

BAB II KAJIAN PUSTAKA

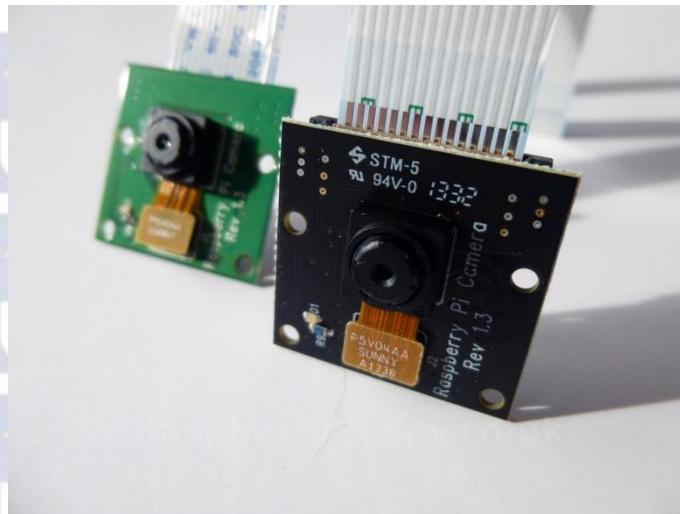
2.1 Perangkat Keras Alat Bantu

Alat bantu dapat berupa tongkat yang paling umum digunakan, selain itu juga terdapat inovasi berupa sepatu, topi, kacamata, sarung tangan, sabuk, gelang, dan sebagainya. Penggunaan tongkat dalam menunjang aktivitas sehari-hari tunanetra masih memiliki banyak kekurangan. Selain membuat tunanetra kesulitan dalam membawanya, jarak pendeteksian objeknya juga masih tergolong dekat.

Aplikasi Raspberry Pi dalam aplikasi elektronik penuntun bagi tunanetra sudah pernah dilakukan oleh Tripathi (2014). Aplikasi tersebut dibuat untuk menunjang aktivitas navigasi tunanetra. Keluaran yang digunakan berupa suara *beep* melalui *headphone*. Kemudian sensornya berupa sensor ultrasonik untuk mendeteksi benda dan kamera *web cam* untuk mendeteksi manusia. Namun, untuk menggunakan alat navigasi tersebut, penyandang tunanetra harus memakai sabuk yang berisi board Raspberry Pi, sensor dan kamera yang beratnya sekitar 500 gram.



Gambar 2.1 Perancangan Alat Navigasi untuk Tunanetra (Tripathi, 2014)



Gambar 2.2 Kamera Raspberry Pi IR dan NoIR (<http://www.raspberrypi-spy.co.uk>)

Pada penelitian ini, penggunaan Raspberry Pi dalam pengenalan tangga dan manusia menggunakan sensor kamera Raspberry Pi 5 MP yang berukuran 25 mm x 20 mm x 9 mm dengan bobot sekitar 3 gram. Penggunaan modul kamera Raspberry Pi yang jauh lebih ringan dari *web cam* dapat mengurangi beban pengguna alat bantu ini. Kamera ini mampu menyajikan gambar 2592 x 1944 piksel dan video 1080 piksel pada 30 *frame per second*, 720 piksel pada 60 *frame per second* dan 480 piksel pada 90 *frame per second*.

Kamera Raspberry Pi terdiri dari 2 jenis, yaitu Kamera Pi IR (*infrared*) dan NoIR (*no infrared*). Kamera Pi IR merupakan kamera yang disertai dengan filter inframerah dan NoIR merupakan kamera tanpa inframerah. Kamera Pi NoIR biasanya digunakan pada kondisi pencahayaan rendah (gelap). Untuk pendeteksian manusia, kedua kamera ini dapat digunakan pada siang hari, namun kamera Pi NoIR memiliki kemampuan lebih dalam menangkap gambar pada malam hari. Jadi, pada penelitian ini digunakan kamera Pi NoIR.

Bentuk fisik alat bantu pada penelitian ini hanya terdiri dari kamera kecil yang diberi gantungan sehingga bisa dengan mudah tergantung pada tubuh tunanetra, seperti kalung. Kamera tersebut terhubung dengan kotak Raspberry Pi dan *power bank* 5800 mAh sebagai sumber tegangan 5V/1A. Kotak dan Raspberry Pi diletakkan di dalam tas yang dibawa oleh penyandang tunanetra. Untuk keluaran suara, pada kotak Raspberry Pi dihubungkan sebuah *headset* yang bisa digunakan oleh penyandang tunanetra untuk mendengarkan perintah arah yang harus dituju.

2.2 *Embedded System*

Embedded system adalah sistem yang tertanam pada suatu perangkat dan memiliki peran dalam menjalankan suatu fungsi tertentu. Secara garis besar, *embedded system* mirip dengan *Operating System (OS)* yang terdapat pada komputer. Namun, *embedded system* tidak bisa melakukan hal umum seperti mengedit data, memutar multimedia hingga simulasi. *Embedded system* hanya mampu melakukan satu tujuan tertentu sesuai dengan program yang diberikan kepadanya. Aplikasi *embedded system* antara lain, rancangan untuk mengoptimasi mesin, pengurangan biaya produk dan meningkatkan performa sistem. Jenis *embedded system* terkenal yang memiliki platform dan pemrogramannya sendiri, antara lain:

a. Arduino

Arduino merupakan *platform* yang bersifat *open source* dengan kombinasi dari perangkat keras, bahasa pemrograman dan *Integrated Development Environment (IDE)*. IDE adalah sebuah perangkat lunak yang berperan untuk menulis program, menyusun menjadi kode biner dan mengunggah ke dalam memori.



Gambar 2.3 Arduino Uno (<http://arduino.cc/en/>)

Keunggulan Arduino sebagai *embedded system* ini adalah harganya murah dibandingkan dengan *embedded system* lainnya, bersifat *open source*, mudah dipelajari bahasa pemrogramannya dan sebagainya.

b. Raspberry Pi

Raspberry Pi merupakan *embedded system* berupa mini komputer seukuran kartu kredit dengan sistem operasi Linux yang dikembangkan oleh Yayasan Raspberry Pi Foundation di Inggris.



Gambar 2.4 Raspberry Pi Model A dan B+ (<http://cdn.tinhte.vn/>)

Raspberry Pi terdiri dari dua model, yaitu model A dan model B+. Raspberry Pi model A ini diluncurkan pada tahun 2009 yang kemudian disusul dengan peluncuran model B pada tahun 2012. Perbedaan model A dan B terletak pada memori yang digunakan. Model A dilengkapi dengan memori 256MB dan model B 512MB. Selain itu, model B dilengkapi dengan kartu jaringan yang tidak terdapat pada model A.

Desain Raspberry Pi adalah *SoC (System-on-a-chip)* yang menanamkan prosesor ARM1176JZF-S dengan 700 MHz, GPU berupa VideoCore IV dan RAM 512MB. Penyimpanan data tidak menggunakan *harddisk* atau *solid-state drive*, melainkan menggunakan kartu SD untuk *booting* dan penyimpanan jangka panjang.



Gambar 2.5 Raspberry Pi 2 Model B

(<https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale/>)

Pada Februari 2015, telah dikeluarkan model Raspberry Pi baru, yaitu Raspberry Pi 2 Model B. Model terbaru ini mempunyai ukuran sama persis dengan Raspberry Pi B+, namun dengan *upgrade* sebagai berikut:

1. Kecepatan 6 kali lebih cepat dibandingkan Raspberry Pi Model B+,
2. Prosesor berupa *quad core* dengan arsitektur ARMv7. Dengan menggunakan prosesor *quad core* ARMv7, Raspberry Pi dapat digunakan untuk menjalankan linux, Snappy Ubuntu Core dan Microsoft Windows 10,
3. Memori RAM 1GB.

Raspberry Pi 2 Model B dilengkapi dengan 8GB NOOBS (*New Out Of the Box Software*). NOOBS adalah perangkat lunak untuk mengatur Raspberry Pi ketika proses *booting* sehingga lebih mudah. *Operating System* yang digunakan merupakan Raspbian. Raspbian adalah sistem operasi Debian Linux dan dioptimisasi khusus untuk perangkat keras Raspberry Pi. Sistem operasi Raspbian ini merupakan program dasar dan program kegunaan yang membuat Raspberry Pi dapat bekerja. Namun, Raspbian menyediakan lebih dari sistem operasi murni. Raspbian dilengkapi dengan lebih dari 35.000 paket program, perangkat lunak yang telah di *compile* dalam format yang mudah dipasang pada Raspberry Pi.

Selain dapat berfungsi sebagai mini komputer pada umumnya, Raspberry Pi juga memiliki sistem *input-output* dan beberapa *interfaces* yang digunakan untuk terhubung ke modul elektronika yang lain. Beberapa *interfaces* yang dimiliki, yakni UART, SPI dan I2C. Selain itu, Raspberry Pi juga dapat secara langsung mengatur logika *input-output* untuk mengendalikan *relay* atau perangkat elektronika lainnya. Sumber daya yang diperlukan Raspberry Pi untuk GPIO adalah 3,3 Volt. Melihat kelebihan yang ada, maka pada penelitian ini digunakan Raspberry Pi 2 Model B untuk mendukung pengolahan citra yang dilakukan. SD Card yang digunakan adalah dengan memori 16GB agar cukup untuk menginstal *library* OpenCV.

2.3 Pengolahan Citra Digital

Menurut Jain (1989), istilah pengolahan citra digital merupakan pemrosesan gambar berdimensi dua melalui komputer digital. Menurut Efford (2000), pengolahan citra digital merupakan istilah umum untuk memanipulasi dan memodifikasi citra dengan berbagai cara. Pengolahan citra digital dapat dibagi menjadi dua jenis kegiatan, yakni:

1. Memperbaiki kualitas suatu gambar (dimensi dua) sehingga dapat lebih mudah diinterpretasi oleh manusia.
2. Mengolah informasi yang terdapat pada suatu gambar untuk pengenalan objek secara otomatis.

Pengolahan citra pada penelitian ini bertujuan untuk pengenalan manusia secara otomatis. Salah satu cabang dari pengolahan citra digital adalah *Computer Vision*. *Computer Vision* memiliki keunggulan dibandingkan cabang lainnya, karena *computer vision* bekerja dengan meniru cara kerja manusia (*human vision*) dalam mengolah citra (Kusumanto, 2012). Terdapat dua metode dalam pengolahan citra untuk pendeteksian manusia, yakni:

- a. HSV (*Hue, Saturation, Value*)

Pemodel warna HSV mendefinisikan warna dalam bentuk *Hue, Saturation* dan *Value*. *Hue* menyatakan warna asli, seperti merah, violet, dan kuning. *Hue* ini pada umumnya digunakan untuk membedakan warna-warna dan menentukan tingkat kemerahan, kehijauan dan warna lainnya dari cahaya. *Hue* dapat berasosiasi dengan panjang gelombang cahaya.

Saturation menyatakan tingkat kemurnian suatu warna, yaitu menunjukkan seberapa banyak warna putih yang diberikan pada warna. Sedangkan, *Value* menyatakan banyaknya cahaya yang diterima oleh mata tanpa memperdulikan warnanya (Fitria Purnamasari, 2009).

Pendeteksian manusia dengan menggunakan metode HSV diawali dengan mengubah citra dari warna RGB (*Red, Green* dan *Blue*) menjadi HSV. Kemudian melakukan pengkompresan pencahayaan, segmentasi warna kulit, penggunaan *low pass filter*, dan mulai program untuk mendeteksi kulit serta pemisahannya dengan lingkungan sekitar. Kelebihan HSV ini adalah warna-warna yang didapatkan sama dengan warna yang ditangkap oleh manusia.

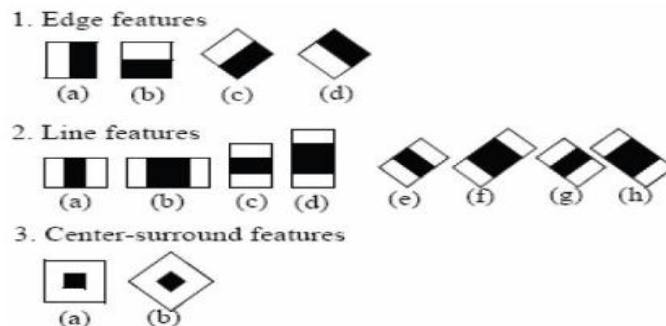
Telah dilakukan suatu penelitian mengenai sistem pendeteksi kulit manusia menggunakan segmentasi warna kulit pada tipe citra HSV oleh Iona Usuman (2012). Hasil penelitian menunjukkan pendeteksian manusia bekerja dengan sangat baik dengan kecepatan rata-rata 1.4 *frame per second* dan tingkat akurasi 87.86%.

b. *Haar-like Feature*

Metode selanjutnya adalah metode *haar-like feature*. *Haar-like feature* pertama kali dikenalkan oleh Viola Jones (2003) dan dikembangkan oleh Lienhart sebagai *real time face detection*. Menurut Kusumanto (2012), terdapat 4 konsep utama dalam mendeteksi objek pada gambar, yakni:

1. Training data dengan menggunakan dua model, seperti berikut:
 - a) *Positive Sample*, berisi gambar dari objek yang ingin dideteksi.
 - b) *Negative Sample*, berisi gambar selain objek yang ingin dideteksi.

2. Fitur segi empat yang disebut fitur *Haar*. *Haar feature* merupakan fitur yang didasarkan pada gelombang tunggal bujur sangkar yang terdiri dari satu interval tinggi dan satu interval rendah. Pada gambar dua dimensi, berarti terdiri dari satu terang dan satu gelap. Sehingga, setiap *haar-like feature* terdiri dari gabungan kotak hitam dan kotak putih.



Gambar 2.6 Macam-macam Variasi Fitur pada *Haar* (Kusumanto, 2012)

Nilai *haar-like feature* diperoleh dari selisih jumlah nilai piksel pada daerah gelap dengan jumlah nilai piksel pada daerah terang, dapat dilihat pada persamaan 2.1 berikut:

$$F(\text{Haar}) = \sum F_{\text{White}} - \sum F_{\text{Black}} \quad (2.1)$$

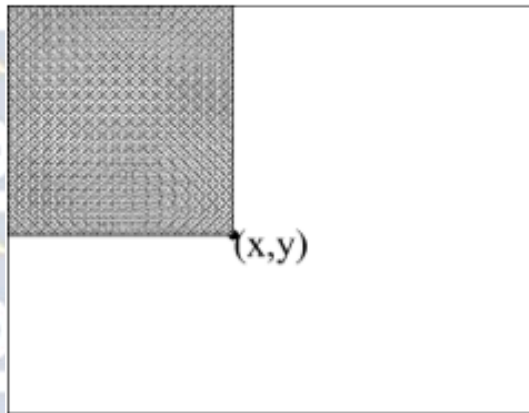
dimana:

$F(\text{Haar})$ merupakan nilai fitur total,

$\sum F_{\text{White}}$ merupakan nilai fitur pada daerah terang, dan

$\sum F_{\text{Black}}$ merupakan nilai fitur pada daerah gelap.

3. Integral gambar untuk mendeteksi fitur secara cepat, kemudian diintegrasikan (dijumlahkan) berdasarkan nilai-nilai piksel yang terdapat pada gambar.



Gambar 2.7 Integral Gambar (x,y) (Viola Jones, 2003)

Berdasarkan Gambar 2.7, citra integral pada titik (x,y) yaitu $ii(x,y)$ dapat dicari dengan menggunakan persamaan 2.2 seperti berikut:

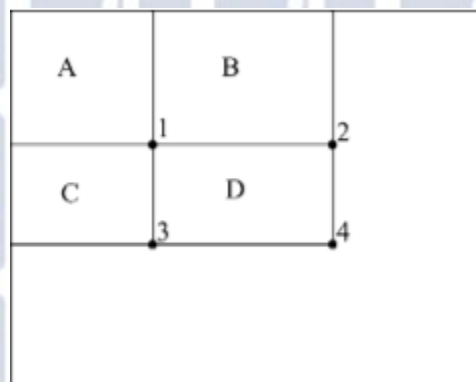
$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y') \quad (2.2)$$

dimana:

$ii(x,y)$ merupakan citra integral pada lokasi (x,y) , dan $i(x',y')$ merupakan nilai piksel pada citra asli

Perhitungan nilai dari suatu fitur dapat dilakukan secara cepat dengan menghitung integral citra pada empat buah titik seperti yang ditunjukkan pada

Gambar 2.8 di bawah ini.



Gambar 2.8 Perhitungan Nilai Fitur (Viola Jones, 2003)

Jika nilai integral citra pada titik 1 adalah A, titik 2 adalah A+B, titik 3 adalah A+C dan titik 4 adalah A+B+C+D, maka jumlah piksel di daerah D didapatkan dengan cara $4 + 1 - (2 + 3)$.

4. Pengklasifikasi bertingkat untuk menghubungkan banyak fitur dengan lebih efisien.

Penelitian metode *haar-like feature* ini pernah dilakukan oleh Miky Kurniadi (2014) tentang *tracking* wajah dengan metode *haar-like feature* menggunakan webcam statis. Metode yang digunakan berupa metode *haar-like feature* yang dibangun oleh OpenCV. Hasil penelitiannya menunjukkan tingkat keberhasilan sebesar 100% untuk mendeteksi wajah tanpa halangan pada jarak 30-150 cm.

Dengan membandingkan kedua metode yang digunakan dalam mendeteksi objek berupa manusia di atas, penelitian ini menggunakan metode *haar-like feature* karena tingkat keberhasilannya dalam mendeteksi manusia lebih tinggi dibandingkan metode HSV. Metode HSV menggunakan segmentasi warna yang memiliki hasil yang cukup bagus jika ditambahkan dengan filter. Namun, pada keadaan tanpa filter, metode *haar-like feature* menghasilkan gambar yang lebih baik dengan langkah yang lebih sederhana, yaitu menggunakan fitur pada OpenCV.

2.4 Metode *Haar-like Feature* pada Raspberry Pi

Aplikasi Raspberry Pi dalam pengolahan citra untuk *face detection* menggunakan metode *haar-like feature* sudah banyak digunakan. Salah satunya adalah pada penelitian yang dilakukan oleh Marut Tripathi (2014) dalam aplikasi elektronik penuntun bagi tunanetra. Alat ini dapat mendeteksi objek dan melakukan

perhitungan jarak antara objek dengan penggunanya. Selain itu, dengan menggunakan kamera *web cam*, alat ini dapat mendeteksi wajah dengan metode *haar cascade*. Program yang digunakan pada *board* adalah pemrograman dengan bahasa Python. Penelitian tersebut menunjukkan bahwa dengan menggunakan Raspberry Pi, alat navigasi dapat mendeteksi objek halangan hingga 300 cm dan objek manusia pada jarak 120 cm.

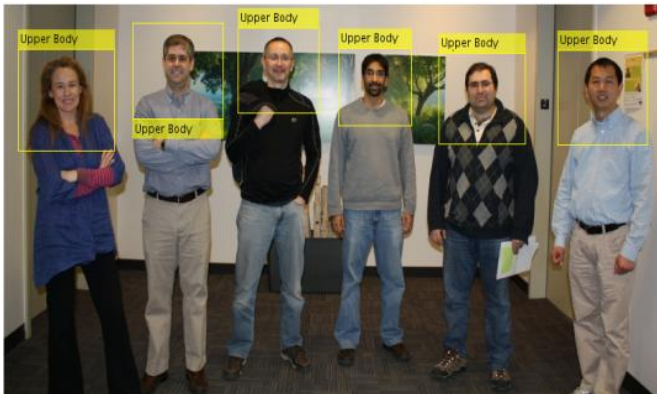
Sebelum memulai pemrograman *haar-like feature* pada Raspberry Pi, diperlukan suatu *Operating System (OS)* yang dijalankan dari SD card pada board Raspberry. OS disimpan pada SD card dan proses *booting* berasal dari SD Card itu sendiri. Raspberry Pi dengan OS Raspbian atau lainnya sudah banyak yang dapat mendukung Python sebagai bahasa pemrograman utama.

Untuk suatu gambar yang mengandung objek serumit wajah, tidak ada sebuah percobaan sederhana yang dapat memberi tahu jika telah ditemukan sebuah wajah dalam sebuah gambar. Sebaliknya, terdapat ribuan fitur kecil yang harus dicocokkan untuk menemukan sebuah wajah. Algoritma *haar* dapat memecah-mecah setiap bagian wajah menjadi lebih kecil untuk mengidentifikasi wajah tersebut.

Setiap gambar yang diperoleh dapat dipecah ke dalam beberapa bagian kemudian dilakukan pendeteksian dengan fitur *haar* seperti yang dijelaskan pada sub bab sebelumnya. Hal ini dapat dilakukan sesuai dengan jenis pendeteksian yang kita inginkan. Terdapat beberapa jenis pendeteksian manusia yang dapat dilakukan, yakni deteksi wajah, deteksi tubuh bagian atas dan deteksi seluruh badan.



Gambar 2.9 Deteksi Wajah (<http://www.mathworks.com>)



Gambar 2.10 Deteksi Tubuh Bagian Atas (<http://www.mathworks.com>)



Gambar 2.11 Deteksi Seluruh Tubuh (Tripathi, 2014)

Pemrograman pada penelitian ini hanya fokus pada pendeteksian manusia melalui wajah. Pendeteksian ini bertujuan agar manusia dapat terdeteksi lebih cepat dan perintah akan diberikan sesuai dengan posisi wajah yang terdeteksi.

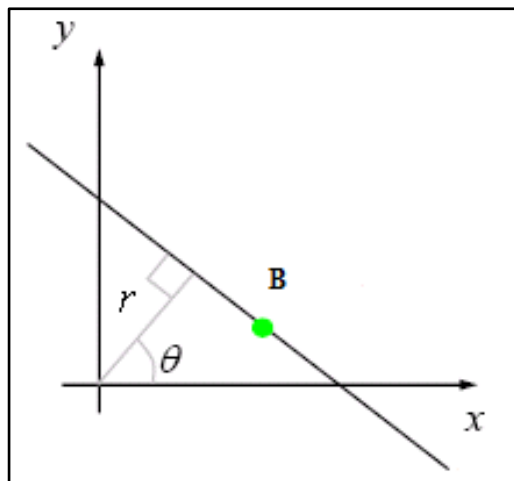
2.5 Metode *Hough Line Transform* pada Raspberry Pi

Alat bantu pengarah jalan yang dirancang tidak hanya dapat mendeteksi manusia, melainkan juga dapat mendeteksi tangga. Tangga merupakan salah satu benda berbahaya yang dapat menghambat perjalanan dari tunanetra. Untuk mendeteksi tangga, digunakan metode pendeteksian garis yang dikenal dengan metode *Hough Line transform*.

Hough Line Transform dirancang untuk menemukan garis pada suatu citra. Metode ini diperkenalkan oleh Paul Hough (1962). Cara kerja dari metode ini adalah membuat persamaan dari suatu piksel kemudian mempertimbangkan semua pasangan yang memenuhi persamaan ini. Semua pasangan ditempatkan pada suatu larik akumulator atau larik transformasi (McAndrew, 2004). Transformasi Hough menggunakan bentuk parametrik dan pemungutan suara terbanyak untuk mendapatkan nilai parameter yang tepat. Apabila dalam suatu citra terdapat beberapa garis yang saling berpotongan pada suatu titik, kemudian apabila titik tersebut ditransformasi ke dalam ruang parameter $m - c$, didapati transformasi dalam ruang parameter $m - c$ adalah sebuah garis lurus dengan persamaan garis yang dinyatakan sebagai, $y_i = mx_i + c$.

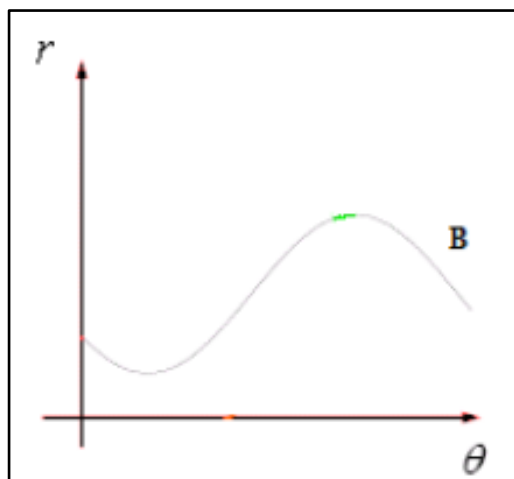
Sebaliknya, jika dalam citra terdapat sebuah garis lurus, dimana garis tersebut mempunyai persamaan $y_i = mx_i + c$, maka jika ditransformasi ke dalam ruang parameter $m - c$ akan diperoleh garis-garis yang saling berpotongan dalam suatu titik. Persamaan transformasi tersebut diperoleh dengan cara memanipulasi persamaan garis, $y_i = mx_i + c$ menjadi bentuk parametrik, $c = y_i - mx_i$.

Perhitungan transformasi dilakukan dengan menghitung tiap titik dalam citra ke dalam semua variasi nilai $m - c$. Pada kenyataannya, apabila ditemui sebuah garis vertikal, maka dapat terjadi masalah dalam penghitungannya karena garis vertikal mempunyai nilai gradien (kemiringan) yang besarnya tak berhingga (∞). Sebagai solusinya, digunakan persamaan $r = x \cos \theta + y \sin \theta$. Dimana gambar koordinat kartesiannya ditunjukkan pada Gambar 2.12.



Gambar 2.12 Garis dalam Koordinat Kartesian dengan Satu Titik Uji (Hanafi, 2014)

Sebuah titik B dalam gambar tersebut apabila ditransformasi ke dalam ruang parameter $r - \theta$ maka akan menjadi seperti pada Gambar 2.13.



Gambar 2.13 Transformasi Satu Titik Uji dalam Ruang Parameter $r-\theta$ (Hanafi, 2014)

Apabila di dalam citra terdapat suatu garis lurus, maka jika garis lurus citra ditransformasi kedalam ruang parameter $r - \theta$ juga akan terjadi suatu titik penumpukan antar kurva sinusoidal dari hasil pentransformasian masing-masing komponen piksel garis yang membentuk sebuah garis lurus.

Pendeteksian tangga menggunakan dasar berupa pencarian garis pada gambar. Metode Hough ini dipilih karena melihat beberapa kelebihan, seperti berikut:

- a. Mampu mendeteksi garis pada bentuk geometri dengan kumpulan tepi meski terputus-putus,
- b. Relatif tidak terpengaruh *noise*,
- c. Parameter yang mempengaruhi *performance* adalah kuantisasi parameter pada ruang Hough.

