

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metode Pengembangan Perangkat Lunak**

Model *waterfall* yang sering juga dikenal sebagai model air terjun adalah model proses pertama yang di perkenalkan. Ia sangat mudah dimengerti dan di gunakan. Dalam model *waterfall*, setiap fase harus diselesaikan sebelum fase berikutnya dapat di mulai dan tidak ada fase yang tumpang tindih. Model *waterfall* adalah pendekatan SDLC paling awal yang digunakan untuk pengembangan perangkat lunak.

Pendekatan *waterfall*, seluruh proses pengembangan perangkat lunak dibagi menjadi fase-fase terpisah. Hasil dari satu fase bertindak sebagai input untuk fase berikutnya secara berurutan. Ini berarti bahwa setiap fase dalam proses pengembangan dimulai hanya jika fase sebelumnya selesai. (Ali Edwar 2019).

Berikut ini penjelasan yang dapat dilakukan untuk masing-masing tahapan:

##### **3.1.1 Analisis**

Analisis kebutuhan merupakan langkah awal untuk menentukan gambaran perangkat lunak yang akan dihasilkan ketika pengembang melaksanakan sebuah proyek pembuatan perangkat lunak.

##### **3.1.2 Desain**

Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda (struktur data, arsitektur perangkat lunak, representasi interface, dan detail algoritma prosedural). Tujuan tahapan ini adalah untuk menentukan secara detail semua material yang akan di gunakan pada perangkat lunak yang akan dikembangkan.

##### **3.1.3 Implementasi**

Implementasi adalah tindakan untuk menjalankan rencana yang telah dibuat. Dengan demikian, implementasi hanya dapat dilakukan jika terdapat sebuah rencana.

### **3.1.4 Pengujian (*Testing*)**

Sekali kode dibuat, pengujian program sudah dapat dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional, yaitu mengarahkan pengujian untuk menentukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

### **3.1.5 Pengembangan (*Development*)**

Tahapan Development merupakan tahapan implementasi software ke customer, pemeliharaan software, dan pengembangan software berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya.

### **3.1.6 Pemeliharaan (*Maintenance*)**

Perangkat lunak akan mengalami perubahan dan penyesuaian setelah disampaikan kepada pelanggan. Perubahan akan terjadi dikarenakan kesalahan-kesalahan, perangkat lunak harus disesuaikan untuk mengakomodasi perubahan-perubahan didalam lingkungan eksternalnya atau karena pengguna membutuhkan pengembangan aspek fungsional atau untuk kerja.

## **3.2 Proses Kerja Aplikasi**

Aplikasi ini berjalan pada kamera letop dengan menggunakan aplikasi visual basic dan bahasa pemrograman C# serta menggunakan metode HOG (*Histogram of Oriented Gradient*). Sebelum merekam gambar atau menangkap suatu objek maka aplikasi harus di jalankan terlebih dahulu dengan mengklik *opent* yang terdapat pada halaman awal aplikasi barulah aplikasi akan berjalan dan melakukan perhitungan manusia secara *real time*.

## **3.3 Metode Pengumpulan Data**

Metode yang digunakan dalam pengumpulan data pada penelitian ini yaitu dengan melakukan studi pustaka (*Library Research*) dan studi lapangan (*Field Research*).

### 3.3.1 Studi Pustaka (*Library Research*)

Studi pustaka adalah teknik pengumpulan data dengan mengadakan studi penelaahan terhadap buku-buku, literatur-literatur, catatan-catatan, dan laporan-laporan yang ada hubungannya dengan masalah yang dipecahkan (Soepomo 2013). Metode ini dilakukan untuk mengumpulkan data melalui panduan-panduan buku referensi dan *literature* lain yang berhubungan dengan pengetahuan teoritis mengenai masalah yang sedang diteliti. Yang kemudian dijadikan data untuk diolah lebih lanjut.

### 3.3.2 Studi Lapangan (*Field Research*)

Adapun metode yang dilakukan dalam studi lapangan adalah sebagai berikut:

a. Observasi

Observasi ini dilakukan pada BMT Assyafi'iyah dengancara mengumpulkan data dengan melalui pengamatan langsung pada objek penelitian yaitu sistem manajemen konfigurasi data, hal tersebut dilakukan guna mendapatkan gambaran secara menyeluruh dan jelas mengenai sistem.

b. Kepustakaan

Untuk mendukung proses penelitian ini, peneliti melakukan pengumpulan data melalui buku referensi atau *literature* lain yang berkaitan dengan penelitian peneliti. Metode kepustakaan merupakan metode penelitian yang dilakukan dengan cara mempelajari sumber-sumber tertulis, seperti buku-buku, dan bahan tertulis lainnya yang berhubungan dengan masalah yang diteliti.

## 3.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk mengetahui spesifikasi fungsi yang mampu dilakukan oleh sistem. Analisis kebutuhan adalah kebutuhan yang berasal dari pemangku kepentingan (*Stakeholders*) termasuk fungsi dan fitur dari sebuah sistem.

Aplikasi deteksi manusia menggunakan metode *Histogram of oriented gradient* ini memiliki beberapa kemampuan, antara lain :

- a. Pendeteksian dilakukan dengan banyak objek manusia atau multidetect, yaitu user melakukan deteksi dari video hasil rekaman kamera yang telah disimpan. Kemudian aplikasi ini melakukan tahapan sesuai dengan tahapan algoritma *Histogram Of Oriented Gradient* dalam tahap ini yang pertama dilakukan adalah normalisasi warna yaitu dengan melakukan convert dari citra berwarna menjadi citra grayscale, kemudian gradient compute yaitu proses menghitung nilai gradient setiap piksel dalam gambar, proses selanjutnya *Spatial Orientation Binning* yaitu proses dimana setiap cells dalam gambar akan dibuat *histogram*. Dalam pembuatan *histogram*nya diperlukan adanya bin untuk mengetahui nilai gradiennya. Bin akan ditentukan sendiri oleh pengguna. Dalam penelitian sebelumnya bin yang digunakan adalah 4 bin orientation, dalam normalisasi block ini terdapat dua geometri block utama yaitu block persegi panjang R-HOG dan melingkar C-HOG akan tetapi dalam penelitian ini yang akan digunakan adalah geometri R-HOG, hasil akhir dalam normalisasi block ini yaitu fitur, tahap selanjutnya adalah Detector Windows merupakan windows atau jendela berukuran 64 x 128 yang digunakan untuk jendela pendeteksian. Jendela pendeteksian ini terdiri dari 8 x 8 piksel dalam tiap cells
- b. Setelah dilakukan deteksi manusia kemudian aplikasi juga dapat menghitung jumlah banyaknya manusia yang terdeteksi tersebut.

### **3.5 Analisis Kebutuhan Non-Fungsional**

Analisis kebutuhan non-fungsional adalah kebutuhan untuk mendeskripsikan kualitas suatu sistem.

- a. Tingkat keakuratan deteksi dan penghitungan manusia pada aplikasi ini termasuk kedalam tingkat keakuratan yang cukup baik, karena cukup sedikit objek yang bukan manusia yang terdeteksi.
- b. Kemampuan dari aplikasi ini mampu mendeteksi manusia dari berbagai sisi (Depan, samping dan belakang) dan juga dalam jarak tertentu, dalam aplikasi ini juga akan terhitung jumlah manusia yang terdeteksi.

### 3.5.1 Analisis Kebutuhan Perangkat Lunak (*Software*)

Komponen yang tidak kalah penting selain perangkat keras adalah perangkat lunak, karena kedua komponen tersebut saling ketergantungan. Perangkat keras tidak akan berarti tanpa perangkat lunak begitu juga sebaliknya. Perangkat keras hanya berfungsi jika diberikan instruksi-intruksi kepadanya. Instruksi-instruksi inilah disebut dengan perangkat lunak. Perangkat lunak yang dapat digunakan dalam perancangan Sistem manajemen konfigurasi data minimal mempunyai spesifikasi sebagai berikut:

- a. Perangkat lunak dalam implementasi :Sistem Operasi Windows atau Linux
- b. Perangkat lunak dalam perancangan sistem : Visual Studio C# 2013

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment (IDE)*, dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe. Saat ini Microsoft Visual Studio terbaru adalah versi 2015.

### 3.5.2 Analisis Perangkat Keras (*Hardware*)

Pada awalnya komputer memang digunakan sebagai mesin hitung yang cepat dan dapat menginput data yang akan diproses sehingga dapat mengeluarkan *output* yang disebut informasi. Namun seiring perkembangan teknologi, komputer telah menjadi alat pengolahan data yang serba guna. Komputer dapat digunakan untuk mengolah data menjadi informasi, sistem komputer membutuhkan tiga komponen, yaitu perangkat keras (*Hardware*), perangkat lunak (*Software*), perangkat akal (*Brainware*) (Nasution, 2004). Perangkat keras yang dapat digunakan dalam perancangan sistem manajemen konfigurasi data minimal mempunyai spesifikasi sebagai berikut:

- a. Perangkat keras minimal dalam implementasi

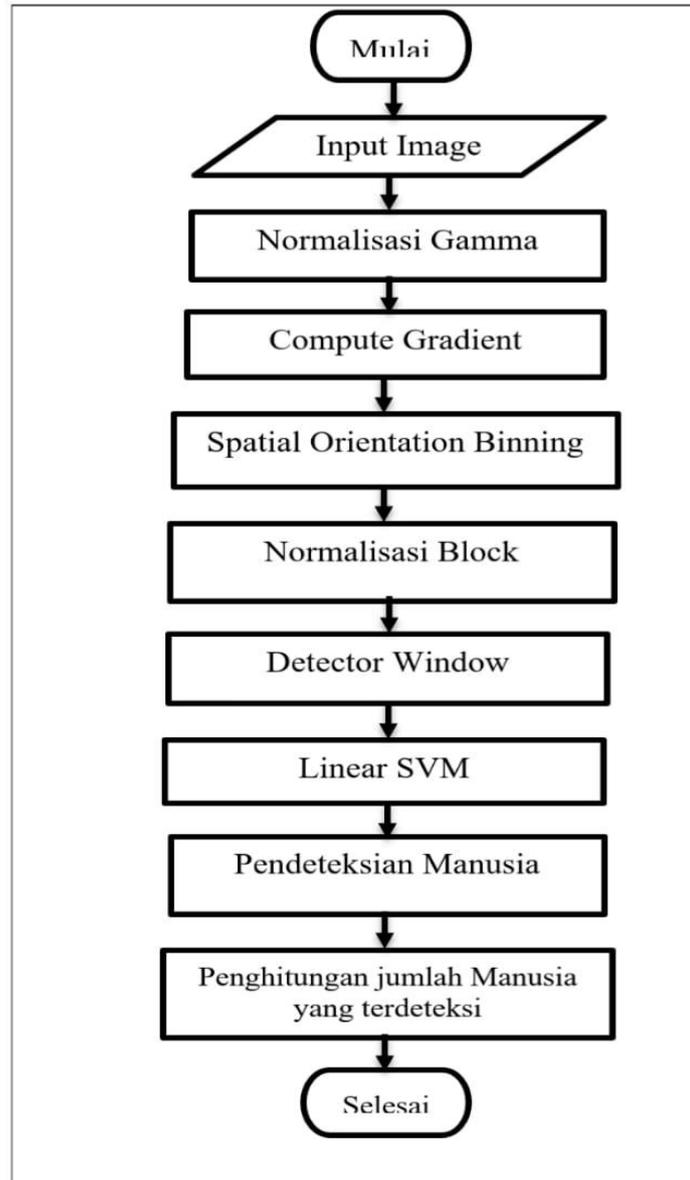
1. Komputer intel celeron
  2. RAM 2 GB
  3. Hardisk 320 GB
  4. Koneksi jaringan komputer lokal atau internet
- b. Perangkat keras dalam perancangan Aplikasi
1. Komputer intel celeron
  2. RAM 2 GB
  3. Hardisk 500 GB

### **3.6 Analisis Pengguna**

Pengguna dari perangkat lunak yang akan dikembangkan ini terdiri dari user, user yaitu melakukan akuisisi citra dan aplikasi akan melakukan deteksi manusia dan perhitungan jumlah manusia yang terdeteksi.

- a. proses Deteksi dan penghitungan manusia

Gambar dibawah ini merupakan alur sistem *Human Counting*. yang dimulai dari tahap awal input *image* hingga *image* tersebut dapat terdeteksi dan terhitung jumlah manusia yang terdeteksi



Gambar 3.1 Flowchart Proses Deteksi Manusia Menggunakan *Histogram Of Oriented Gradient*

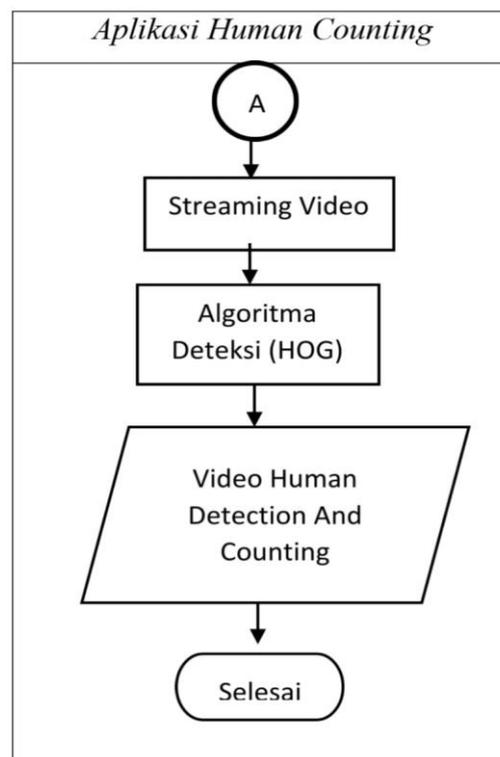
Pada gambar 3.1 merupakan alur atau tahapan mengenai proses pendeteksian hingga proses *human counting* menggunakan algoritma *Histogram of Oriented Gradient Feature descriptor*, pada tahap pertama yaitu input image atau akuisisi citra yaitu proses penginputan image atau objek manusia yang akan dideteksi.

Proses selanjutnya adalah Normalisasi Warna yaitu proses merubah citra asli menjadi citra *grayscale*, proses selanjutnya menghitung *gradient* yaitu proses menghitung nilai gradien setiap piksel dalam gambar, proses selanjutnya *Spatial Orientation Binning* yaitu proses menjadikan setiap cells dalam gambar menjadi

*histogram* dalam pembuatan *histogramnya* diperlukan adanya bin untuk mengetahui nilai gradiennya, proses selanjutnya Normalisasi Block dalam tahap ini biasanya terjadi tumpang tindih karena setiap cells kontribusi nilai lebih dari sekali dan hasil akhirnya yaitu berupa fitur dari objek yang di deteksi, proses selanjutnya Pendeteksian Windows (*Detector Windows*) yaitu proses penyeleksian fitur objek sesuai data training objek yang dilakukan pendeteksian dilakukan per piksel dalam gambar, proses selanjutnya adalah pengklasifikasi objek menggunakan *Support Vector Machine*.

Setelah objek dilakukan klasifikasi dan ditemukan hasilnya manusia kemudian manusia tersebut terdeteksi dan juga akan terhitung jumlah manusia yang terdeteksi oleh sistem tersebut.

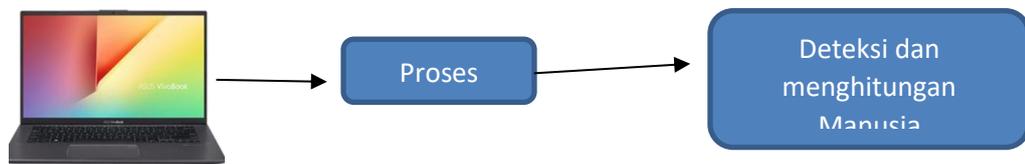
b. *Flowchart* Proses streaming untuk deteksi objek



Gambar 3.2 Aplikasi Proses Pengiriman Data ke *Client*

Gambar 3.2 Meupakan alur proses aplikasi client dalam melakukan deteksi manusia dan penghitungan jumlah manusia. untuk selanjutnya dilakukan deteksi manusia menggunakan algoritma HOG setelah manusia terdeteksi lalu dilakukan penghitungan jumlah manusia yang terdeteksi.

### 3.7 Arsitektur Sistem Secara keseluruhan

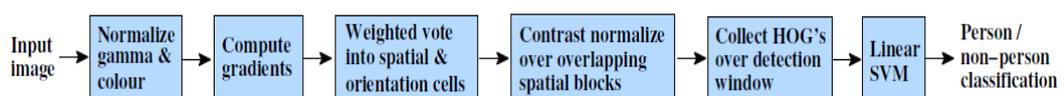


Gambar 3.3 Arsitektur Sistem Secara Keseluruhan

Pada gambar diatas merupakan gambar mengenai alur sistem secara keseluruhan. Pada tahap awal dari proses ini yaitu kamera merekam gambar kemudian data tersebut di convert menjadi image untuk dilakukan deteksi manusia dan penghitungan manusia yang terdeteksi oleh aplikasi deteksi manusia

### 3.8 *Histogram of Oriented Gradient*

*Histogram Of Oriented Gradients* ini digunakan untuk mengekstraksi fitur pada obyek gambar dengan menggunakan obyek manusia. Berdasarkan langkahnya, proses awal pada metode HOG adalah mengkonversi citra RGB (Red, Green, Blue) menjadi grayscale, yang kemudian dilanjutkan dengan menghitung nilai gradien setiap piksel. Setelah mendapatkan nilai gradien, maka proses selanjutnya yaitu menentukan jumlah bin orientasi yang akan digunakan dalam pembuatan *histogram*. Proses ini disebut spatial orientation binning. Namun sebelumnya pada proses gradient compute gambar pelatihan dibagi menjadi beberapa cell dan dikelompokkan menjadi ukuran lebih besar yang dinamakan block. Sedangkan pada proses normalisasi block digunakan perhitungan geometri R-HOG. Proses ini dilakukan karena terdapat block yang saling tumpang tindih. Berbeda dengan proses pembuatan *histogram* citra yang menggunakan nilai – nilai intensitas piksel dari suatu citra atau bagian tertentu dari citra untuk pembuatan *histogramnya*, berikut adalah tahapan dalam *Histogram Of Oriented Gradient*.



Gambar 3.4 Algoritma Metode HOG

a. Input Gambar dan Normalize gamma and color

Input gambar yang digunakan adalah gambar normal yang akan digunakan untuk sampel pendeteksian, yang kemudian akan di konversi kedalam bentuk *grayscale* untuk menghindari kontribusi intensitas yang berbeda dan menormalisasi warna agar dapat dibaca atau dihitung komputer dengan cepat untuk setiap warna (RGB). Seperti pada gambar 3.5



a. Normal

b. *Grayscale*

Gambar 3.5 Contoh Gambar *Input* (Normal).

Proses normalisasi warna yaitu dengan membagi seluruh warna menurut kelompok warna RGB pada setiap piksel. Kemudian akan diambil nilai rata-rata dari perhitungan normalisasi itu yang akan dikirim kembali untuk proses berikutnya. Algoritmanya adalah dengan menjumlahkan seluruh nilai RGB, kemudian dibagi 3, sehingga diperoleh nilai rata-rata dari RGB, nilai rata-rata itulah yang dapat dikatakan sebagai *grayscale*.

b. Menghitung Gradient Image

Setelah mengubah gambar kedalam grayscale, langkah selanjutnya adalah menghitung gradient pada setiap piksel gambar. Metode yang biasa digunakan adalah 1-D *centered*, dengan mencari nilai tengah dari piksel. Dengan mariks seperti berikut :

[ -1, 0, 1 ]

Menggunakan rumus derivatif parsial untuk fungsi *image*  $f(x,y)$  :

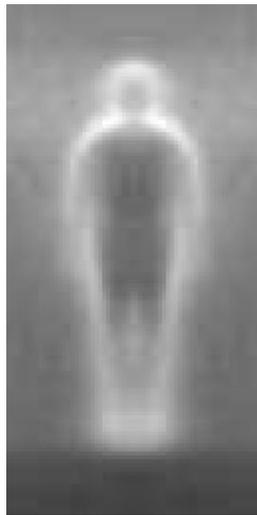
1). Untuk sumbu  $x = \frac{\partial f}{\partial x} = \frac{f(x+h) - f(x-h)}{2h}$

2). Untuk sumbu  $y = \frac{\partial f}{\partial y} = \frac{f(y+h) - f(y-h)}{2h}$

Akan diperoleh nilai  $x$  dan  $y$  yang digunakan untuk menghitung *gradient* :

1). *Magnitude* (besar *gradient*) :  $R = \sqrt{x^2 + y^2}$

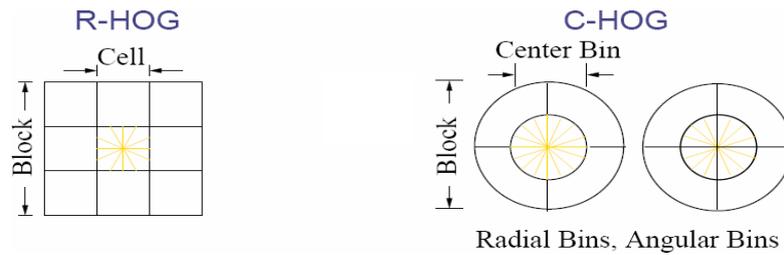
2). Orientasi *gradient* (dalam sudut) :  $\theta = \text{arc tan}\left(\frac{y}{x}\right)$



Gambar 3.6 Contoh Gambar Gradient.

### c. Menghitung *Histogram* dan Orientasi *cell*

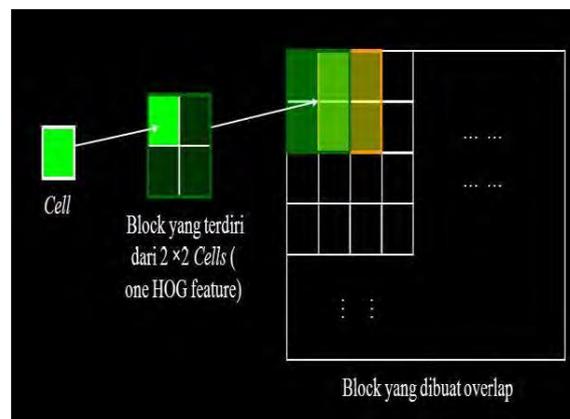
Setelah terbentuk sebuah *gradient* dari citra digital kemudian dihitung untuk citra dibagi ke dalam *cell-cell* yang berukuran  $n \times n$ . Langkah ini bertujuan untuk menghasilkan pengkodean yang sensitif terhadap konten gambar lokal, namun tetap tahan terhadap perubahan kecil dalam pose dan penampilan. Gambar dibagi kedalam beberapa daerah kecil yang disebut sel. Untuk setiap sel akan diakumulasikan *histogram* 1-D membentuk lokal semua piksel dalam sel. Kombinasi *cell-level histogram* 1-D membentuk representasi *histogram* orientasi dasar. Setiap *histogram* orientasi membagi berbagai sudut *gradient* menjadi angka tetap dalam *bins* yang telah ditentukan.



Gambar 3.7 Contoh Gambar R-HOG dan C-HOG, *cell* yang menyusun *block*.

d. *Normalize Overlape Block*

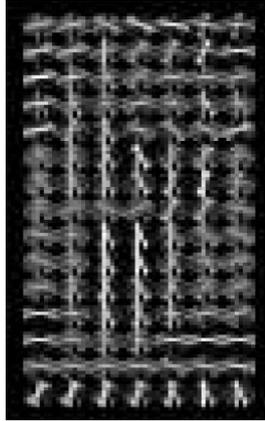
Langkah selanjutnya yaitu masing-masing *cell* tersebut dibentuk histogram-histogram yang nantinya akan digabungkan ke kelompok-kelompok yang lebih besar berukuran  $2n \times 2n$  piksel. Kelompok-kelompok gabungan histogram ini disebut dengan *block*. Setelah itu, dilakukan normalisasi terhadap histogram di setiap *block*. Hasil normalisasi kemudian digabungkan untuk mendapatkan data fitur HOG. *Block* ini saling overlape dengan *block* satu dengan yang lainnya seperti digambarkan di gambar 3.8, *block* pertama digambar kiri, dan *block* kedua digambar kanan.



Gambar 3.8 Contoh Hubungan *Cell*, *Block*, dan *Block Overlape*

e. Fitur HOG.

Bagian sistem selanjutnya adalah visualisasi *feature* HOG. Visualisasi ini berupa garis vektor yang secara keseluruhan menunjukkan karakteristik dari obyek.



Gambar 3.9 Contoh Gambar *Feature* HOG.

*Feature* HOG juga dipakai sebagai input dari *learning* SVM. *Feature* HOG dirubah kedalam *feature vector* dengan ukuran 4608x1. Ukuran *feature vector* dihasilkan dari perkalian dari ukuran block (2x2 *cell*), jumlah bin 9, dan banyaknya *block* yang terbentuk dari *image* dengan ukuran 128x64. *Feature vector* inilah yang dipakai sebagai input untuk proses *learning* SVM. Dalam proses *learning* dilakukan 2 tahap, yaitu *training* data dan tes model. Pada *training* data, *feature* HOG dari *image* positif diberi label +1 yang menandakan bahwa *image* yang dilatih merupakan *feature image* positif dan *feature image* negatif diberi label -1 yang menandakan *image* yang dilatih merupakan *feature image* negatif. *Training* SVM ini menghasilkan sebuah model yang akan dites dengan menggunakan *classify* SVM.

### 3.9 Testing atau Pengujian

Pada tahapan ini aplikasi yang telah diimplementasikan akan di uji secara fungsional dan logik. Hal ini di lakukan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Pengujian aplikasi dilakukan dengan menggunakan metode *black-box testing*. Pengujian yang akan dilakukan yaitu dengan menguji lama waktu *loading* atau *respon time* dari masing masing halaman yang terdapat pada aplikasi. Pengujian dilakukan dengan menggunakan tiga buah perangkat yang berbeda yang memiliki spesifikasi dengan kriteria rendah, sedang dan tinggi dilihat dari segi perangkat kerasnya.