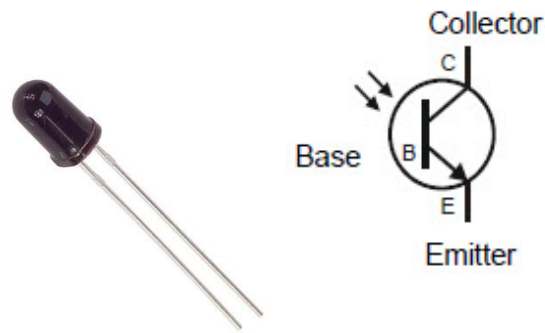


II.3 Teori Pendukung

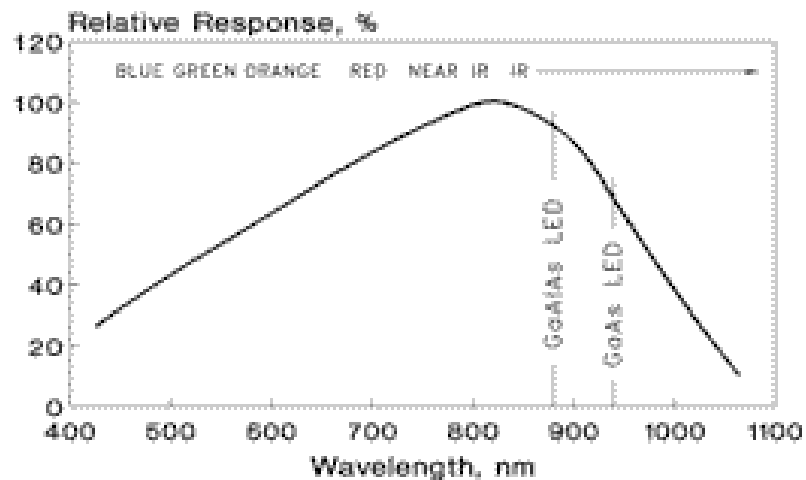
II.3.1 Phototransistor

Phototransistor adalah sebuah transistor yang kaki basisnya (B) terbuka dan terbuat dari komponen photoconductive.



Gambar II.1 Gambar Phototransistor dan Rangkaian Phototransistor

Output sebuah phototransistor tergantung pada panjang gelombang dari cahaya yang masuk. Phototransistor bereaksi terhadap cahaya dengan range spektrum panjang gelombang yang lebar mulai dari spektrum mendekati ultraviolet melewati spektrum cahaya tampak hingga mendekati spektrum inframerah. Berdasarkan Gambar II.2 respon puncak berada disekitar spektrum inframerah.



Gambar II.2 ResponsePhotransistor

Dalam tugas akhir ini penulis menggunakan Phototransistor sebagai sensor cahaya pendeteksi hama tikus.

II.3.2 Bahasa Pemrograman

Bahasa pemrograman merupakan sebuah instruksi standar yang bertugas untuk memerintah komputer. Sering disebut juga dengan bahasa komputer atau bahasa pemrograman komputer. Bahasa pemrograman juga bisa dikatakan sebagai alat untuk menampung suatu himpunan dari aturan sintaks dan semantik yang khususnya dipakai untuk mendefinisikan sebuah program yang ada di komputer. Bahasa pemrograman yang digunakan pada Raspberry penulis adalah bahasa Python.

II.3.3 OpenCV

Open Source Computer Vision Library (OpenCV) adalah sebuah library perangkat lunak yang ditujukan untuk pengolahan citra digital dinamis secara real-time. Library ini sudah sangat familiar dengan pengolahan citra computer vision. Computer vision memungkinkan komputer dapat melihat seperti manusia. Dengan computer vision tersebut komputer dapat mengambil keputusan, melakukan aksi dan mengenali suatu objek. OpenCV memiliki antar muka yang mendukung bahasa pemrograman C++, Python dan Java. Dalam tugas akhir ini, penulis menggunakan Library OpenCV untuk mendeteksi tikus dan menghitungnya.

II.3.4 Template Matching

Template matching merupakan suatu proses sederhana untuk membandingkan sebuah *template* pada basis data dengan citra masukan yang mengandung *template* tertentu. Klasifikasi umum yang digunakan dengan *template matching* adalah berdasarkan area dan fitur. Citra masukan akan dibandingkan dengan setiap *template*. Jika $I(x,y)$ adalah citra masukan dan $T_n(x,y)$ adalah *template* n maka fungsi $s(I, T_n)$ akan mengembalikan nilai yang menunjukkan seberapa cocok *template* n dengan citra masukan. Berikut adalah fungsi pencocokan:

$$s(I, Tn) = \sum_{i=0}^w \sum_{j=0}^h (I(i, j) - Tn(i, j))^2 \dots \dots \dots (2.11)$$

Keterangan :

$s(I, Tn)$ = selisih antara matrik I dan matrik Tn

$I(i, j)$ = matrik yang mewakili citra masukan

$Tn(i, j)$ = matrik x dan y yang mewakili citra *template*

w = lebar citra

h = tinggi citra

Template ditempatkan pada pusat bagian citra yang akan dibandingkan dan dihitung seberapa banyak titik yang paling sesuai dengan *template*. Citra *template* yang paling sesuai dengan citra masukan dapat dilihat berdasarkan nilai terbesar kesesuaian titik antara citra masukan dan citra *template*. [6]