

## 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, teori-teori yang berhubungan dengan aplikasi yang akan dibuat, seperti perkembangan *dough* saat fermentasi, perhitungan perbedaan *pixel* dalam gambar bitmap, pengecekan gelembung pada *dough*, Naïve Bayes Classifier untuk penentuan keputusan tingkat kesempurnaan *dough*.

##### 2. Perancangan Sistem

Merancang aplikasi yang akan dibuat dalam bentuk *flowchart*, *mockup user interface*.

##### 3. Pengumpulan Data

Data dikumpulkan dengan cara memantau proses perkembangan *dough* dalam tahap fermentasi dengan melihat perbesaran dari *dough* dan melihat gelembung yang terdapat pada *dough*.

##### 4. Pemrograman Sistem

Pemrograman aplikasi dilakukan dengan Android Studio dengan bahasa pemrograman Java. Pada tahap ini dilakukan implementasi algoritma Naïve Bayes dan metode *bitmap image comparison* ke dalam alat pendeteksi.

*Bitmap image comparison* digunakan untuk menentukan adanya perubahan besaran dari *dough* dan foto diambil untuk pengecekan gelembung pada

*dough*. Dari hasil pengecekan besaran dan gelembung akan diklasifikasi menggunakan algoritma Naïve Bayes bahwa *dough* sudah sempurna atau belum.

#### 5. Pengujian (*Testing*)

Proses pengujian aplikasi dengan penggunaan *smartphone* android Samsung Galaxy Note 3. Proses pengujian dilakukan dengan mengambil gambar dari *dough* sebanyak tiga kali, yaitu ketika awal, saat sudah berkembang tanpa *blitz*, dan saat sudah berkembang dengan *blitz*.

#### 6. Evaluasi

Evaluasi dilakukan dengan proses analisis berdasarkan data yang diuji untuk mengetahui akurasi aplikasi dalam memutuskan bahwa *dough* sudah sempurna atau belum.

### 3.2 Sistem Pengecekan

Proses pertama untuk pengecekan *resting* dari *dough* menggunakan metode *bitmap image comparison* dimana akan membandingkan antara foto awal *dough* dan foto *dough* ketika ingin dicek dengan membandingkan warna di titik yang sama pada kedua foto. Dalam proses pengecekan perbesaran, untuk membedakan antara *dough* dan latar belakang digunakan kondisi warna *red* atau *green* harus lebih besar dari warna *blue* sebanyak dua puluh, kondisi ini dibuat karena warna hitam yang digunakan pada latar belakang memiliki warna *red*, *green*, dan *blue* yang saling berdekatan.

Proses pengecekan kedua dengan mengambil gambar *dough* dari depan menggunakan bantuan cahaya *blitz* dari kamera untuk melihat gelembung-gelembung yang terdapat pada *dough*, dengan menggunakan cahaya dapat terlihat

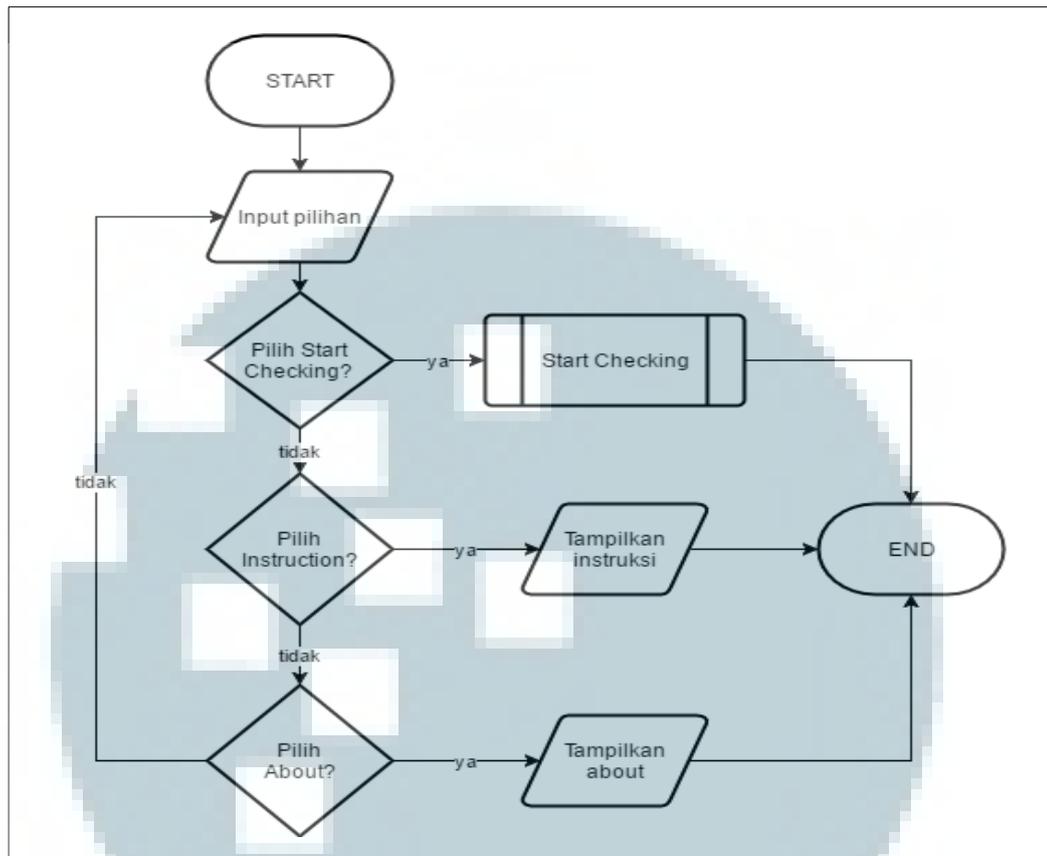
gelembung-gelembung memiliki perbedaan warna karena tingkat kedalaman lubang tersebut. Dalam proses pengecekan gelembung akan menghitung titik tengah pada gambar terlebih dahulu, setelah mendapatkan titik tengah akan dibuat daerah untuk pengecekan. Pengecekan dilakukan dengan *looping* di dalam daerah pengecekan dan mengambil warna pixel yang sudah di normalisasi menjadi *grayscale*, dan setiap pixel terendah akan dicatat untuk dibandingkan dengan pixel sebelahnya dalam daerah pengecekan. Jika terdapat pixel yang lebih kecil sebesar lima belas dari pixel di kiri dan di kanan titik tersebut, maka *counter* gelembung akan ditambahkan dan adonan dianggap bergelembung.

Hasil yang didapat dari kedua proses tersebut akan diklasifikasi menggunakan algoritma Naïve Bayes yang sangat baik digunakan untuk pengklasifikasian dengan menggunakan teorema Bayes sebagai dasarnya dan ditambahkan asumsi atribut independen Naïve, sehingga mendapatkan hasil yang optimal dan lebih akurat.

### **3.3 Perancangan Sistem**

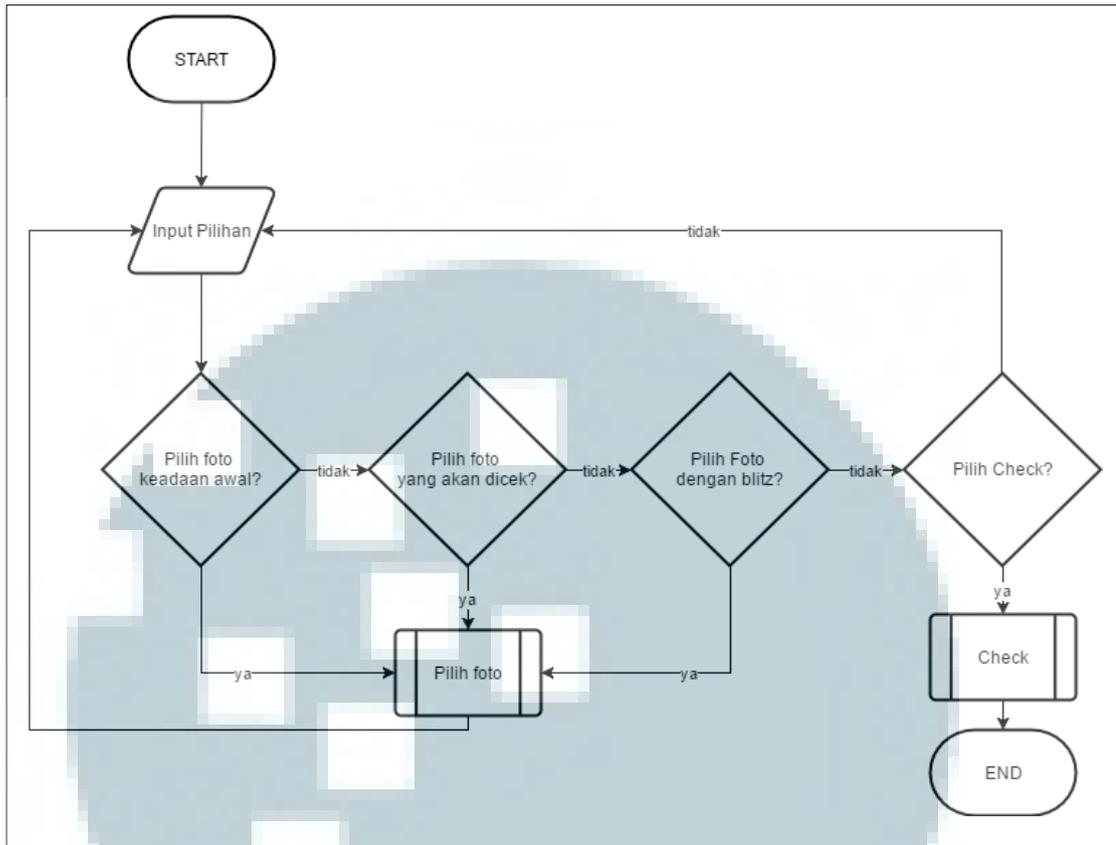
Sebelum tahapan pemrograman sistem dilakukan, maka dibuatlah *flowchart*, *Data Flow Diagram* (DFD), struktur tabel, dan *mockup* yang menjadi landasan pemrograman sistem secara keseluruhan.

### 3.3.1 Flowchart



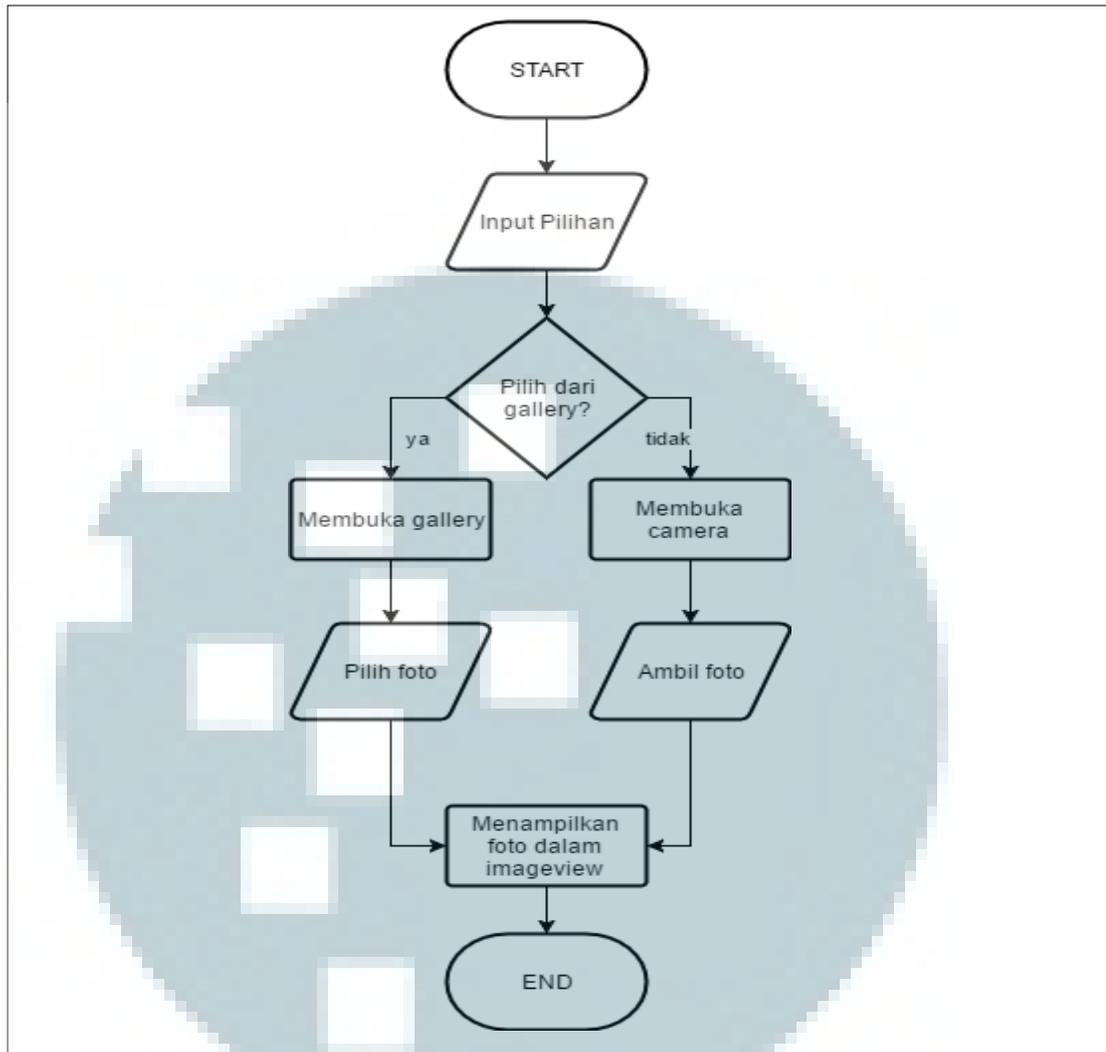
Gambar 3.1 Flowchart Menu Utama Aplikasi

Perancangan yang dibuat untuk aplikasi yang dibangun seperti pada Gambar 3.1. Aplikasi yang dibuat memiliki tampilan awal dengan tiga *button*, yaitu “Start checking”, “Instruction”, dan “About”. *Instruction* berisikan cara penggunaan aplikasi, *About* berisikan tentang pengembang aplikasi, dan *Start checking* untuk memulai proses pengecekan.



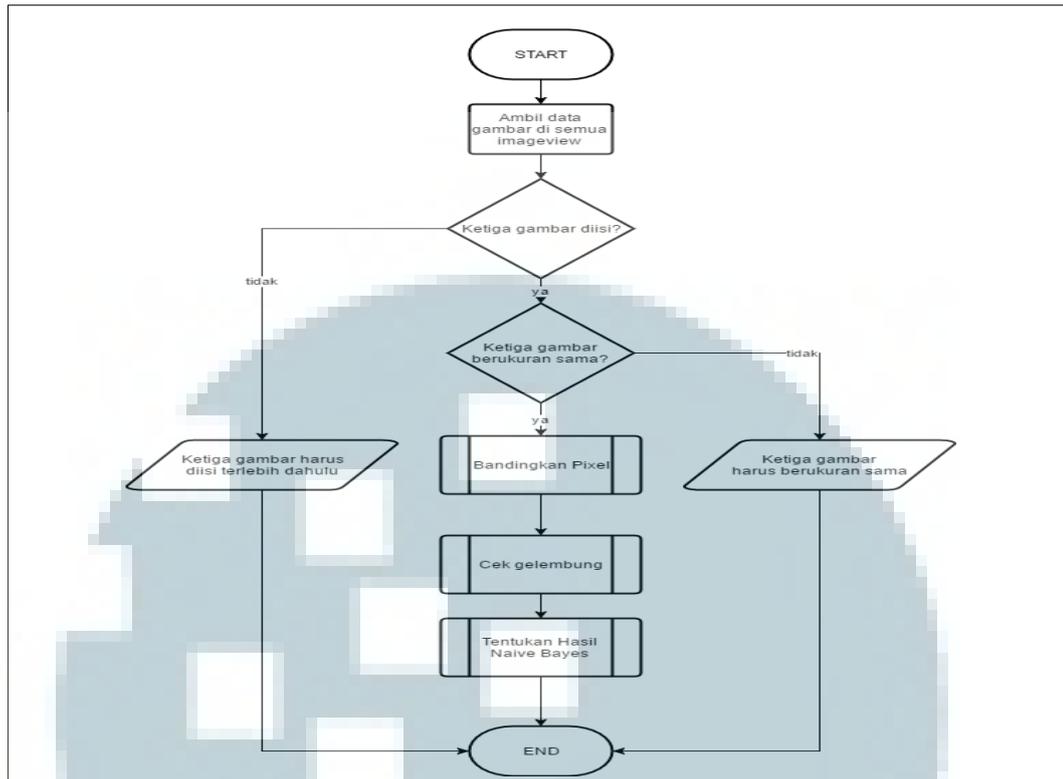
Gambar 3.2 Flowchart Start Checking

Proses *start checking* yang dibuat memiliki empat *button*, yaitu “Foto keadaan awal”, “Foto yang akan dicek”, “Foto dengan blitz”, dan “Check”. Ketika memilih *button* “Foto keadaan awal”, “Foto yang akan dicek”, dan “Foto dengan blitz” akan muncul pesan secara *pop up* yang akan memberikan dua pilihan yaitu mengambil gambar dari *gallery* atau secara langsung mengambil foto untuk dilakukan pengujian. Ketika memilih mengambil gambar dari *gallery* maka akan membuka *gallery* foto dalam *smartphone* dan dapat langsung memilih gambar, dan saat memilih mengambil foto akan langsung masuk ke aplikasi dasar *smartphone* yaitu kamera untuk mengambil foto *dough*. Jika memilih *button* “*Check*” dilakukan pengecekan terhadap ketiga gambar.



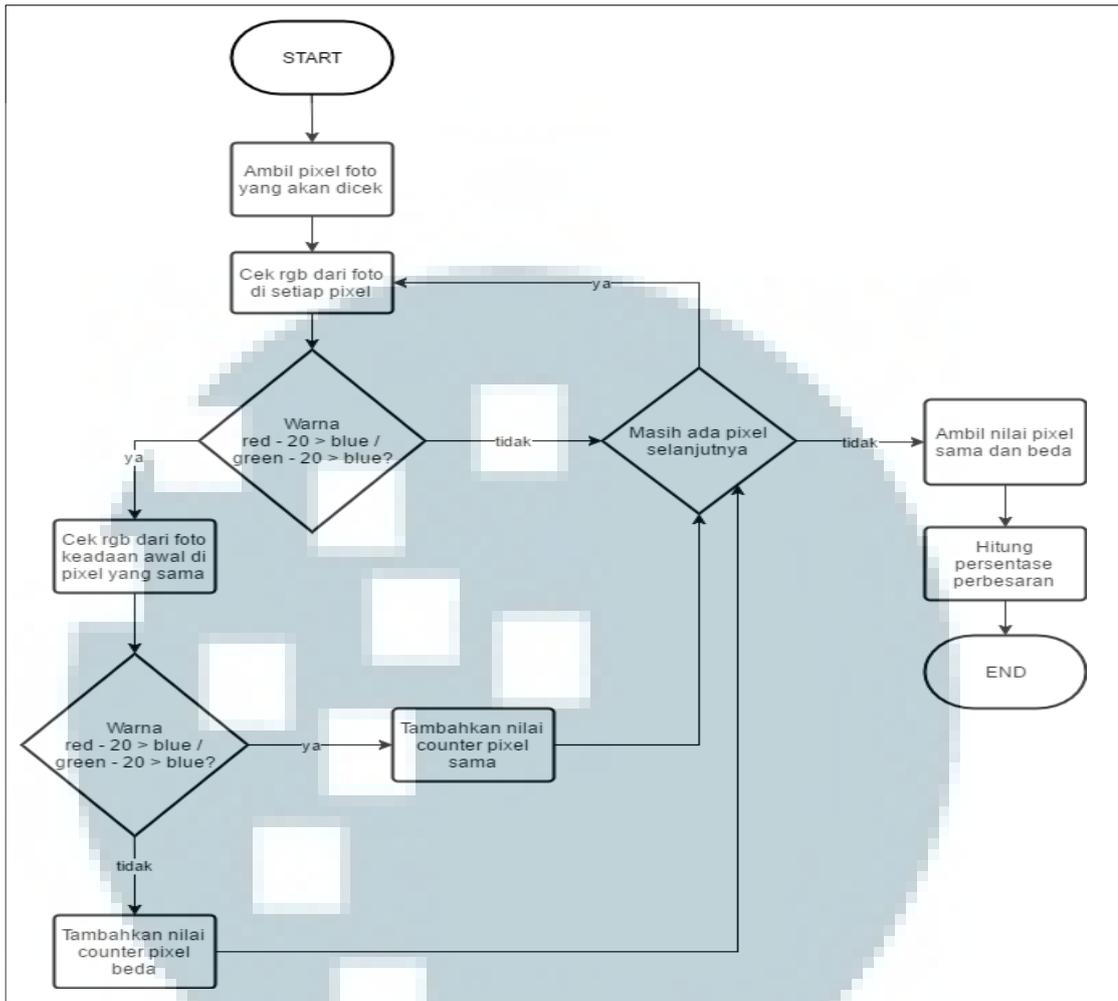
Gambar 3.3 *Flowchart* Pilih Foto

Gambar 3.3 merupakan proses pemilihan Foto ketika *button* “Foto keadaan awal”, “Foto yang akan dicek”, “Foto dengan blitz” ditekan. Muncul *pop up message* berupa pilihan untuk memilih foto dari *gallery* atau mengambil foto dari kamera. Foto yang dipilih atau diambil menggunakan kamera akan ditampilkan ke dalam *imageview* sesuai *button* yang dipilih.



Gambar 3.4 *Flowchart Check*

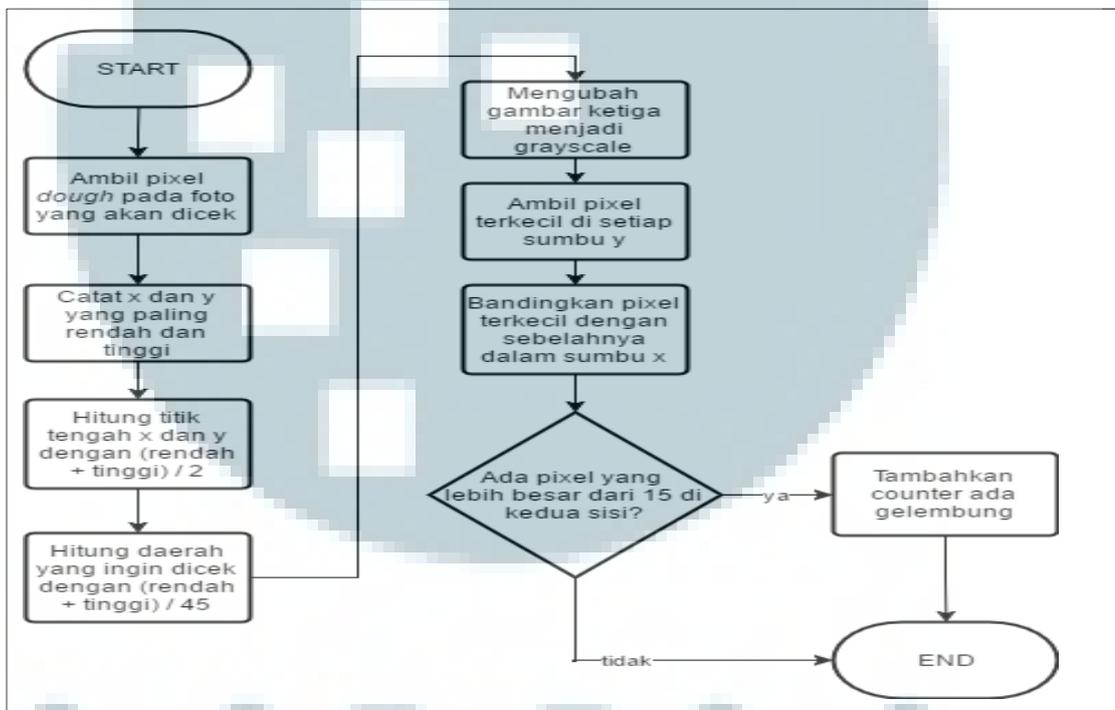
Pada Gambar 3.4 merupakan proses ketika *button* “Check” ditekan. Ketiga data yang sudah dipilih sesuai dalam *imageview* akan diambil dan dicek. Ketiga foto sudah diisi atau belum, jika foto belum diisi akan muncul *pop up message* yang memberitahu bahwa ketiga foto harus diisi untuk melanjutkan pengecekan, jika ketiga foto sudah diisi akan melanjutkan pengecekan dimensi dari ketiga foto. Foto yang diambil harus berukuran sama untuk melakukan pengecekan *pixel*. Jika terdapat dimensi yang berbeda akan muncul *pop up message* yang memberitahu bahwa ketiga foto harus berukuran sama, dan akan lanjut untuk membandingkan *pixel* jika ketiga foto sudah berukuran sama. Setelah membandingkan *pixel* untuk mengecek perbesaran gambar, terdapat pengecekan gelembung yang terdapat dalam *dough* dan kedua hasil pengecekan akan diklasifikasikan hasilnya menurut Naïve Bayes.



Gambar 3.5 Flowchart Bandingkan Pixel

Proses membandingkan *pixel* dapat dilihat pada Gambar 3.5. Pertama mengambil *pixel* foto yang akan dicek dan melakukan pengecekan warna di setiap *pixel*. Jika *red* atau *green* lebih besar dari *blue* sebanyak dua puluh maka akan melakukan pengecekan yang sama di *pixel* yang sama pada foto keadaan awal. Jika pada kondisi awal *red* dan *green* tidak lebih besar dari *blue* sebanyak dua puluh maka tambah *pixel* beda ditambah satu, sebaliknya jika lebih besar maka *pixel* sama ditambah satu. Apabila pada pengecekan foto yang akan dicek *red* dan *green* tidak lebih besar dari *blue* sebanyak dua puluh maka proses pengecekan akan dilanjutkan ke *pixel* berikutnya, jika tidak terdapat *pixel* selanjutnya proses pengecekan dilanjutkan dengan menghitung perbesaran *dough* yaitu *pixel* yang

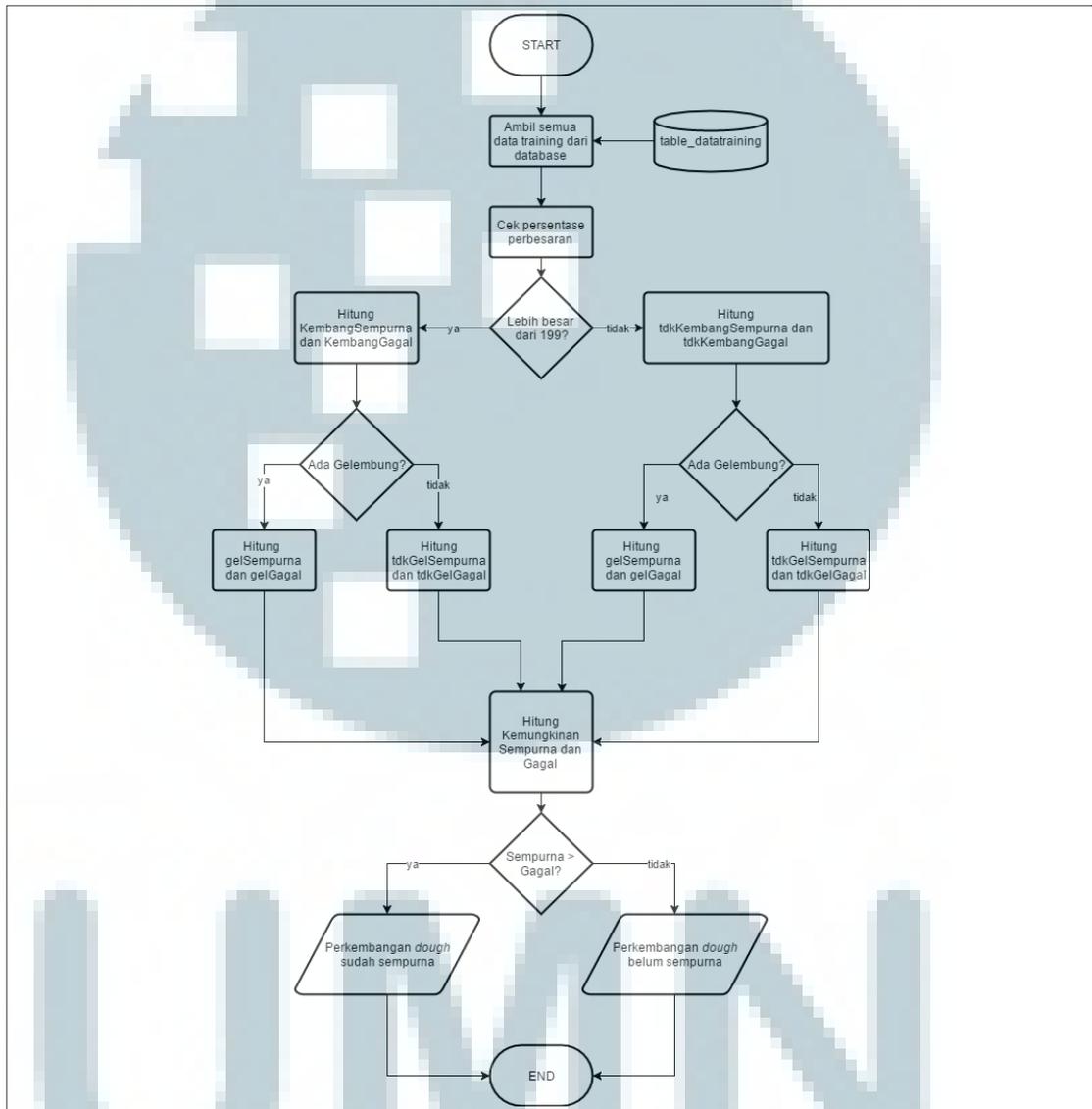
sama ditambah *pixel* yang beda (total) dibagi oleh *pixel* yang sama dan dikali seratus untuk mendapatkan persentase perbesaran gambar. Persentase perbesaran digunakan untuk menentukan perkembangan *dough*, jika persentase mencapai dua ratus persen maka perkembangan *dough* sudah dua kali lipat dan dapat dikatakan sudah mengembang sempurna seperti menurut penelitian Bernstein (2014), sebaliknya jika di bawah dua ratus persen maka adonan belum mengembang sempurna.



Gambar 3.6 *Flowchart* Cek Gelembung

Pada Gambar 3.6 merupakan pengecekan gelembung yang diproses setelah membandingkan *pixel*. Pengecekan dimulai dengan mengambil *pixel* pada foto yang akan dicek dan mengambil x dan y yang paling rendah dan paling tinggi. Menggunakan data x dan y paling rendah dan paling tinggi untuk mendapatkan titik tengah dengan menambah nilai rendah dan tinggi lalu dibagi dua. Setelah mendapatkan titik tengah, hitung daerah yang menjadi acuan untuk pengecekan.

Setelah mendapat daerah yang akan dicek, ubah foto ketiga yaitu foto dengan *blitz* menjadi *grayscale* dan ambil *pixel* terkecil dengan menghitung setiap *pixel* di dalam daerah acuan. *Pixel* terkecil disimpan untuk dibandingkan dengan *pixel* di sumbu y yang sama. Jika terdapat *pixel* yang lebih besar sebanyak lima belas di kedua sisi maka foto memiliki gelembung.



Gambar 3.7 Flowchart Tentukan Hasil Naïve Bayes

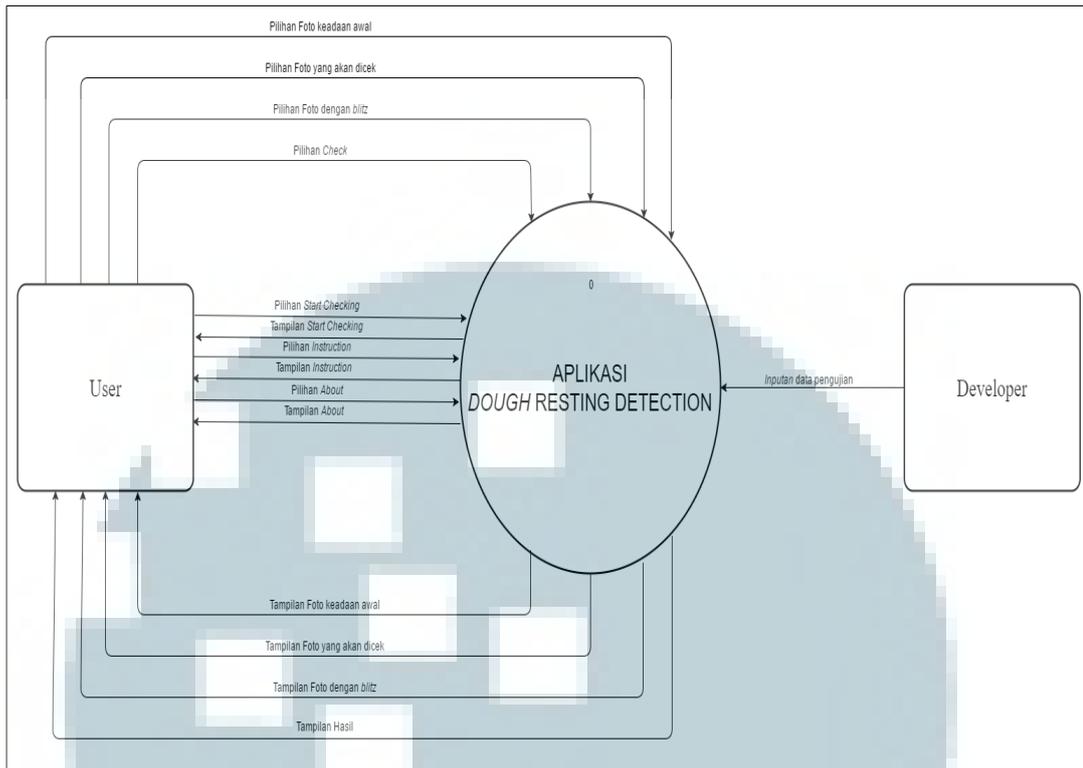
Proses pengecekan yang terakhir yaitu untuk klasifikasi data menggunakan teorema Naïve Bayes. Seperti pada Gambar 3.7 proses pengecekan bermula dengan mengambil semua data dari *database* lalu mengambil data persentase

perkembangan yang sudah dicek sebelumnya. Jika di atas dua ratus maka pengecekan lanjut dengan variabel kembang, dan jika di bawah dua ratus akan menggunakan variabel tdkKembang. Setelah menghitung peluang sempurna dan gagal proses dilanjutkan dengan mengambil data pengecekan gelembung. Setelah mendapatkan semua data peluang independen lalu menghitung peluang keseluruhan sempurna dan gagal. Peluang sempurna dan gagal digunakan untuk menentukan apakah *dough* sudah sempurna atau belum dengan melihat peluang mana yang lebih besar.

### 3.3.2 Data Flow Diagram (DFD)

DFD merupakan sebuah diagram yang menggambarkan aliran data pada sebuah aplikasi. Dengan DFD dapat terlihat jelas *input* atau *output* data dalam aplikasi yang dibangun memasuki proses apa saja. Berikut adalah diagram konteks dapat dilihat pada Gambar 3.8.

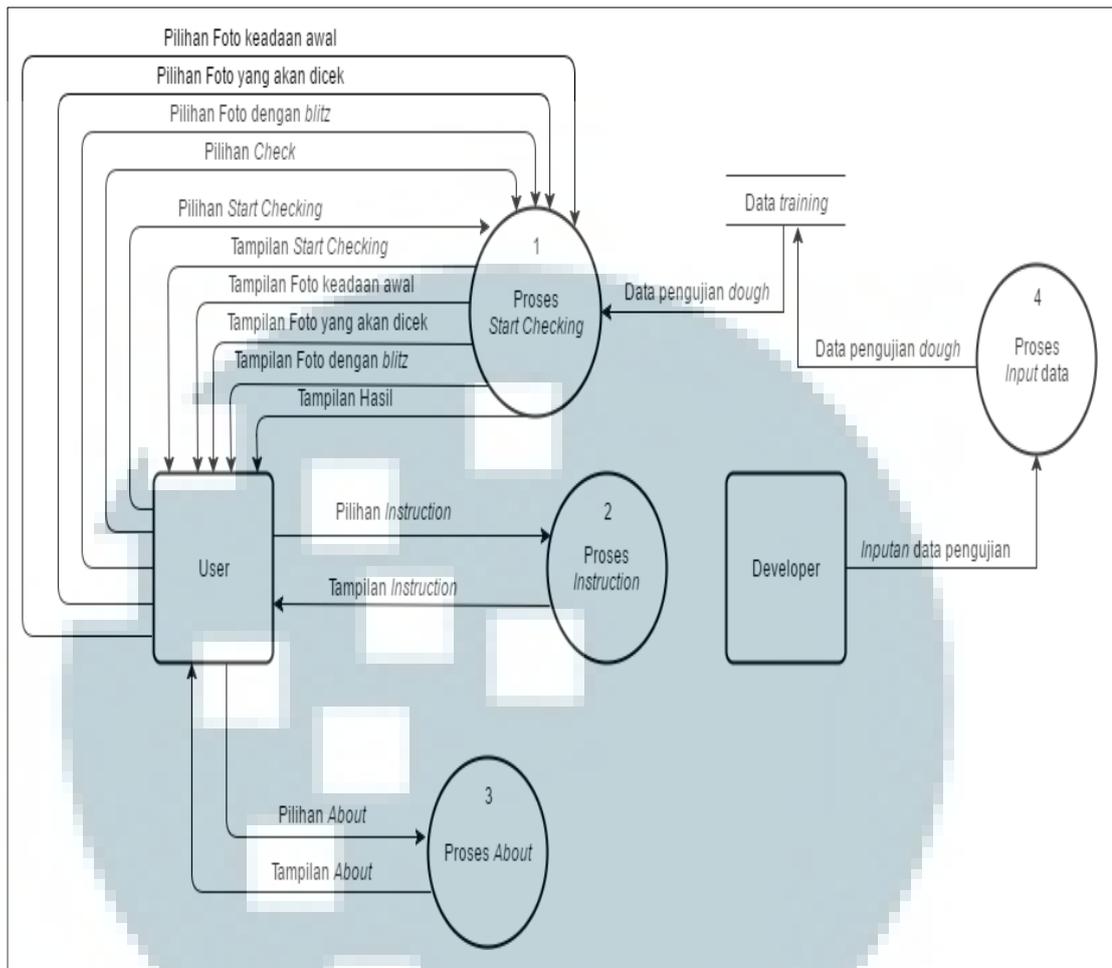
UMMN



Gambar 3.8 Diagram Konteks

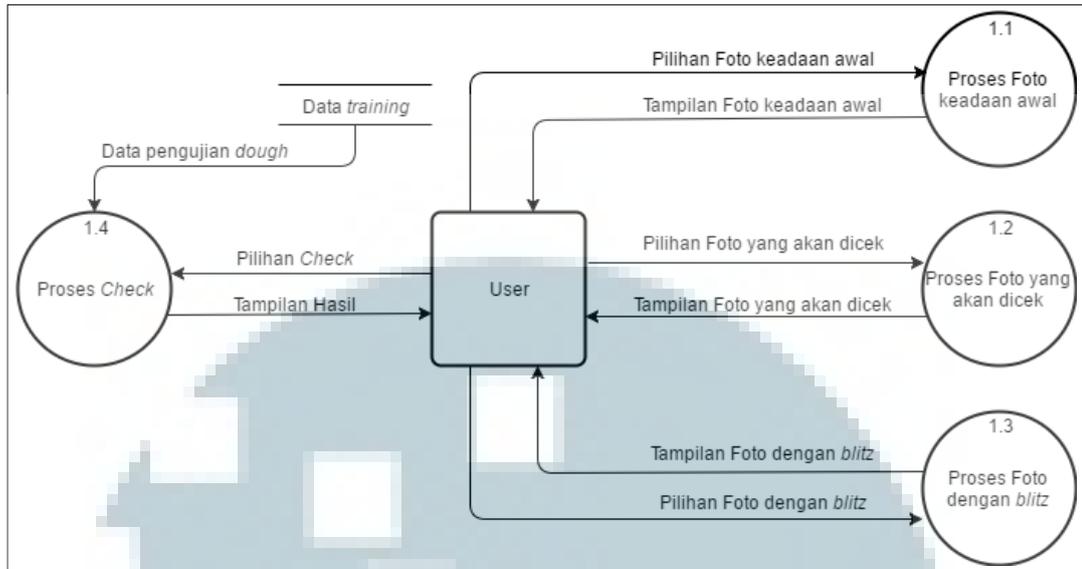
Seperti pada Gambar 3.8 aplikasi yang dibangun memiliki aliran data *input* dan data *output* yang terbagi menjadi beberapa proses untuk mengolah *input* sehingga menghasilkan *output* untuk membantu pengguna dalam menggunakan aplikasi yang dibangun. Pembagian proses dalam aplikasi dapat dilihat pada Gambar 3.9.

UMMN



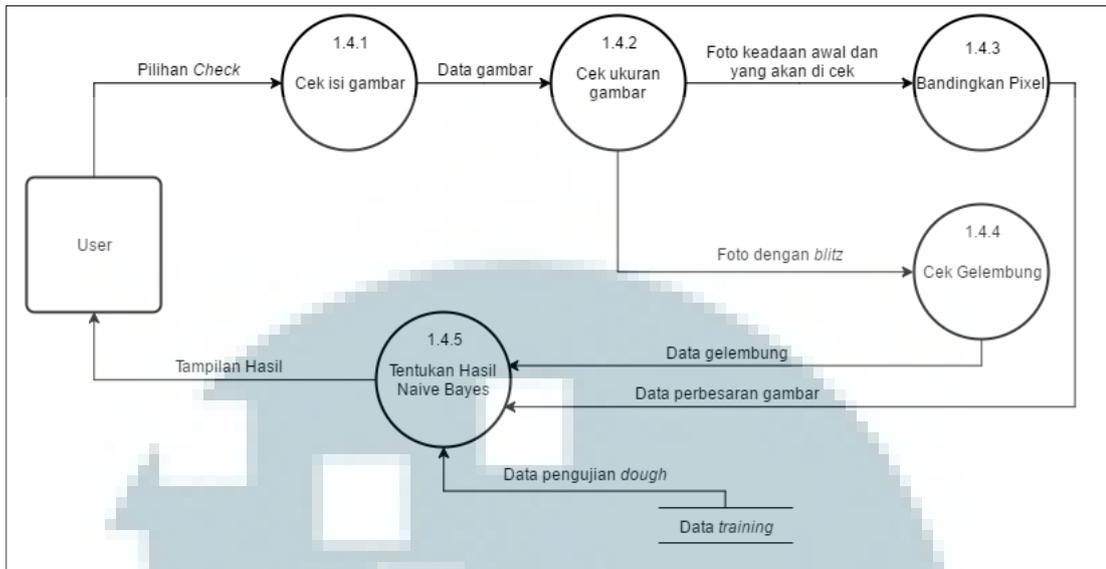
Gambar 3.9 Data Flow Diagram Level 0

Seperti pada Gambar 3.9 terdapat empat proses, yaitu proses *start checking*, proses *instruction*, proses *about*, dan proses *input data*. Untuk proses *instruction* data yang dimasukkan berupa memilih *instruction* dan proses akan menampilkan *instruction* ke pengguna. Proses *about* menerima data memilih *about* dan menampilkan *about* ke pengguna. Proses *input data* hanya dilakukan oleh *developer* yaitu memasukkan data pengujian ke dalam *database*. Pada proses *start checking* terdapat banyak *input* dan *output* antara proses dengan pengguna, proses *start checking* akan menampilkan halaman *start checking* ketika dipilih oleh pengguna. Perinciannya dapat dilihat dalam Gambar 3.10.



Gambar 3.10 Data Flow Diagram Level 1 Proses *Start Checking*

Seperti pada Gambar 3.10 data *input* dari pengguna ke proses foto keadaan awal adalah memilih foto keadaan awal, dan proses foto keadaan awal akan mengembalikan *output* berupa menampilkan foto keadaan awal. Proses foto yang akan dicek menerima *input* berupa memilih foto yang akan dicek, dan mengembalikan *output* berupa menampilkan foto yang akan dicek ke pengguna. Proses foto dengan *blitz* mengolah *input* memilih foto dengan *blitz* yang akan dikembalikan sebagai *output* berupa menampilkan foto dengan *blitz*. Proses *check* dapat dijalankan ketika menerima *input* memilih *check* dan data pengujian *dough* dari *database* data *training* yang akan mengeluarkan *output* berupa menampilkan hasil ke pengguna.



Gambar 3.11 Data Flow Diagram Level 2 Proses Check

Seperti pada Gambar 3.11 Proses *Check* dibagi menjadi 5 proses, yaitu cek isi gambar yang berfungsi untuk memastikan gambar tidak kosong, cek ukuran gambar untuk memastikan ketiga gambar berukuran sama. Proses bandingkan pixel untuk mengetahui perbesaran gambar dan proses cek gelembung untuk memastikan adanya gelembung pada adonan. Hasil dari kedua proses tersebut berupa data gelembung dan perbesaran gambar akan digunakan pada proses selanjutnya yaitu tentukan hasil Naïve Bayes. Proses tentukan hasil Naïve Bayes mengambil kedua data pengecekan dan data pengujian dari data *training* dan akan menampilkan hasil ke user.

### 3.3.3 Struktur Tabel

Dalam pembangunan aplikasi dibutuhkan *database* untuk menyimpan data *training* yang didapat dari hasil pengujian. Struktur tabel *table\_datatraining* adalah sebagai berikut.

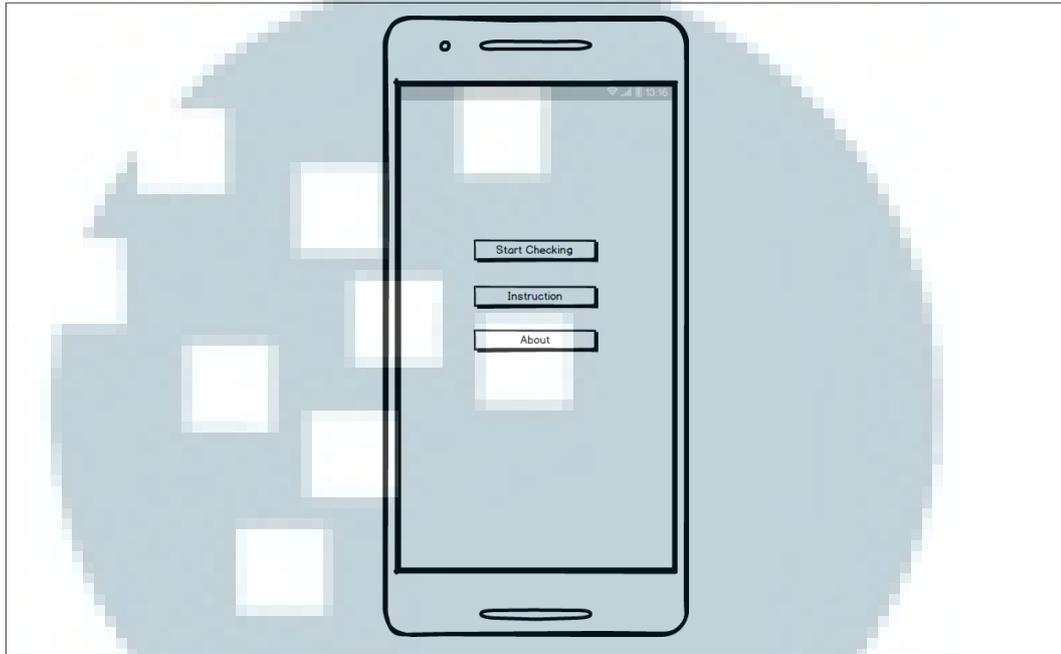
Tabel 3.1 Struktur Tabel *table\_datatraining*

Name	Type	Size	Description
id	Integer	2	Primary Key
kembang	Varchar2	5	-
gelembung	Varchar2	5	-
hasil	Varchar2	8	-

Data yang digunakan sebagai acuan untuk menghitung klasifikasi berdasarkan algoritma Naïve Bayes disimpan di dalam *table\_datatraining* seperti pada Tabel 3.1. Terdapat kolom *id* dengan tipe data integer yang menjadi *primary key* dimana tidak akan terdapat data *dough* dengan *id* yang sama sehingga data tidak tercampur. Terdapat kolom *kembang* dan *gelembung* dengan tipe data *varchar2* yang akan digunakan untuk mengisi data perkembangan dari *dough* dan gelembung yang terdapat pada *dough*. Terdapat kolom *hasil* yang akan menyimpan data yang berupa data *dough* sudah sempurna atau belum.

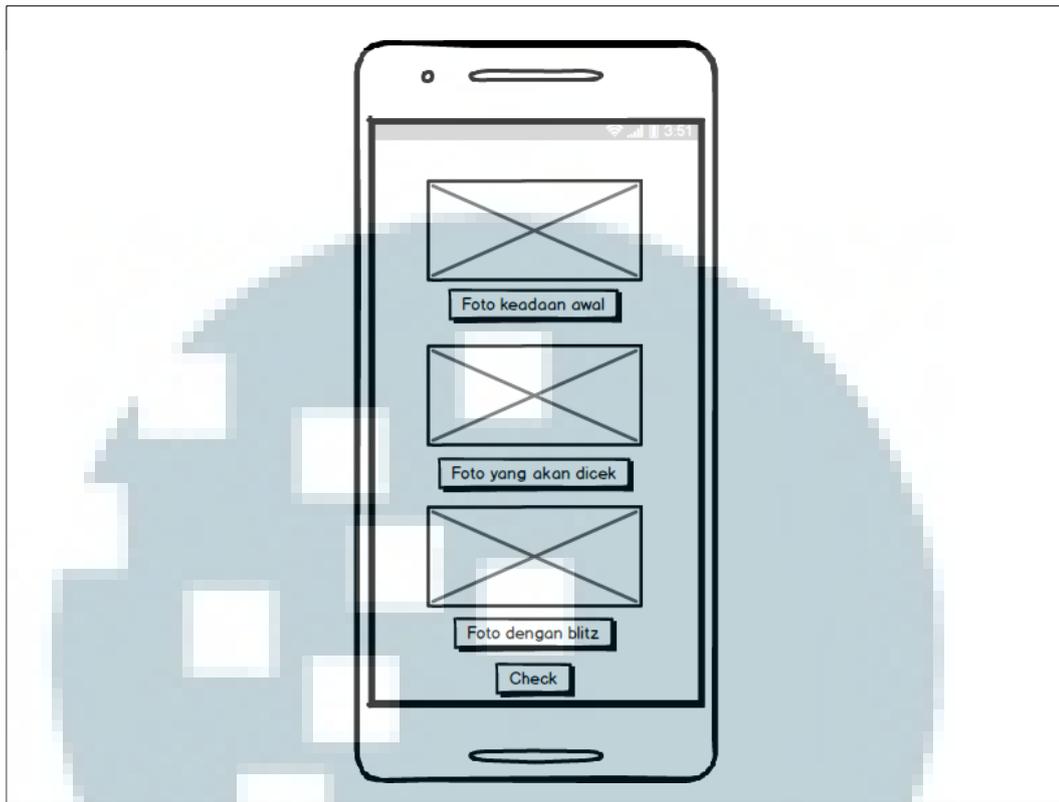
### 3.3.4 Perancangan Tampilan Antarmuka (Mockup)

Agar memudahkan proses pengembangan aplikasi, rancangan tampilan antarmuka dibuat lebih dahulu. Adapun tampilan rancangan antarmuka dapat dilihat pada Gambar 3.12



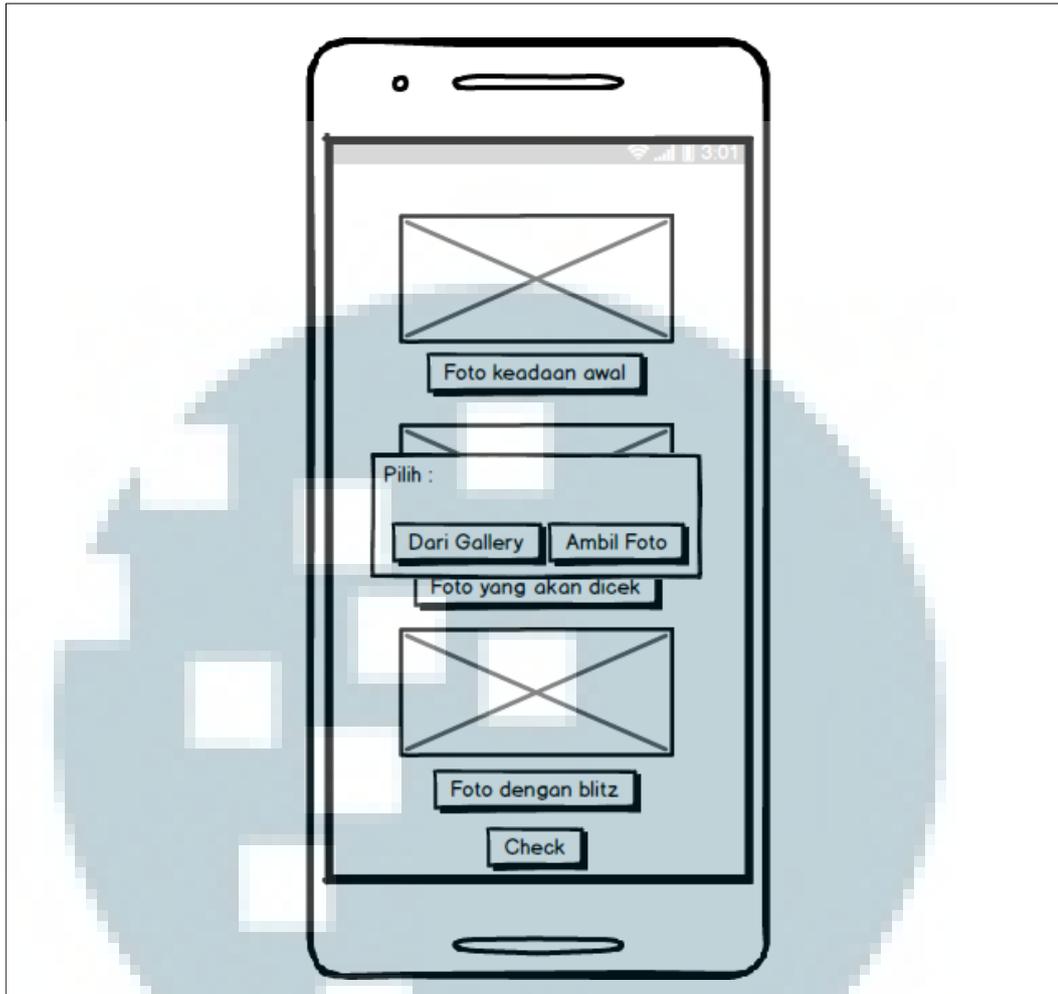
Gambar 3.12 *Mockup* Tampilan Menu Utama

Terdapat tiga buah *button* pada menu utama yaitu *Start Checking* yang berguna untuk melakukan proses pengecekan, *button* instruksi untuk menampilkan instruksi penggunaan aplikasi, dan *button about* menampilkan identitas pengembang aplikasi.



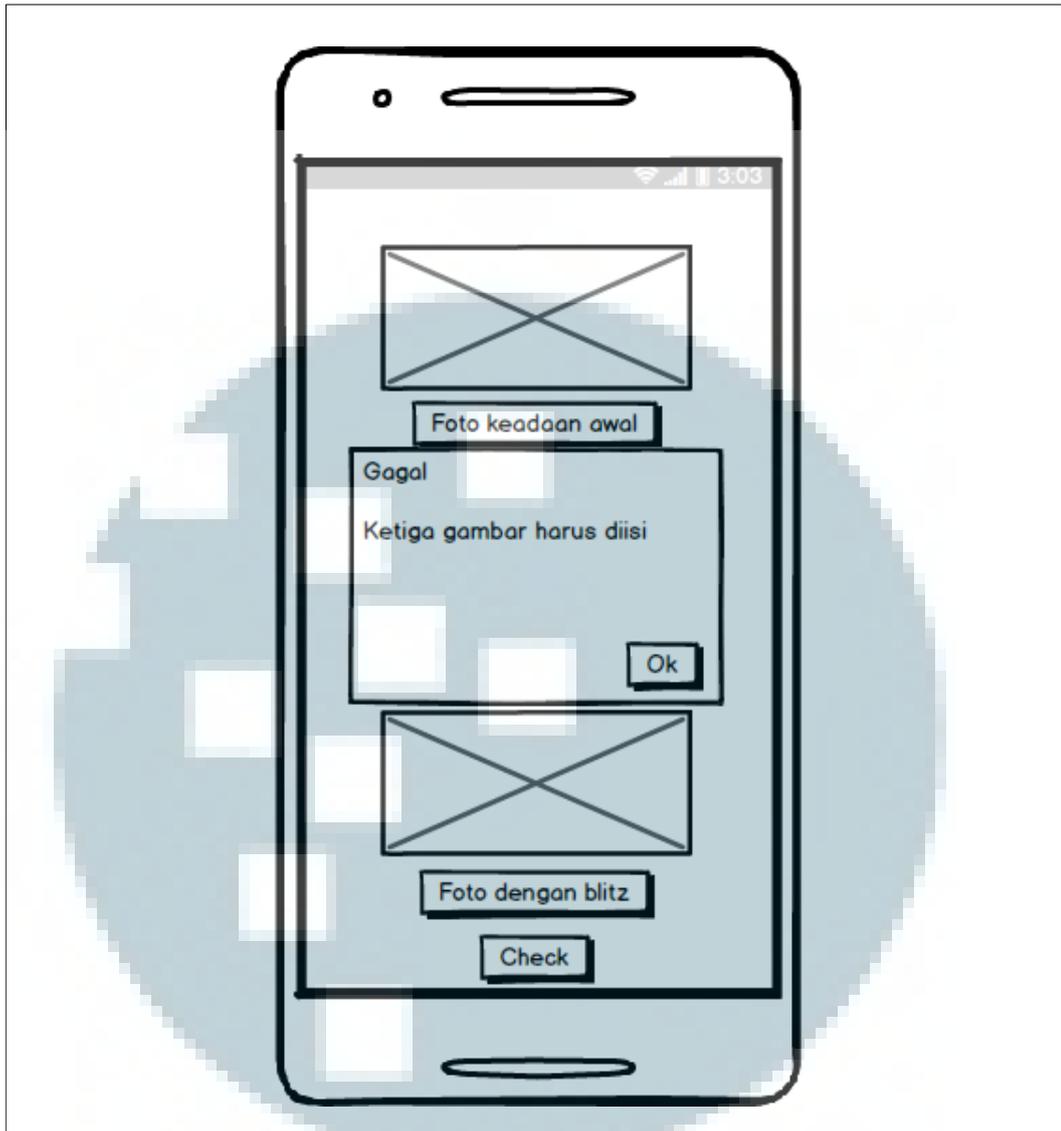
Gambar 3.13 *Mockup* Tampilan Start Checking

Gambar 3.13 menunjukkan *mockup* tampilan aplikasi ketika memilih *Start Checking*. Seperti yang dapat terlihat pada gambar di atas, pada tampilan antarmuka terdapat tiga *Imageview* dan empat *button*. Pada kondisi awal *Imageview* kosong dan ketika *button* foto keadaan awal, foto yang akan dicek, dan foto *blitz* ditekan akan muncul *pop up message* untuk memilih ingin mengambil gambar melalui kamera atau melalui *gallery* dalam *smartphone* seperti pada Gambar 3.14.



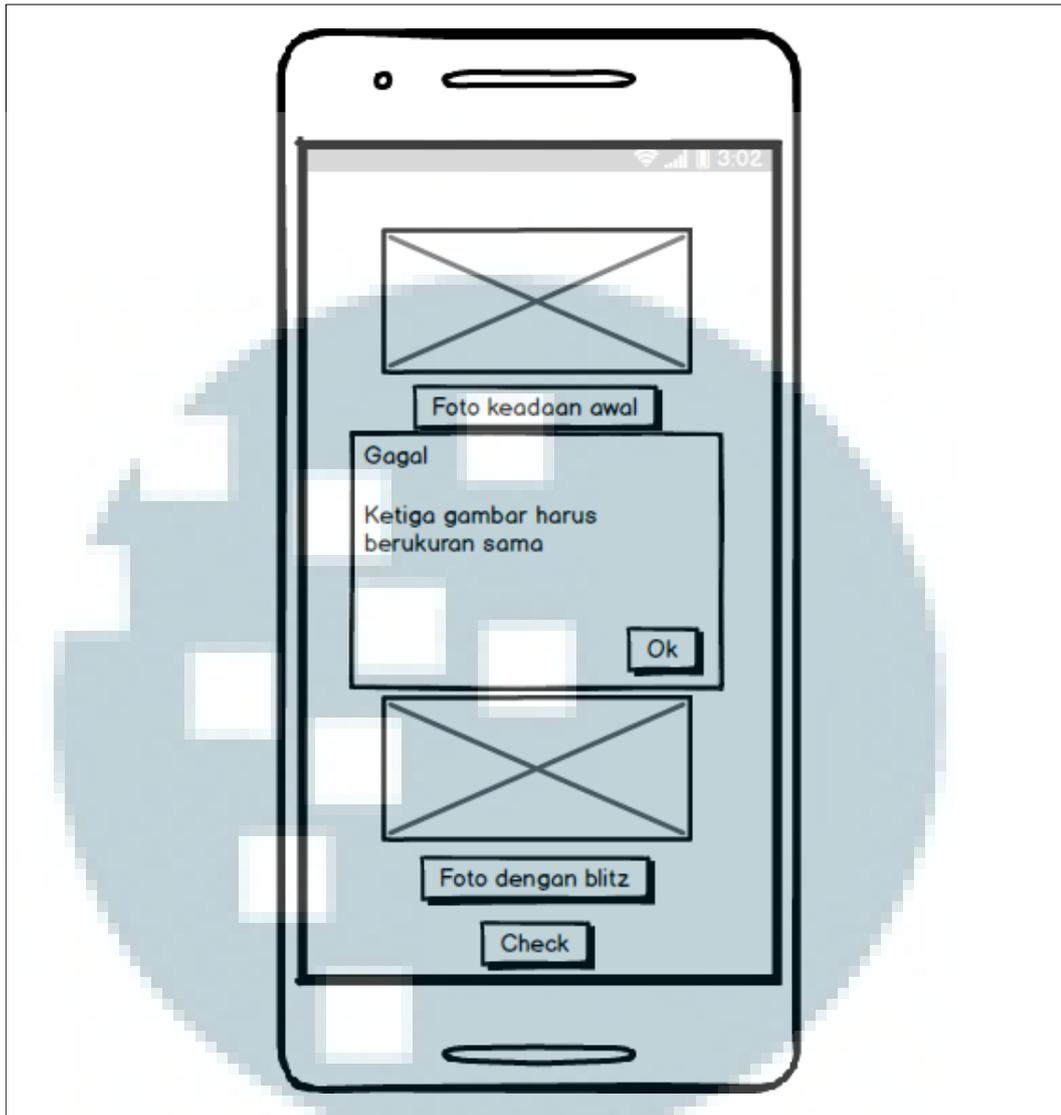
Gambar 3.14 *Mockup* Tampilan Pemilihan Pengambilan Gambar

Pada Gambar 3.14 terlihat terdapat dua *button* yaitu dari *gallery* dan ambil foto. Jika memilih dari *gallery* akan masuk ke dalam aplikasi *gallery* dan dapat memilih gambar secara langsung. Setelah gambar dipilih akan langsung ditampilkan di dalam *Imageview* di bagian atas *button* yang dipilih. Ketika memilih ambil foto maka akan masuk ke dalam aplikasi *camera* dan dapat langsung mengambil gambar, setelah gambar diambil akan langsung ditampilkan dalam *Imageview* di atas *button* yang dipilih.



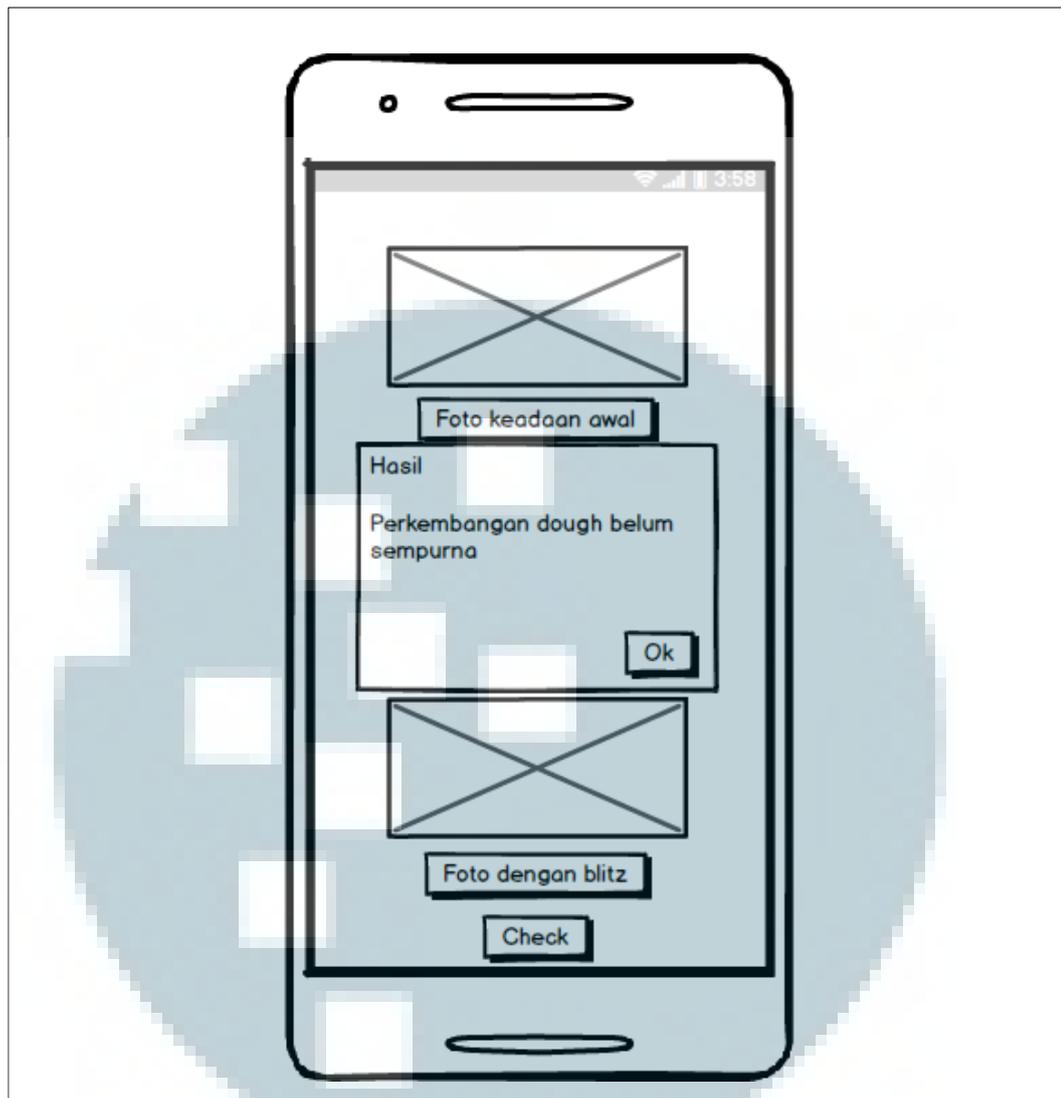
Gambar 3.15 *Mockup* Tampilan Gambar Tidak Diisi

Gambar 3.15 menunjukkan *mockup* tampilan ketika terdapat gambar yang belum diisi atau dipilih. Seperti gambar di atas akan muncul *pop up message* yang berguna untuk memberitahu bahwa terdapat gambar yang masih kosong sehingga tidak dapat melanjutkan ke proses lebih lanjut.



Gambar 3.16 *Mockup* Tampilan Gambar Tidak Berukuran Sama

Gambar 3.16 menunjukkan *mockup* tampilan ketika terdapat gambar yang memiliki ukuran tidak sama. Seperti gambar di atas akan muncul *pop up message* untuk memberitahu bahwa terdapat gambar yang memiliki ukuran berbeda, sehingga tidak dapat digunakan untuk proses lebih lanjut ketika mengambil tinggi dan lebar gambar.



Gambar 3.17 *Mockup* Tampilan Hasil Belum Sempurna

Gambar 3.17 merupakan hasil proses ketika kedua syarat terpenuhi, yaitu ketiga gambar harus diisi dan harus berukuran sama seperti pada Gambar 3.15 dan Gambar 3.16. Pada Gambar 3.17 menunjukkan *pop up message* ketika *button check* ditekan dan hasil dari pemrosesan ketiga gambar *dough* menunjukkan bahwa *dough* belum berkembang dengan sempurna.



Gambar 3.18 *Mockup* Tampilan Hasil Sempurna

Pada Gambar 3.18 sama seperti Gambar 3.17 yang merupakan hasil dari pemrosesan ketiga gambar yang memenuhi kedua syarat pengecekan, yaitu gambar harus diisi dan ketiga gambar berukuran sama. Hasil pada *pop up message* dalam Gambar 3.18 menunjukkan bahwa *dough* telah mencapai perkembangan yang sempurna.