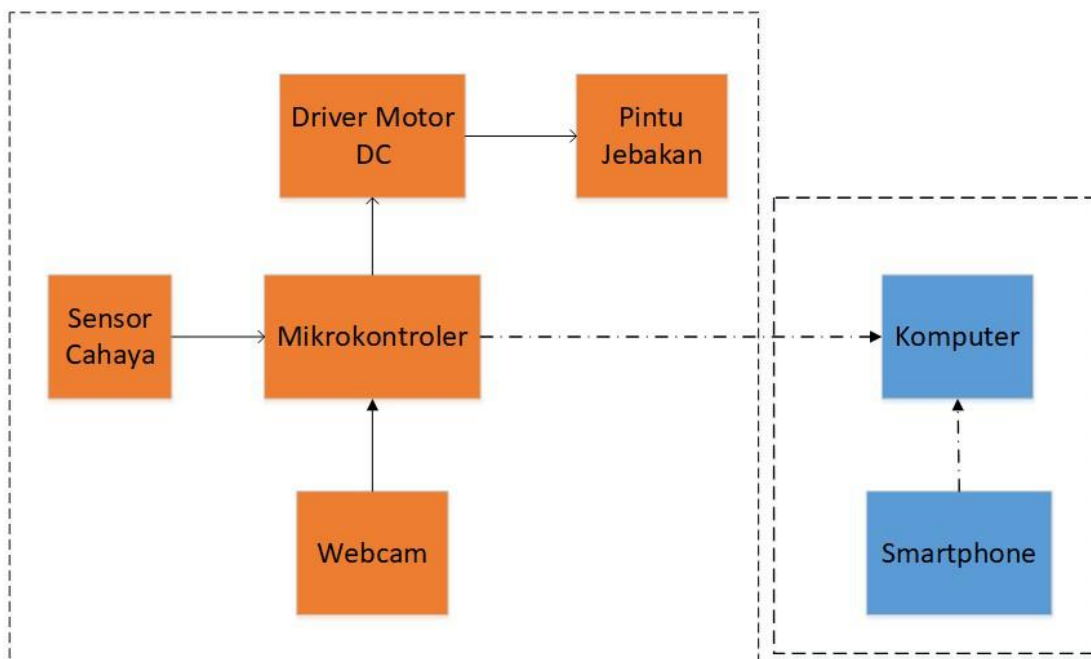


III.1 Persiapan

Dalam pembuatan perangkat cerdas tikus ini akan dijelaskan menjadi beberapa poin terlampir yakni ilustrasi sistem keseluruhan, blok diagram keseluruhan, dan blok diagram yang dikerjakan.

III.1.1 Blok Diagram Keseluruhan

Secara umum sistem yang akan dikerjakan mempunyai dua fungsi yakni bagian kontrol dan monitoring. Terdapat dua sistem pengerjaan meliputi bagian *hardware* dan *software*. *Software* sendiri muncul karena sistem akan beroperasi pada pengolah data. Sedangkan untuk *hardware* berkaitan dengan perancangan *box* alat, mekanik alat, dan skematik.



Gambar III.1 Blok Diagram Sistem Keseluruhan

Pada gambar setiap blok memiliki fungsi dan peran masing-masing yakni :

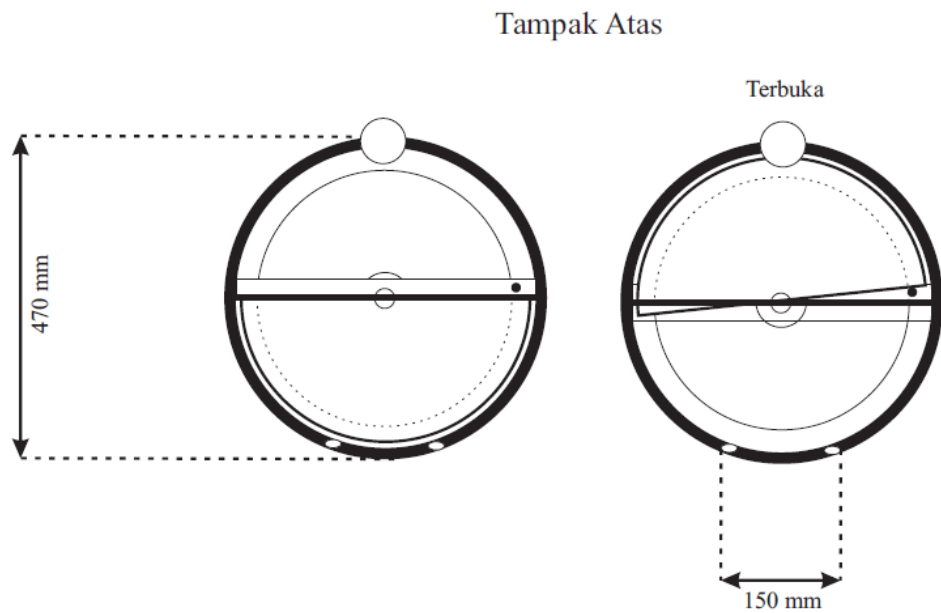
- a. Sensor Cahaya mendeteksi adanya tikus yang bergerak masuk ke perangkat dengan menghasilkan output analog yang akan di olah oleh mikrokontroller

- b. Driver motor berfungsi sebagai pengatur gerakan mekanik alat,
- c. Pintu jebakan merupakan sistem mekanik untuk menangkap tikus.
- d. Webcam atau kamera akan menangkap gambar tikus yang sudah terperangkap.
- e. Mikrokontroller adalah untuk menjalankan program perangkap tikus, program pengolahan citra, serta program pengiriman gambar melalui wifi.
- f. Komputer berfungsi untuk pengolahan citra, dan pengiriman data terhadap smartphone.
- g. Smartphone berfungsi untuk sistem monitoring.

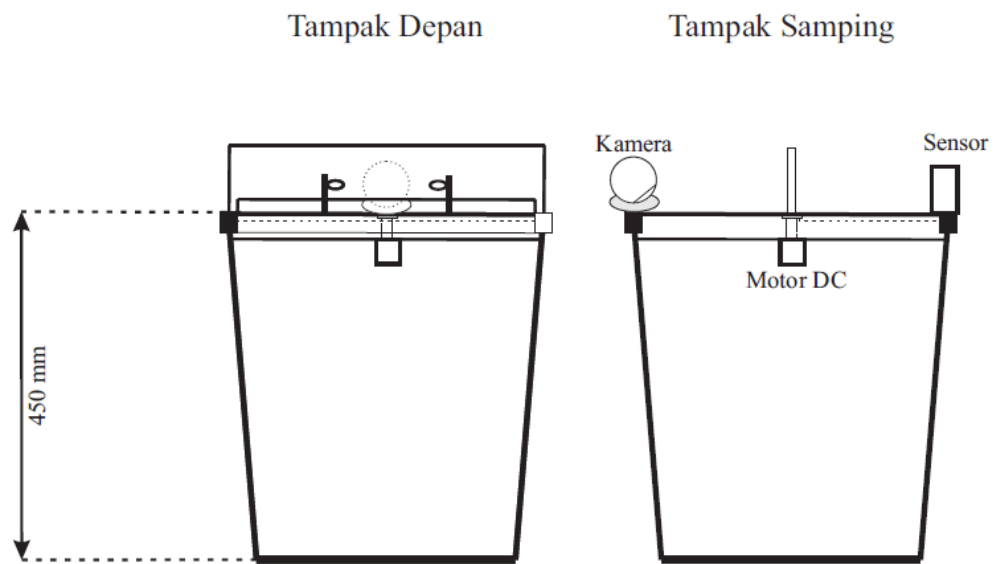
III.1.1 Perancangan Perangkap Tikus

Gambar III.4 dan gambar III.5 merupakan perancangan pada Perangkap Tikus tampak atas dan tampak samping dan depan, perangkap tikus ini akan dibuat dengan ukuran tinggi 45cm dan jari-jari 23,5cm. Adapun bahan utama terbuat dari ember plastic dengan maksud agar perangkap berbentuk lingkaran sempurna dan berwarna jelas agar memudahkan dalam proses pengolahan citra. Sedangkan penyangga-penyangga dibuat dari besi. Dan pintu jebakan dibuat dari bahan akrilik. Kemudian terdapat juga *casing* kecil yang berisi komponen-komponen utama seperti driver motor dc dan mikrokontroller dengan ukuran 15cm x 10cm x 8cm.

Pada sistem pintu jebakan menggunakan sistem pintu buka tutup, ketika ada tikus terdeteksi pintu akan terbuka dan membuat tikus terperangkap. Kemudian dengan cepat pintu akan kembali ke posisi tertutup sehingga tikus tidak dapat kabur.



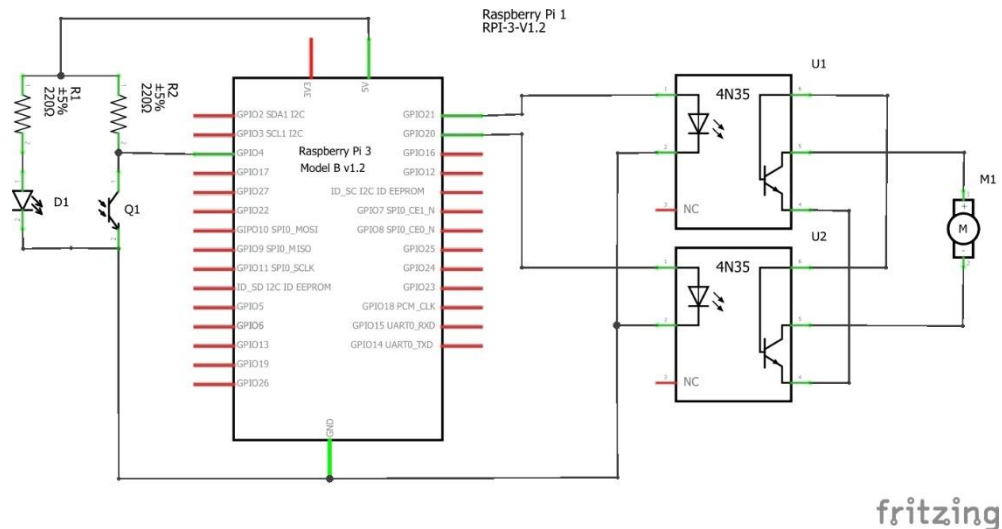
Gambar III.4 Perancangan Perangkat Tikus Tampak Atas



Gambar III.5 Perancangan Perangkat Tikus Tampak Depan dan Samping.

III.1.2 Skematik Elektronik yang Digunakan

Pada gambar III.5 merupakan rangkaian skematik. Yaitu terdapatnya sensor cahaya. Adapun hal yang merupakan bagian kontrol meliputi komponen *driver motor* untuk menggerakkan motor DC.

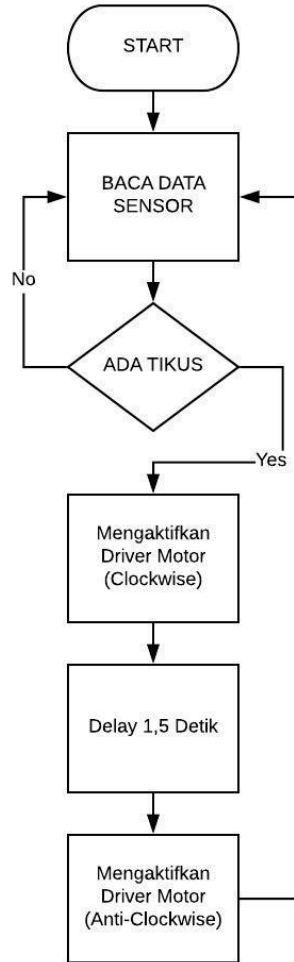


Gambar III.2 Skematik Rangkaian

III.1.3 Diagram Alir Sistem Perangkat

Tujuan utama dari sistem ini yaitu alat perangkap tikus menangkap dengan cepat tikus yang terdeteksi oleh sensor cahaya. Sistem kerja sensor cahaya dapat bekerja dengan baik yaitu phototransistor dan led super bright. Dengan menarik kabel dari phototransistor akan di dapat output sesuai dengan sensor cahaya. Bila ada tikus yang menghalangi sensor, maka output yang di dapat berupa output tegangan bernilai tinggi. Sedangkan ketika tidak ada tikus nilai outputnya kecil atau sama dengan nol.

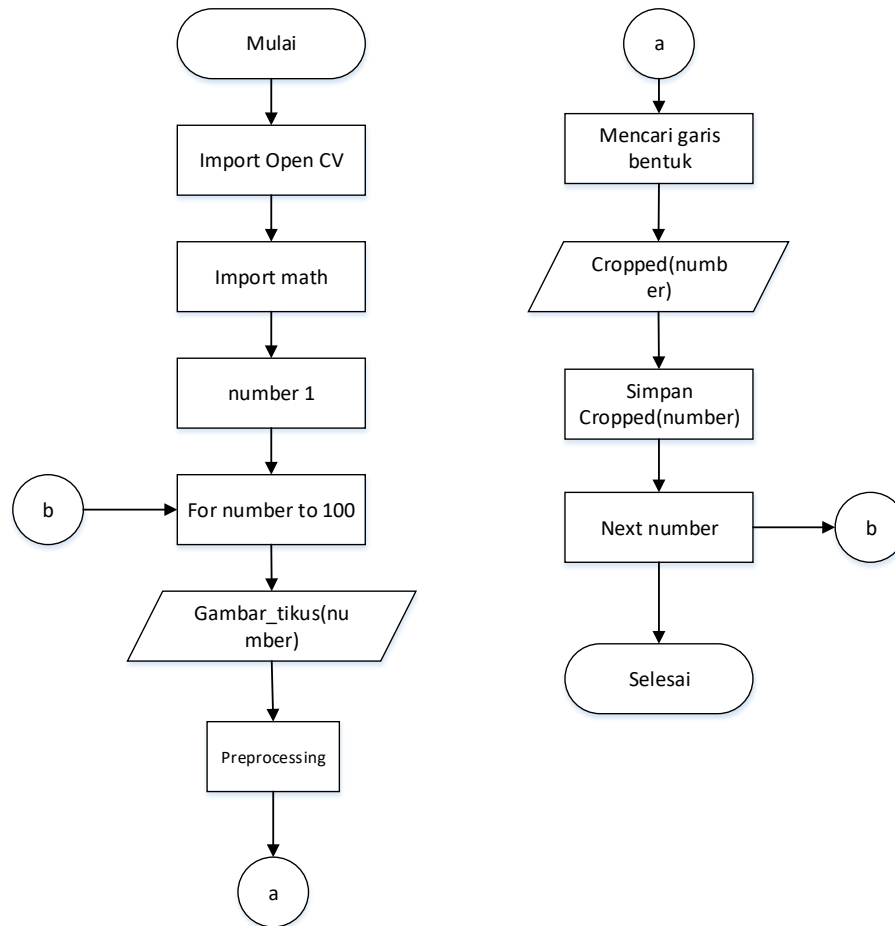
Dari gambar III.5 dapat di lihat flowchart dari program Perangkat Tikus ini secara umum. Sensor cahaya akan menjadi input untuk mikrokontroller. Ketika ada tikus terdeteksi, sensor cahaya ini akan mengirimkan signal dc atau 1. Sedangkan ketika tidak ada tikus sensor cahaya mengirimkan 0 atau low level signal..



Gambar III.3 Flowchart Program Perangkap Tikus

III.1.4 Diagram Alir Pengolahan Citra

Flowchart untuk pengerjaan sistem ini dapat dilihat pada gambar di bawah ini:



Gambar III.4 Diagram Alir Pembuatan *Template*

Sebelum melakukan proses *template matching*, *template* harus disiapkan terlebih dahulu. Gambar III.4 menunjukkan alur pembuatan *template* pada sistem ini. Pertama-tama yang harus dilakukan adalah memasukkan OpenCV library dan *math*. OpenCV library berisi library khusus untuk melakukan *image processing*. Library *math* diinputkan karena saat pembuatan program terdapat operasi matematika. Open CV library pada program ini diperlukan karena ada *preprocessing* atau perbaikan gambar. Terdapat perulangan yang banyaknya sesuai dengan *template* yang akan dibuat. Pada proses tersebut kumpulan gambar satu ekor tikus yang ditangkap akan diproses untuk dilakukan *cropping*. Ukuran gambar yang dicrop sesuai objek yang terdeteksi pada

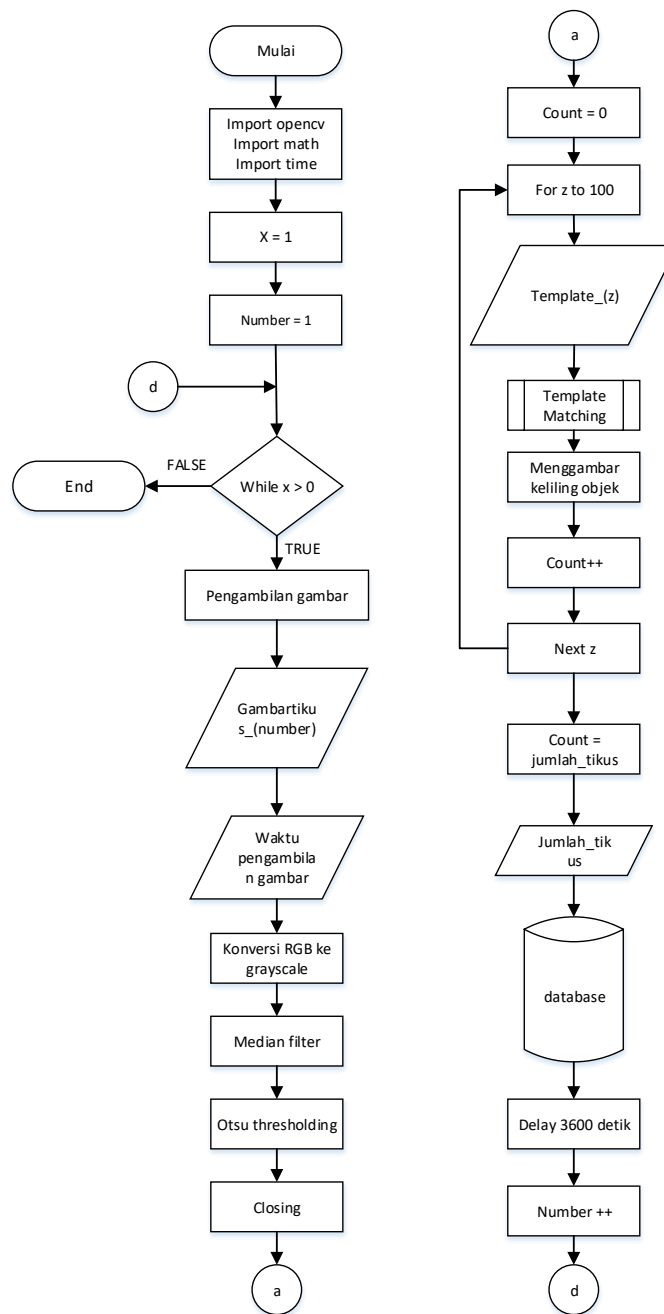
gambar itu. Setelah hasil *cropping* untuk satu gambar didapatkan maka program akan menyimpan gambar itu secara otomatis dan mengulangi proses tersebut untuk gambar yang lain.

Setelah proses pembuatan *template* selesai maka akan dilanjutkan dengan *image processing*. Berikut adalah block diagram atau alur *image processing* yang akan dilakukan:



Gambar III.5 Blok Diagram *Image Processing*

Gambar III.5 menunjukkan alur *image processing*. Input merupakan gambar yang ditangkap oleh *webcam* yang dipasang di atas perangkat. Kamera sudah diatur agar mengambil satu gambar setiap satu jam. Gambar yang tertangkap kemudian masuk ke bagian *preprocessing*. Bagian *preprocessing* merupakan proses perbaikan citra. *Preprocessing* dilakukan dengan menggunakan metode *median filter*, *Otsu thresholding* dan *closing*. Perbaikan ini bertujuan agar proses selanjutnya lebih mudah dilakukan. Perbaikan ini akan menghapus *noise* yang ada pada gambar tersebut. Setelah gambar diperbaiki maka akan dilakukan proses *feature extraction* dan *classification*. Untuk proses *feature extraction* dan *classification* digunakan metode *template matching*. Proses tersebut akan menghitung jumlah objek yang terdeteksi berdasarkan *template* yang tersedia. *Template* yang ada akan ditempatkan pada pusat bagian gambar yang akan dibandingkan lalu dihitung seberapa banyak titik yang sesuai dengan *template* tersebut. Nilai yang kesesuaiannya paling besar akan menandakan bahwa *template* tersebut paling sesuai dengan gambar yang diinputkan. Banyaknya *template* yang sesuai akan dihitung sehingga menjadi jumlah tikus yang tertangkap. Hasil akhir dari *image processing* ini adalah jumlah objek yang terdeteksi dihitung sebagai jumlah tikus yang tertangkap



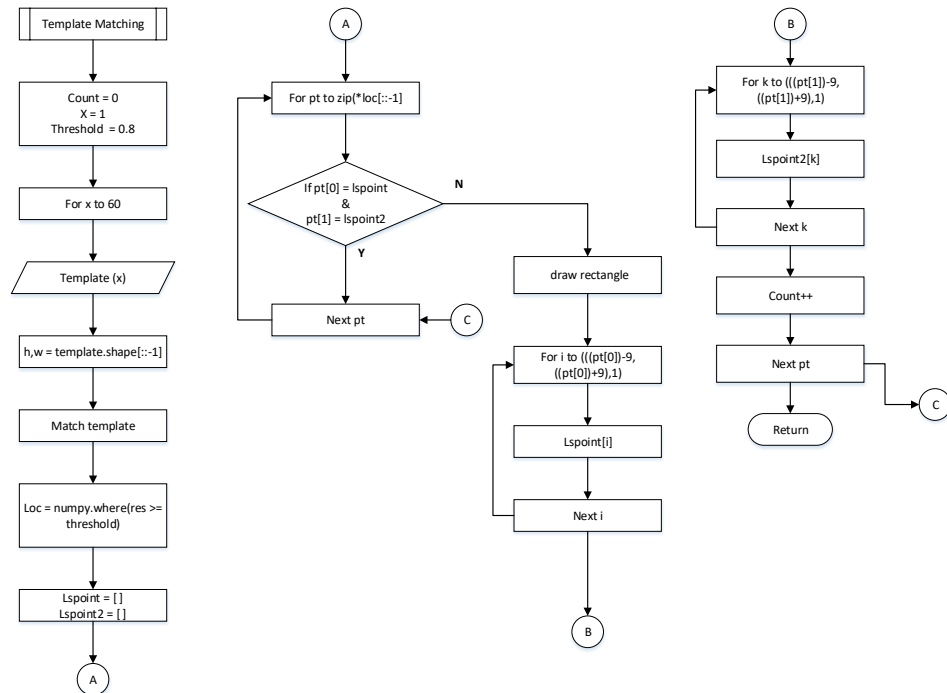
Gambar III.6 Diagram Pengolahan Citra

Gambar III.5 menunjukkan alur dari aplikasi *image processing* yang dibuat. Seperti pembuatan *template*, pada program ini beberapa *library* perlu diinputkan. Program ini membutuhkan Open CV *library* untuk proses *image processing*, *math library* untuk operasi matematika dan *time library* untuk mengatur waktu. Program

tersebut dibuat agar bisa melakukan perulangan tanpa henti sehingga program tersebut akan terus berjalan. Program tersebut akan mengambil gambar secara terus menerus tetapi hanya menyimpan gambarnya satu kali dalam satu jam. Gambar yang disimpan kemudian diproses ke tahap selanjutnya yaitu *preprocessing* atau proses perbaikan citra. Gambar yang diambil merupakan gambar/citra RGB. Kemudian gambar tersebut akan dikonversi menjadi citra *grayscale*. Gambar tersebut akan diperbaiki dengan teknik *median filter*. Teknik ini akan menghilangkan gangguan yang ada pada gambar tersebut. Setelah itu gambar akan disegmentasi menggunakan *Otsu thresholding*. Teknik ini akan memisahkan antara *background* dengan *foreground* dengan begitu proses untuk pengenalan objek akan lebih mudah. Lalu gambar akan diproses menggunakan teknik *closing*. *Closing* merupakan salah teknik morfologi citra yang bertujuan untuk menggabungkan objek-objek yang berdekatan. Teknik ini berguna juga menghilangkan lubang-lubang kecil pada citra yang diproses.

Setelah gambar diperbaiki proses selanjutnya adalah *template matching*. Gambar yang hasil perbaikan akan dibandingkan dengan *template* yang tersedia. Proses *template matching* akan dilakukan secara berulang sesuai dengan jumlah *template* yang akan dibandingkan. Ketika ditemukan nilai kesesuaian titik yang paling besar antara gambar input dan *template* maka objek akan terdeteksi. Objek yang terdeteksi itu akan dianggap sebagai tikus. Objek yang terdeteksi akan diberi garis disekelilingnya sebagai tanda. Lalu program akan menghitung gambar yang sudah diberi garis tersebut sebagai tikus yang masuk ke perangkap. Program lalu akan mengulangi proses *template matching* untuk setiap *template* yang tersedia. Ketika semua *template* sudah diproses maka akan didapatkan hasil akhir berupa jumlah tikus yang terdeteksi. Setelah proses ini selesai maka akan ada data yang disimpan di database server.

III.1.5 Diagram Alir Proses *Template Matching*



Gambar III.7 Diagram Alir *Template Matching*

Data yang disimpan di database berupa waktu pengambilan gambar, jumlah tikus, dan gambar masukan akan disimpan di database server yang telah dibuat. Data yang ada di database akan bertambah setiap jamnya selama aplikasi tersebut dijalankan. Setelah data tersebut terkirim program akan melakukan proses awal kembali yaitu mengambil gambar melalui *webcam* setelah delay selama satu jam.