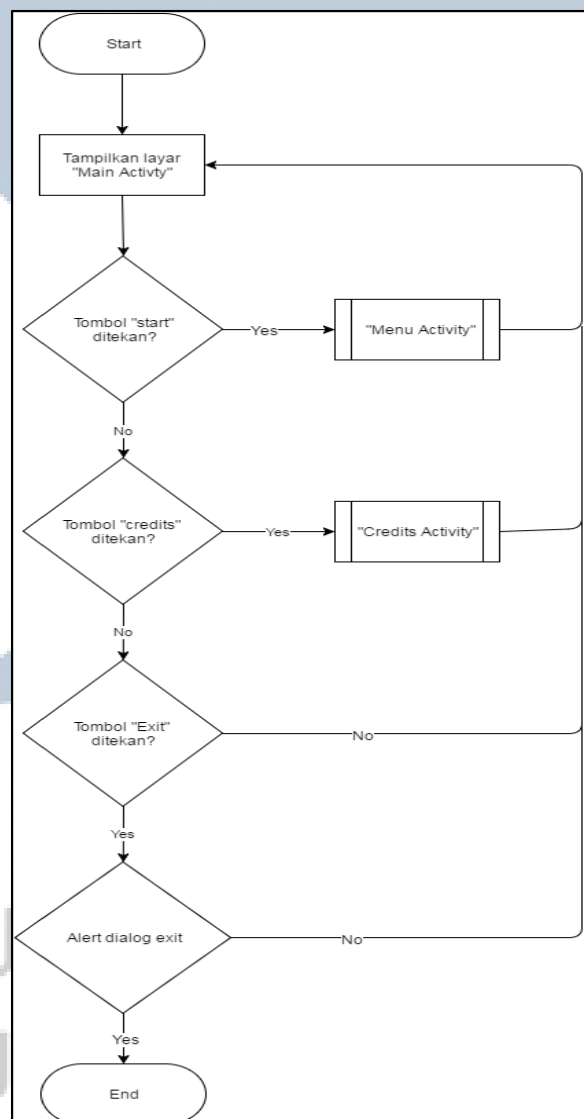
Gambar 3.3 DFD *level 3*

3.3 Flowchart

Flowchart adalah urutan sekuensial yang dikerjakan dalam suatu program. Suatu *flowchart* dapat dengan jelas menggambarkan instruksi dari sebuah program atau *subroutines*.

3.3.1 Flowchart main activity

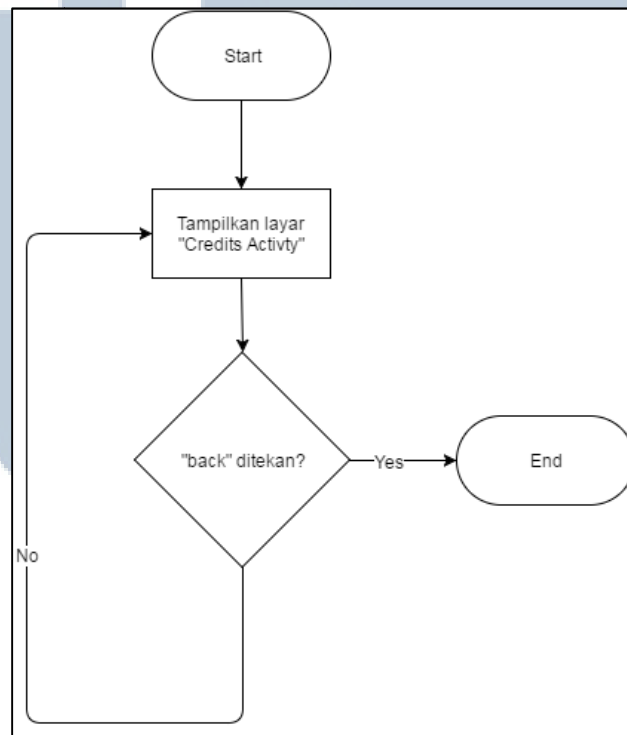
Main activity dimulai dengan menampilkan layar *main activity*. Tombol *start* digunakan untuk mulai masuk ke *menu activity*, tombol *credits* akan mengarahkan ke *credits activity*, dan tombol *exit* yang digunakan untuk keluar program. Saat tombol *exit* ditekan maka sistem akan mengeluarkan *alert dialog*. Jika ya ditekan maka program dihentikan dan jika tidak maka akan kembali ke tampilan awal *main activity*. Gambar 3.4 menjelaskan alur *flowchart main activity*.



Gambar 3.4 Flowchart main activity

3.3.2 Flowchart credits activity

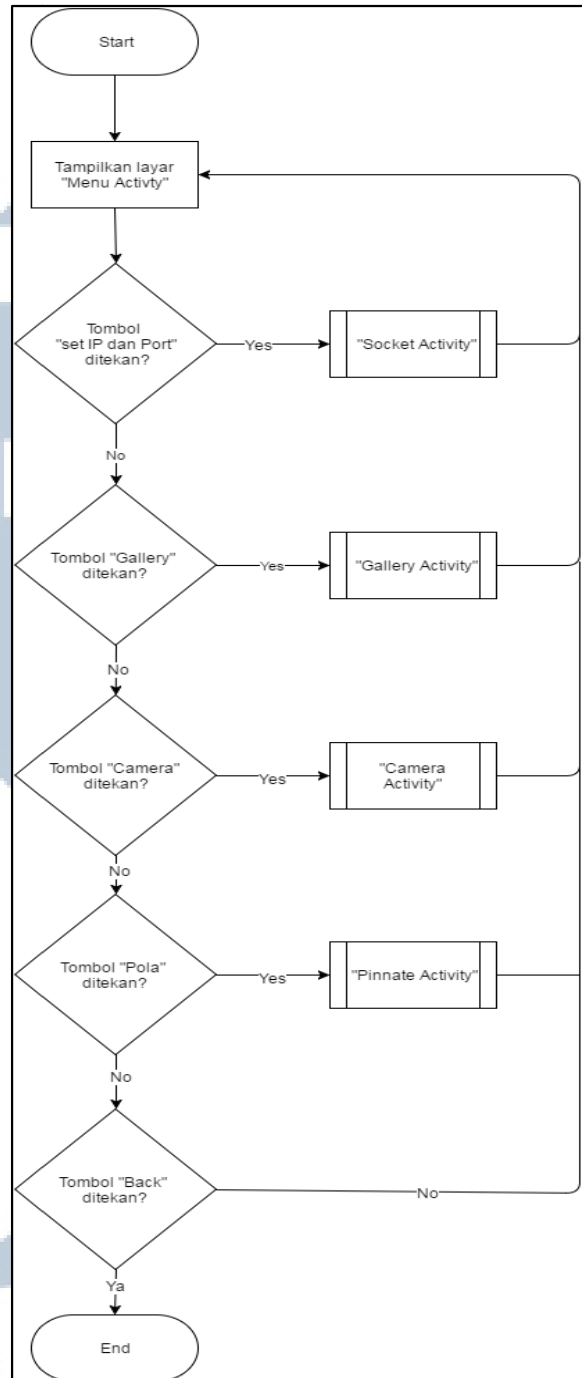
Credits activity dimulai dengan menampilkan layar *credits*, dengan menekan tombol *back* pada perangkat, maka akan dibawa kembali ke *main activity*.



Gambar 3.5 Flowchart credits activity

3.3.3 Flowchart menu activity

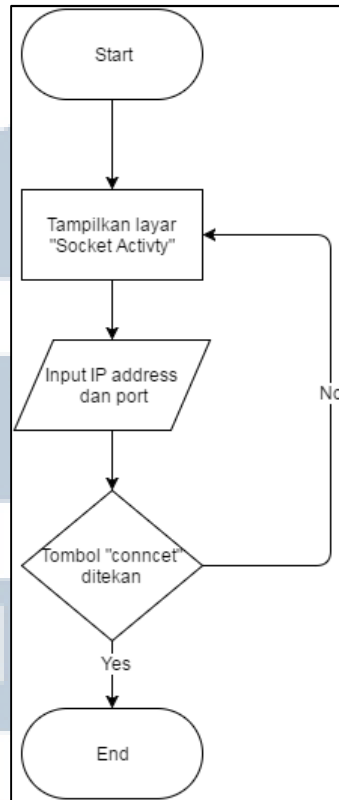
Menu activity dimulai dengan menampilkan layar yang berisi lima tombol. Tombol pertama digunakan untuk mengisi *IP address* dan *port* yang dilakukan pada *socket activity*. Tombol *gallery* dan *camera* memiliki fungsi serupa yaitu untuk mengambil *image* yang akan dijalankan pada layar *activity* masing-masing. Tombol pola digunakan untuk masuk ke pola *activity* yang berisi informasi berbagai macam bentuk pola tulang daun.



Gambar 3.6 *Flowchart menu activity*

3.3.4 Flowchart socket activity

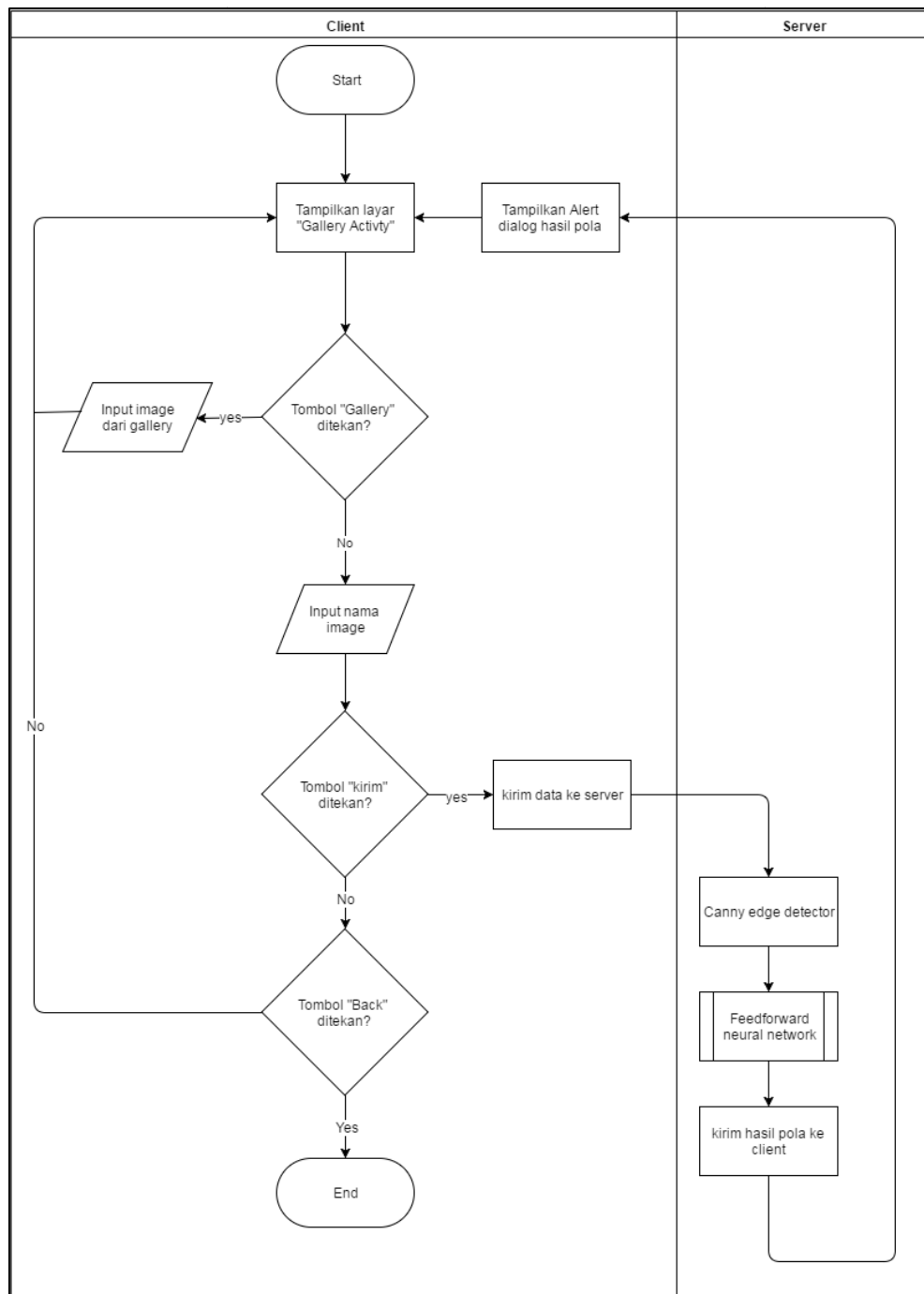
Flowchart socket akan menampilkan *text* yang dapat digunakan *user* untuk memasukkan *IP address* dan *port* untuk membangun *socket connection*. Setelah *user* menekan tombol *connect* maka akan kembali ke halaman *menu activity*.



Gambar 3.7 Flowchart socket activity

3.3.5 Flowchart gallery activity

Seperti yang terlihat pada Gambar 3.8, *flowchart* dibagi ke dalam dua bagian yaitu sisi *server* dan *client*. Di sisi *client*, *user* dapat memilih gambar dari *gallery smartphone* serta memasukkan nama *image*. Tombol kirim digunakan untuk mengirimkan *image* dan *file name* ke sisi *server*. *Server* akan memproses *image* dengan *edge detection*, lalu melakukan perhitungan jaringan saraf tiruan untuk memperoleh *output* pola yang kemudian akan dikirim kembali ke *client*. *Client* yang menerima *output* pola akan menampilkan informasi *output* pola dengan *dialog alert*.

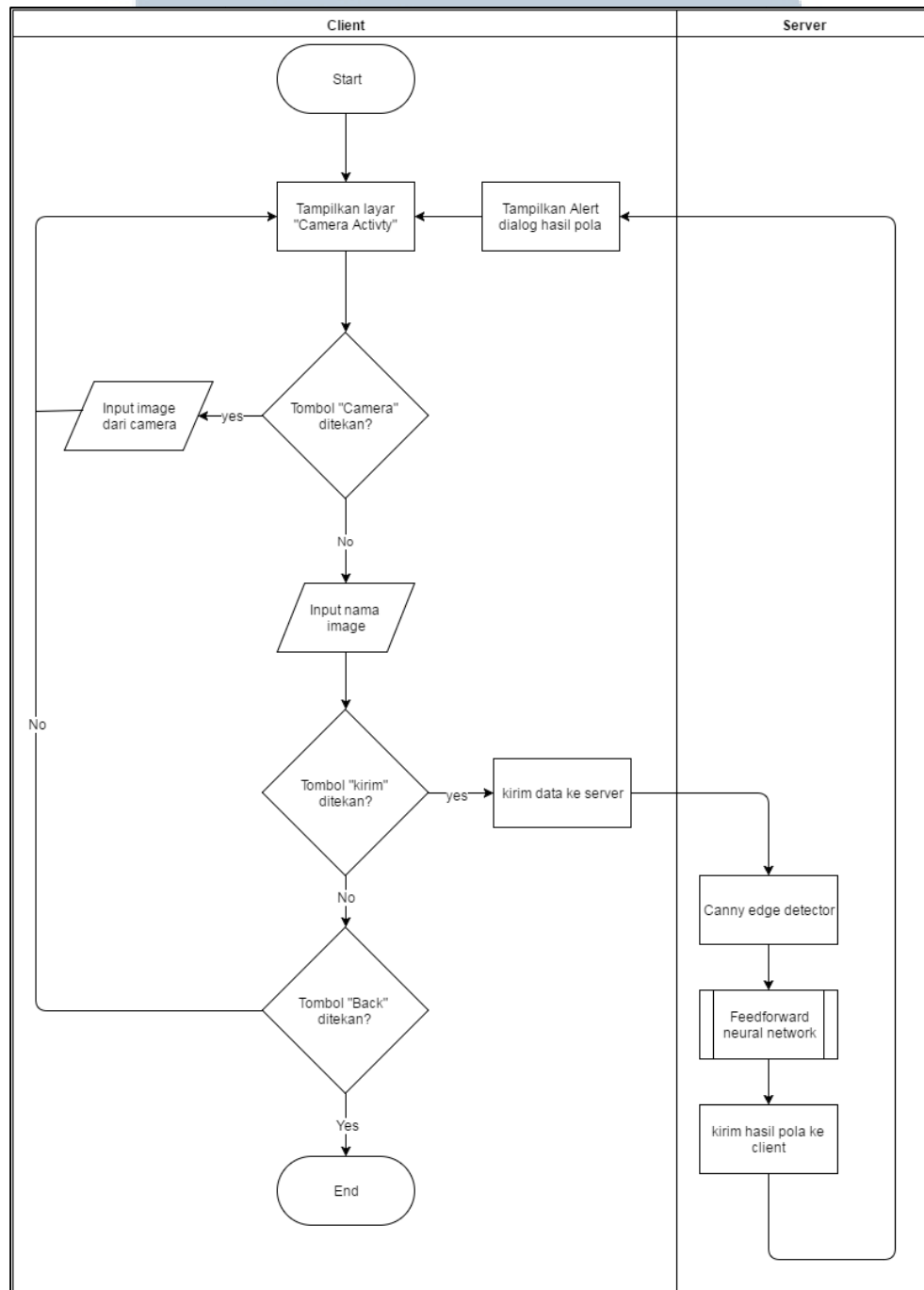


Gambar 3.8 Flowchart gallery activity

3.3.6 Flowchart camera activity

Seperti yang terlihat pada Gambar 3.9, *flowchart* dibagi ke dalam dua bagian yaitu sisi *server* dan *client*. Di sisi *client*, *user* dapat memilih gambar dengan menggunakan *camera* dari perangkat *smartphone* serta memasukkan nama

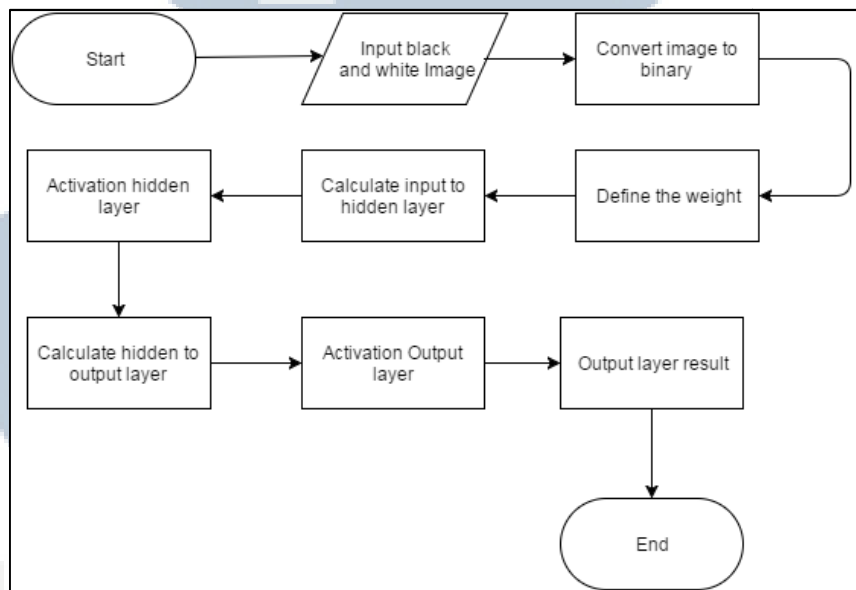
image. Tombol kirim digunakan untuk mengirimkan *image* dan *file name* ke sisi *server*. *Server* akan memproses *image* dengan *edge detection*, lalu melakukan perhitungan jaringan saraf tiruan untuk memperoleh *output* pola yang kemudian akan dikirim kembali ke *client*. *Client* yang menerima *output* pola akan menampilkan informasi *output* pola dengan *dialog alert*.



Gambar 3.9 Flowchart camera

3.3.7 Flowchart feedforward

Flowchart feedforward menjelaskan alur logika *feedforward* dari algoritma jaringan saraf tiruan. Proses *feedforward* digunakan untuk memperoleh hasil output pola dari *image* daun. *Image* dari proses *image processing* dengan *edge detection* yang berupa *image grayscale* akan dilakukan konversi ke dalam *bit biner*. Proses *feedforward* dimulai dengan menggunakan bobot yang sudah ditetapkan lalu menghitung nilai tiap *node* dari *input layer* ke *hidden layer* yang kemudian akan diaktifkan dengan menggunakan fungsi aktivasi. Nilai dari *hidden* ke *output layer* juga dihitung dan digunakan fungsi aktivasi untuk menghitung *output node* pola daun pada *output layer*. *Output* berupa nilai dari masing-masing pola.

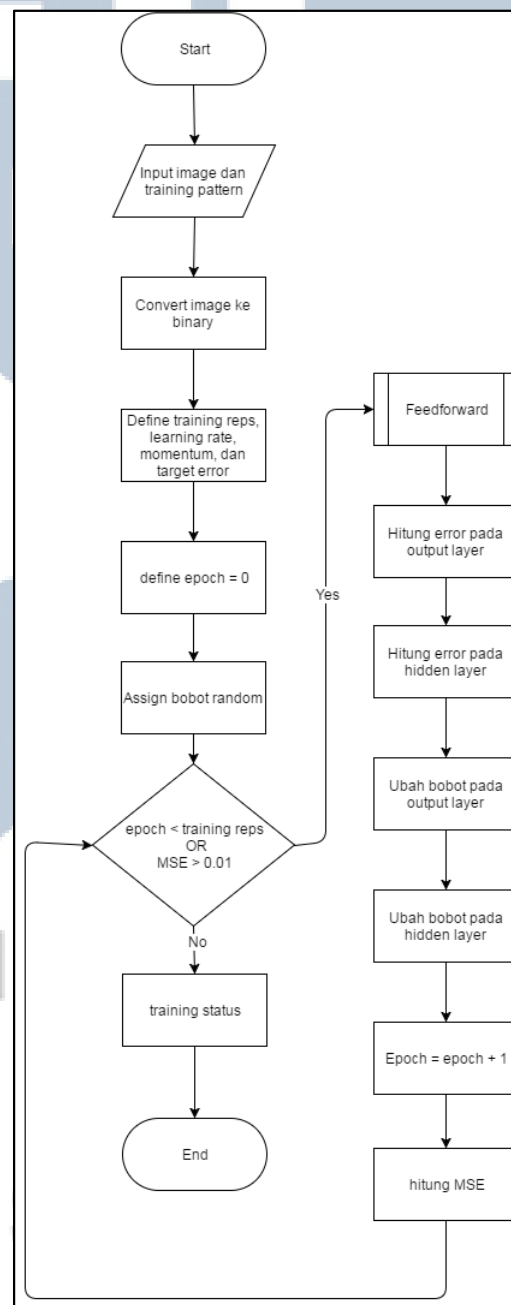


Gambar 3.10 *Flowchart feedforward*

3.3.8 Flowchart backpropagation

Flowchart backpropagation menjelaskan alur mundur dari jaringan saraf tiruan untuk memperoleh nilai bobot yang optimal yang nanti akan digunakan pada *testing* data. Langkah awal dimulai dengan memasukkan *set training* data

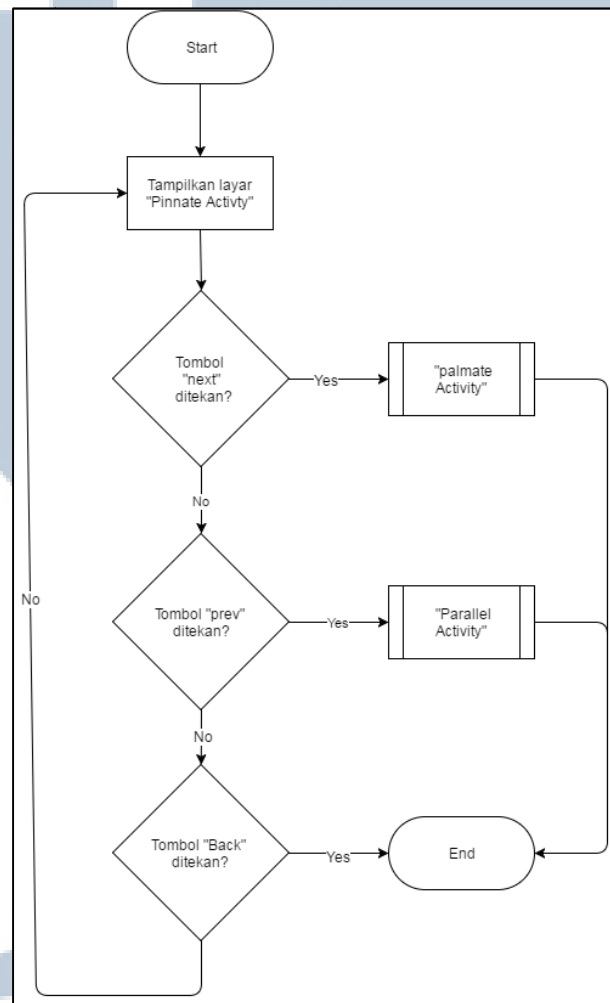
yang terdiri dari *image* dan *target pattern* masing-masing *image*. *Image* yang dimasukkan akan diubah ke bentuk *bit*. Lalu dilakukan inisialisasi atribut *backpropagation*. Proses akan dihentikan jika *epoch* telah mencapai jumlah maksimal repitisi training atau nilai MSE lebih kecil dari target *error* yang ditetapkan. Selama kondisi belum tercapai maka sistem akan terus melakukan perubahan bobot di masing-masing *layer*.



Gambar 3.11 Flowchart backpropagation

3.3.9 Flowchart pinnate activity

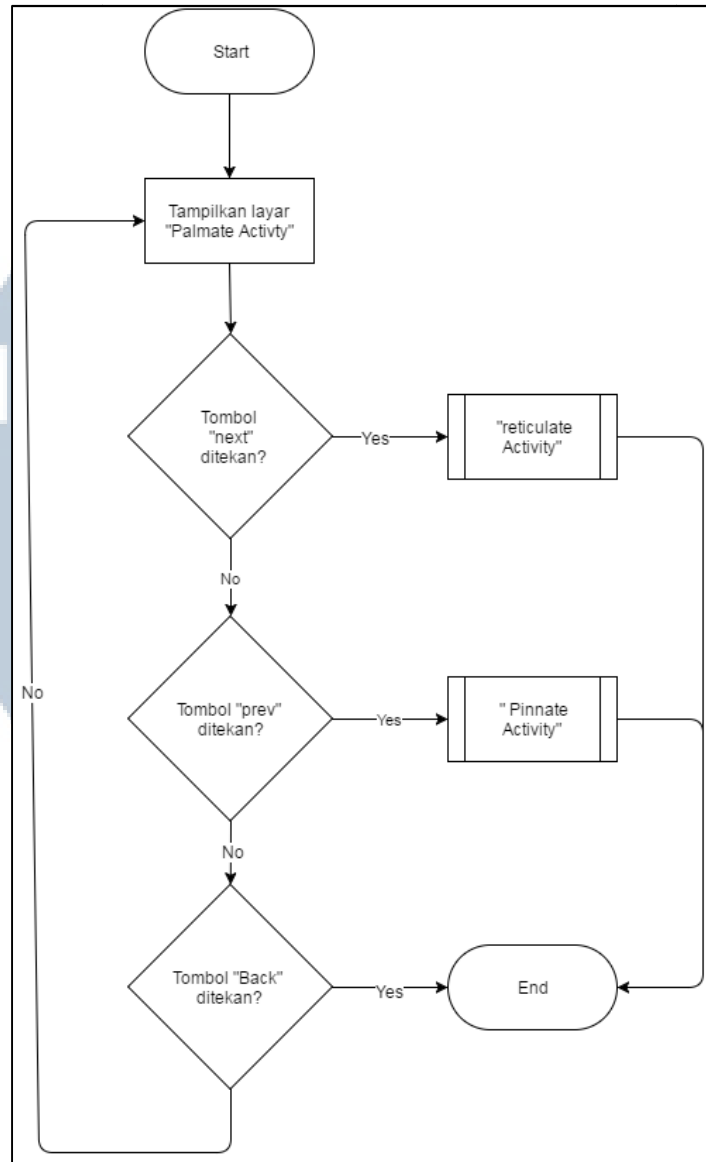
Flowchart pinnate activity akan menampilkan layar yang berisi informasi mengenai tipe pola *pinnate*. Tombol *next* dan *prev* digunakan untuk menampilkan *activity* pola lainnya. *Back* untuk kembali ke *menu activity*.



Gambar 3.12 *Flowchart pinnate activity*

3.3.10 Flowchart palmate activity

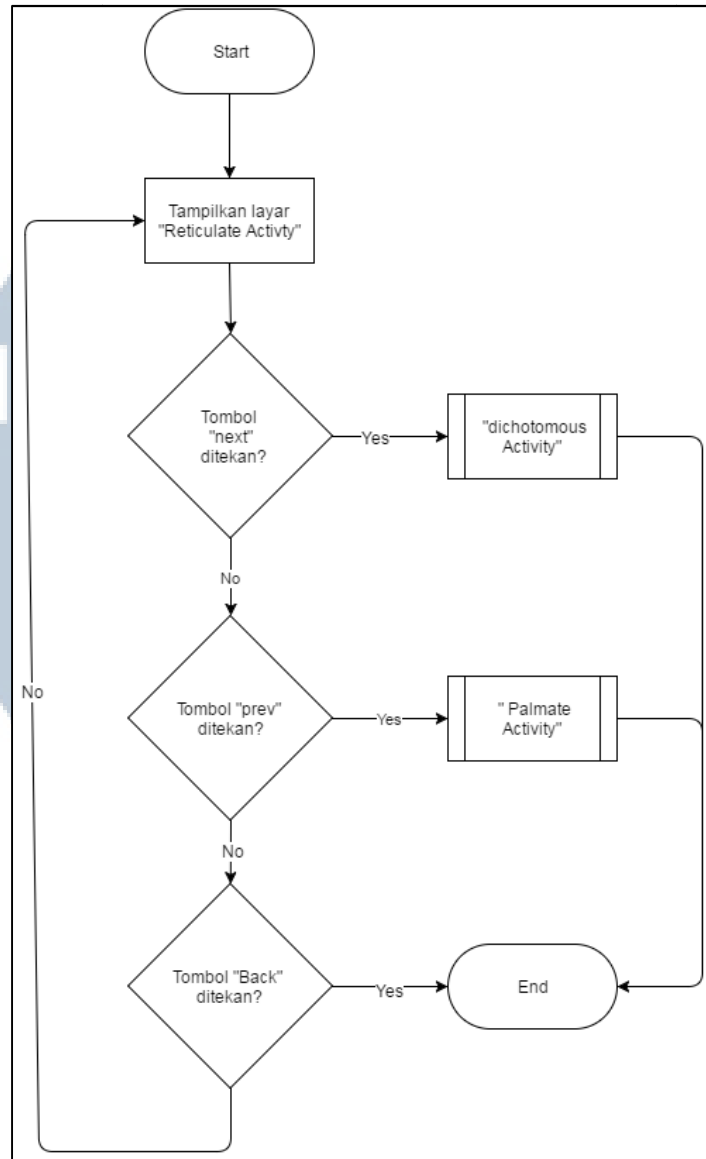
Flowchart palmate activity akan menampilkan layar yang berisi informasi mengenai tipe pola *palmate*. Tombol *next* dan *prev* digunakan untuk menampilkan *activity* pola lainnya. *Back* untuk kembali ke *menu activity*.



Gambar 3.13 *Flowchart palmate activity*

3.3.11 Flowchart reticulate activity

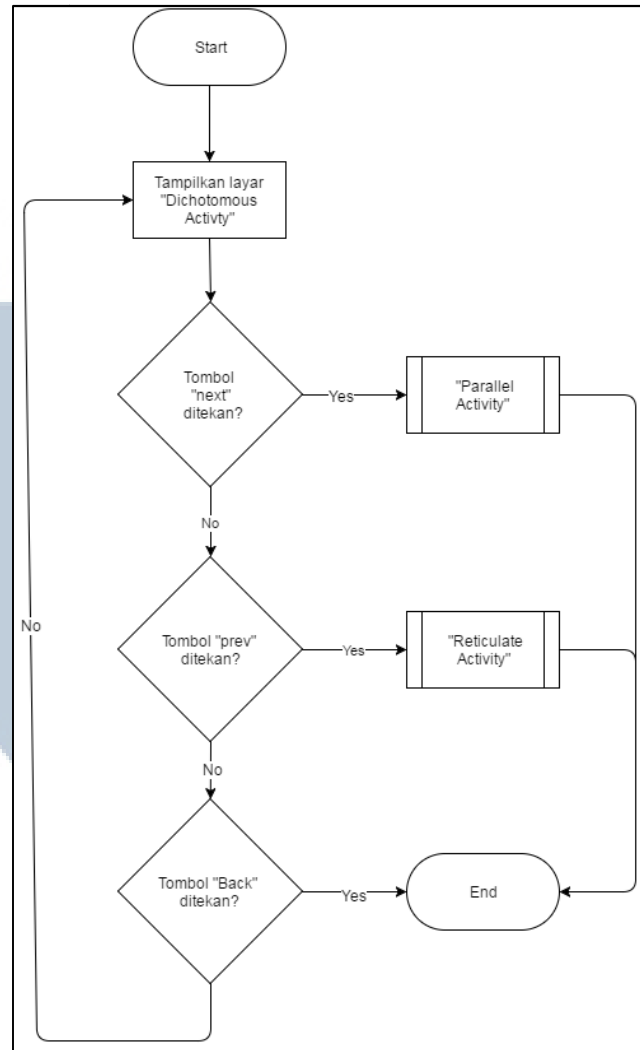
Flowchart reticulate activity akan menampilkan layar yang berisi informasi mengenai tipe pola *reticulate*. Tombol *next* dan *prev* digunakan untuk menampilkan *activity* pola lainnya. *Back* untuk kembali ke *menu activity*.



Gambar 3.14 *Flowchart reticulate activity*

3.3.12 Flowchart dichotomous activity

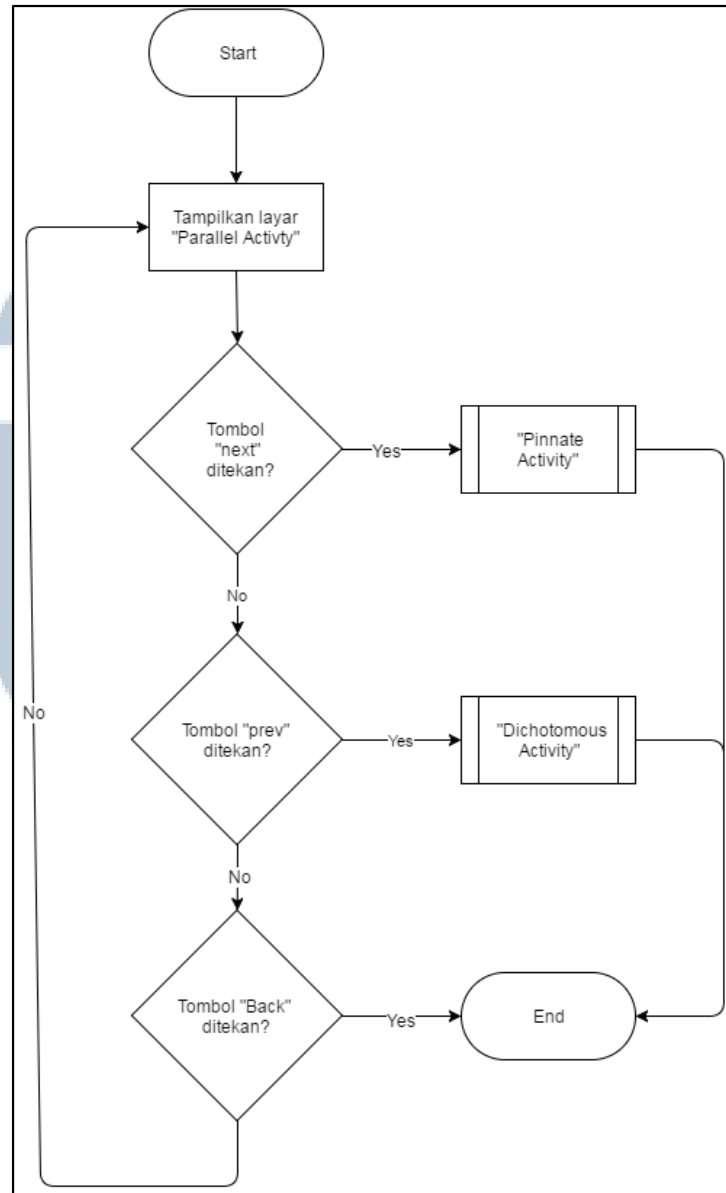
Flowchart dichotomous akan menampilkan layar yang berisi informasi mengenai tipe pola *dichotomous*. Tombol *next* dan *prev* digunakan untuk menampilkan *activity* pola lainnya. *Back* untuk kembali ke *menu activity*.



Gambar 3.15 *Flowchart dichotomous activity*

3.3.13 Flowchart parallel activity

Flowchart parallel akan menampilkan layar yang berisi informasi mengenai tipe pola *parallel*. Tombol *next* dan *prev* digunakan untuk menampilkan *activity* pola lainnya. *Back* untuk kembali ke *menu activity*.

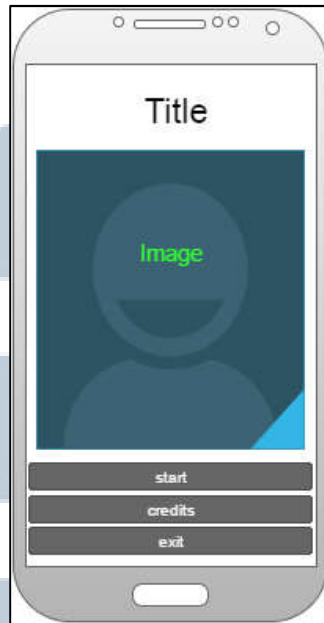


Gambar 3.16 Flowchart parallel activity

3.4 Desain antar muka

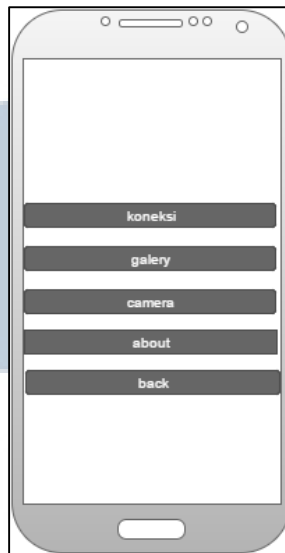
Sebelum melakukan pembuatan aplikasi *client* berbasis android, perlu disiapkan sebuah tampilan yang baik sehingga *user interface* tersebut dapat dengan mudah digunakan oleh *user*.

Gambar 3.17 menunjukkan halaman utama saat program dibuka, yang terdiri dari tiga tombol utama dengan judul di atas dan di tengah antara judul dan *button* akan digunakan *image* yang menjadi *icon* utama dari *program*.



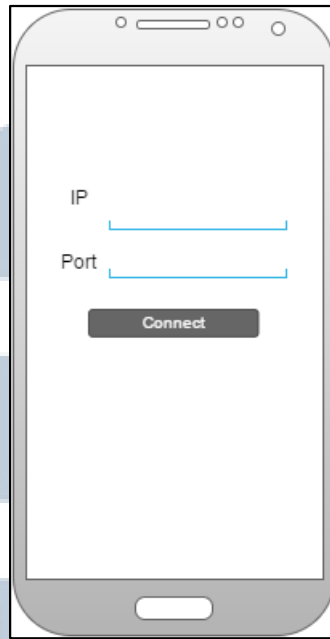
Gambar 3.17 *Mockup main*

Gambar 3.18 menunjukkan *menu activity* yang terdiri atas berbagai tombol yang digunakan dengan fungsionalitas dan fitur yang berbeda. Tampilan *button* di tengah sehingga *button* mudah digunakan oleh *user*.



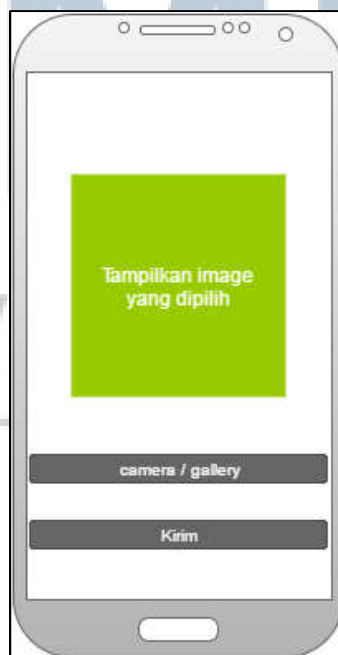
Gambar 3.18 *Mockup menu*

Gambar 3.19 Menunjukkan tampilan saat *user* akan memasukkan *IP* dan *port*. *User* akan diarahkan untuk mengisi kolom *IP* dan *port*. Tombol *connect* digunakan untuk membangun jaringan *socket*.



Gambar 3.19 *Mockup socket*

Gambar 3.20 menunjukkan tampilan *gallery* dan *image*, keduanya akan dibuat dalam dua layar yang berbeda, namun secara tampilan hampir serupa hanya dengan fitur yang berbeda saja. Saat tombol *camera* atau *gallery* ditekan dan dipilih, maka *image* akan ditampilkan terlebih dahulu sebelum *user* dapat mengirimkan gambar dengan menekan tombol kirim.



Gambar 3.20 *Mockup gallery dan camera*

Gambar 3.21 menunjukkan tampilan informasi pola daun. Tombol *next* dan *prev* digunakan untuk pergantian layar antara satu layar informasi dengan layar informasi pola daun lainnya.



Gambar 3.21 *Mockup information*

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA